

# **On Agent-Based Semantic Service Coordination**

Cumulative Habilitation Script  
Faculty 6 - Natural Sciences and Technology I  
Saarland University

Kumulative Habilitationsschrift  
zur Erlangung der *venia legendi*  
für das Fach Informatik  
eingereicht bei der  
Naturwissenschaftlich-Technischen Fakultät I  
Universität des Saarlandes

Dr. rer. nat. Matthias Klusch

Saarbrücken, June 2008



---

# Contents

<b>Selected Publications</b> .....	10
<b>Summary</b> .....	13
<b>Preface</b> .....	15

---

## Part I Foundations

---

<b>1 Service-Oriented Computing and Agents</b> .....	21
1.1 Service-Oriented Architectures .....	21
1.1.1 SOA Principle and Relation to Business Process Modeling .....	21
1.1.2 SOA Example .....	22
1.1.3 Technical and Business Drivers .....	24
1.1.4 Developing Service-Oriented Applications .....	26
1.1.5 Critique .....	27
1.2 Web Services .....	28
1.2.1 The W3C Web Services Framework .....	28
1.2.2 Web Service Description and Interaction .....	29
1.2.3 Web Service Discovery .....	32
1.2.4 Web Service Composition .....	35
1.2.5 Service Negotiation and Contracting .....	39
1.3 Web Services and Rational Agents .....	40
1.3.1 Rational Agency .....	40
1.3.2 Relation Between Service and Agent .....	41
1.3.3 Types of Service-Agent Interaction .....	42
1.3.4 Agent-Based Web Service Coordination in Brief .....	44
1.4 Critique .....	47
1.5 Further Readings .....	48

<b>2</b>	<b>Semantic Web</b>	51
2.1	Architecture	52
2.2	Ontologies	54
2.2.1	Classification	54
2.2.2	Ontology Alignment	57
2.2.3	Ontology Aignment Scenarios	58
2.3	Description Logics	59
2.3.1	Syntax and Semantics	59
2.3.2	Relation to FOL	60
2.3.3	Reasoning and Complexity	62
2.4	Semantic Web Ontology Languages	64
2.4.1	RDF and RDFS	64
2.4.2	OWL	66
2.4.3	WSMO and WSML	70
2.4.4	WSMO Framework	71
2.5	Semantic Web Ontologies and Rules	74
2.5.1	Motivation	74
2.5.2	Issues of Combining Rules With Ontologies	77
2.5.3	Combination Strategies	79
2.5.4	DLP	80
2.5.5	SWRL	80
2.5.6	Horn-SHIQ	82
2.5.7	DL+log	83
2.5.8	Nomonotonic DLP	84
2.6	Semantic Web Applications	85
2.7	Critique	87
2.8	Further Readings	90
<b>3</b>	<b>Semantic Web Services</b>	91
3.1	Issues of Semantic Service Description	91
3.1.1	Parts of Service Semantics	92
3.1.2	Structured Representation	92
3.1.3	Monolithic Logic-Based Representation	92
3.1.4	Data Semantics	93
3.1.5	Reasoning on Service Semantics	93
3.2	SAWSDL	93
3.2.1	Annotating WSDL Components	94
3.2.2	Limitations	95
3.3	OWL-S	96
3.3.1	Service Profile	96
3.3.2	Service Process Model	98
3.3.3	Service Grounding	99
3.3.4	Software Support	100
3.3.5	Limitations	101
3.4	WSML	102



3.4.1	Goal	102
3.4.2	Service Capability	104
3.4.3	Service Interface	105
3.4.4	Software Support	106
3.4.5	Limitations	106
3.5	Monolithic DL-based Service Descriptions	107
3.6	Semantic Web Service Coordination	108
3.7	Semantic Web Service Applications	109
3.8	Critique	110
3.9	Further Readings	112

---

## Part II Semantic Service Discovery

---

Introduction	115
4 Hybrid Service Matching with LARKS	141
5 Hybrid Semantic Matching of OWL-S Services	173
6 Hybrid Semantic Matching of WSML Services	182
7 Semantic Service Discovery in Pure P2P Networks	193

---

## Part III Semantic Service Composition

---

Introduction	205
8 Service Composition Planning with OWLS-XPlan1	223
9 Advanced Dynamic Service Composition with OWLS-XPlan2	240

---

## Part IV Agent-Based Service Negotiation

---

Introduction	247
10 Secure Negotiation of Coalitions	271
11 Negotiation of Fuzzy-Valued Coalitions	287
12 Negotiation of Fuzzy Coalitions	303
13 Dynamic Coalition Forming	319

---

**Part V Agent-Based Business Application Services**

---

<b>Introduction</b> .....	329
<b>14 CASA: Integrated Timber Production and Trading</b> .....	339
<b>15 AGRICOLA: Mobile Resource Planning for Cereal Harvesting</b> .....	347
<b>16 CASCOM: Mobile Emergency Medical Assistance</b> .....	357
<b>17 KDEC: Secure Distributed Data Clustering</b> .....	371

---

**Part VI Quantum Agent-Based Service Coordination**

---

<b>Introduction</b> .....	393
<b>18 Quantum Computing and Agents: A Manifesto</b> .....	405
<b>19 Programming of Quantum Search Agents</b> .....	423
<b>20 Quantum Matchmaker Agents</b> .....	430
<b>References</b> .....	437
<b>Description of Author's Contribution to Joint Work</b> .....	459

---

## List of Figures

1.1	Example of a SOA proposed by IBM (2006) . . . . .	23
1.2	Current Web and Web service technology standards . . . . .	30
1.3	WSDL service description structure . . . . .	31
1.4	Web service interaction life cycle . . . . .	32
1.5	Example of Web service interaction life cycle . . . . .	34
1.6	BPEL business service process flow overview . . . . .	37
1.7	Basic types of service-agent interaction . . . . .	42
1.8	Building blocks of agent-based service coordination. . . . .	45
2.1	Layered Semantic Web architecture . . . . .	52
2.2	Types of ontologies . . . . .	55
2.3	Ontology alignment . . . . .	57
2.4	Centralized ontology alignment . . . . .	58
2.5	Decentralized ontology alignment . . . . .	59
2.6	Description Logics: Syntax and Semantics . . . . .	61
2.7	Correspondence between DL and FOL . . . . .	62
2.8	Tractable fragments of OWL . . . . .	69
2.9	WSML language variants. . . . .	73
3.1	Example of semantic annotation of WSDL elements in SAWSDL. . . . .	95
3.2	OWL-S service description elements . . . . .	96
3.3	OWL-S service profile structure. . . . .	97
3.4	Example of OWL-S 1.1 service profile. . . . .	98
3.5	OWL-S service process model. . . . .	99
3.6	Example of OWL-S service process model. . . . .	100
3.7	Grounding of OWL-S in WSDL. . . . .	101
3.8	WSML service and goal description. . . . .	103
3.9	Example of a service request (goal) in WSML. . . . .	103
3.10	Example of service capability in WSML. . . . .	104
3.11	Example of WSML service interface. . . . .	105
3.12	Example of a monolithic DL-based semantic service description . . . . .	108

3.13	Categories of Semantic Web service matchmakers . . . . .	117
3.14	Categories of Semantic Web service discovery architectures. . . . .	130
7.1	Classes of Semantic Web service composition planners . . . . .	207
13.1	CASCOM semantic service coordination architecture. . . . .	335

To Bärbel, Dieter and Dagmar.

## Selected Publications

The following selected publications document my main research activities on agent-based semantic service coordination and are presented in this cumulative habilitation script together with introductions to the relevant research areas. A description of my contributions to joint work is provided in the final section of this thesis.

### Semantic Service Discovery

- 1 K. Sycara, M. Klusch, S. Widoff, J. Lu: LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace. *Autonomous Agents and Multi-Agent Systems*, 5(2), pages 173 - 204, Kluwer Academic, 2002.
- 2 M. Klusch, B. Fries, K. Sycara: Automated Semantic Web Service Discovery with OWLS-MX. Proceedings of the 5th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), Hakodate, Japan, pages 915 - 922, ACM Press, 2006.
- 3 F. Kaufer, M. Klusch: WSMO-MX: A Logic Programming Based Hybrid Service Matchmaker Proceedings of the 4th IEEE European Conference on Web Services (ECOWS), Zurich, Switzerland, pages 161 - 170, IEEE CS Press, 2006.
- 4 M. Klusch, U. Basters: Risk Driven Semantic P2P Service Retrieval. Proceedings of the 6th IEEE International Conference on P2P Computing (P2P 2006), Cambridge, UK, pages 161 - 170, IEEE CS Press, 2006.

### Semantic Service Composition

- 5 M. Klusch, A. Gerber, M. Schmidt: Semantic Web Service Composition Planning with OWLS-XPlan. Proceedings of the 1st International AAAI Fall Symposium on Agents and the Semantic Web, Arlington VA, USA, pages 55 - 62, AAAI Press, 2005.
- 6 M. Klusch, A. Gerber: Fast Composition Planning of OWL-S Services and Application. Proceedings of the 4th IEEE European Conference on Web Services (ECOWS), Zurich, Switzerland, pages 181 - 190, IEEE CS Press, 2006.
- 7 M. Klusch, K-U. Renner: Dynamic Re-Planning of Composite OWL-S Services. Proceedings of the 2nd IEEE Workshop on Semantic Web Service Composition, Hongkong, China, IEEE CS Press, 2006.

## Agent-Based Service Negotiation

- 8 B. Blankenburg, M. Klusch: BSCA-P: Privacy Preserving Coalition Forming Among Rational Web Service Agents. *Kuenstliche Intelligenz*, 1/06, pages 19 - 25, BoettcherIT Verlag, February 2006.
- 9 B. Blankenburg, M. Klusch: On Safe Kernel Stable Coalition Forming Among Agents. Proceedings of the 3rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS), New York, USA, pages 580 - 587, ACM Press, 2004.
- 10 B. Blankenburg, M. Klusch, O. Shehory: Fuzzy Kernel-Stable Coalitions Between Rational Agents. Proceedings of the 2nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Melbourne, Australia, pages 9 - 16, ACM Press, 2003.
- 11 B. Blankenburg, M. Klusch: BSCA-F: Efficient Fuzzy Valued Stable Coalition Forming Among Agents. Proceedings of the 4th IEEE Conference on Intelligent Agent Technology (IAT), Compiègne, France, IEEE Computer Society Press, 2005.
- 12 B. Blankenburg, M. He, M. Klusch, N. Jennings: Risk Bounded Formation of Fuzzy Coalitions Among Service Agents. Proceedings of the 10th International Workshop on Cooperative Information Agents, Edinburgh, UK, Lecture Notes in Artificial Intelligence (LNAI), 4149, pages 332 - 346, Springer, 2006.
- 13 M. Klusch, A. Gerber: Dynamic Coalition Formation among Rational Agents. *IEEE Intelligent Systems*, 17(3), pages 42 - 47, IEEE CS Press, May/June 2002.

## Agent-Based Business Application Services

- 14 A. Gerber, M. Klusch: Agent-based Integrated Services Network for Timber Production and Sales. *IEEE Intelligent Systems*, 17(1), pages 32 - 39, IEEE CS Press, January/February 2002.
- 15 A. Gerber, M. Klusch: AGRICOLA: Agenten für mobile Planungsdienste in der Landwirtschaft. *Künstliche Intelligenz*, 1/04, pages 38 - 42, arendtap Verlag, 2004.
- 16 T. Möller, H. Schuldt, A. Gerber, M. Klusch: Next Generation Applications in Healthcare Digital Libraries using Semantic Service Composition and Coordination. *Health Informatics*, 12(2), pages 107-119, SAGE publisher, 2006.
- 17 J. Costa da Silva, M. Klusch, S. Lodi, G. Moro: Privacy-preserving agent-based distributed data clustering. *Web Intelligence and Agent Systems*, 4(2):221 - 238, IOS Press, 2006.

## Toward Quantum Agent-Based Service Coordination

- 18 M. Klusch: Toward Quantum Computational Agents. In: Computational Autonomy and Agents. M Nickles, M Rovatsos, G Weiss (eds.), Lecture Notes in Artificial Intelligence, 2969, pages 170 - 186, Springer, 2004.
- 19 M. Klusch, R. Schubotz: Programming and Simulation of Quantum Search Agents. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC), Montreal, Canada, IEEE Press, 2007.
- 20 M. Klusch: Quantum Matchmaking. Extended version of short paper "Coordination of Quantum Internet Agents" published in the Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), New York, USA, ACM Press, 2005.

### Kurzzusammenfassung der kumulativen Habilitationsschrift

Die ausgewählten, wissenschaftlichen Veröffentlichungen, auf die ich meinen Habilitationsantrag stütze, repräsentieren meine Forschung zu intelligenten und kooperativen Informationssystemen im Internet seit 1998. Thematischer Schwerpunkt ist dabei die Agentenbasierte Koordination von Diensten im semantischen Web.

Meine Beiträge hierzu umfassen insbesondere innovative Verfahren zur wissensbasierten Suche nach geeigneten Diensten (Kapitel 4 - 6), deren Kompositionsplanung (Kapitel 8 & 9) und Verhandlung in profitablen, sicheren Koalitionen (Kapitel 10 - 13). Diese koordinierenden Aktivitäten werden weitgehend automatisch, proaktiv und transparent für den Benutzer durch individuell rational kooperierende Agenten durchgeführt.

Die praktische Eignung der Verfahren wurde in prototypisch realisierten Agentenbasierten Informationssystemen für verschiedene Anwendungen im Bereich des elektronischen Gesundheitswesens und Landwirtschaft demonstriert (Kapitel 14 - 16). Verfahren und Aspekte einer sicheren verteilten Datenanalyse und Wissensentdeckung durch kooperierende autonome Agenten werden in Kapitel 17 diskutiert.

Als eine Vision für das Internet der Zukunft nach 2020 sehe ich sogenannte Quantenagenten, die auf vernetzten, hybriden Quantenrechnern im sogenannten Quanteninternet ihre Dienste signifikant effizienter und inhärent sicherer koordinieren können als dies im heutigen Internet möglich ist (Kapitel 18 - 20).

Die oben gelisteten Veröffentlichungen werden in dieser Arbeit thematisch geordnet mit jeweiligen Einführungen zum aktuellen Stand der Forschung in den relevanten Wissenschaftsgebieten präsentiert. Eine persönliche Einschätzung des Anteils meiner Beiträge zu diesen Publikationen ist im letzten Abschnitt *Description of Author's Contribution to Joint Work* zu finden.



## Summary

This work is concerned with the study of agent-based service coordination in the Internet, with particular focus on the Semantic Web. Both service-oriented computing and Semantic Web use intelligent agents to coordinate Web services in terms of discovery, composition planning and execution. Besides, in competitive environments with pay per use services, the agents need to enter negotiations in order to resolve their conflicting goals, and to maximize the individual or social welfare of the multi-agent system. However, despite recent advances in related fields, only little, if at all, is known about the characteristics, potential, and limits of these coordination activities and their interrelationships.

Thus, we first present innovative hybrid solutions to the key problems of semantic discovery, and composition planning of Semantic Web services. In particular, we provide strong evidence in favor of the proposition that strict logic-based semantic service selection, as realized by the majority of contemporary Semantic Web service matchmakers, is in general weaker than hybrid semantic service selection. We also contribute to the convergence of Semantic Web services and peer-to-peer computing by means of an original solution to the problem of decentralized and efficient service retrieval in unstructured peer-to-peer networks.

The problem of automated composition planning of Semantic Web services in dynamic environments has not been sufficiently solved yet. For this purpose, we developed the first heuristic-based dynamic composition planner for OWL-S services, and applied it successfully to selected use cases in the e-health domain.

These contributions are complemented by a number of original game-theoretic negotiation protocols. These allow rational provider and consumer agents of non-charge free services to form safe and profitable coalitions under uncertainty. Proof-of-concept implementations of these protocols are used in deployed systems of agent coordinated business application services in different domains such as agriculture and e-health.

Finally, in line with the vision of networked quantum computers to become part of the future Internet beyond 2020, we are pioneering the field of quantum agent-based service coordination. In particular, preliminary design and simulation results of different types of quantum matchmaker agents show that they can be realized, and perform, under certain constraints, way beyond their classic counterparts.



---

## Preface

The continuous proliferation of Web services that encapsulate business software and hardware assets, e-commerce or social software applications in the Web 2.0 holds promise to revolutionize the way of interaction within today's society and economy. In a world of ubiquitous services, it is just the service value that counts for a customer and not the software components or networked computing devices that implement the service.

Web services are adopted by major business stakeholders such as IBM, Credit Suisse, Microsoft, Siemens, and SAP as the basic and reusable building blocks of service-oriented architectures out of which new and interoperable, loosely coupled business process applications and systems can be created. This leads to added value client-provider interaction within and across enterprises. It comes as no surprise, that the currently prevailing application domains of Web service technology are enterprise application integration, e-government and e-business (cf. chapter 1).

One major challenge for this technology is intelligent service coordination, that is, to discover, compose, negotiate, and execute heterogeneous Web services in an efficiently automated and coherent way with minimal human intervention in a given application context. In fact, intelligent Web service coordination is considered a key enabler of advanced e-business applications that go beyond of what contemporary approaches to manually orchestrated business services can deliver.

For example, efficient means for automatically searching and composing Web services would allow collaborating service providers to extend their business by wholesaling requested components and resources faster and more flexible on demand. The same principle applies to dynamically adapting the user-generated contents of provided Web 2.0 services to the changing needs of individual service user communities - which would allow the respective providers to stay competitive in the market.

However, from the perspective of strong AI, in particular symbolic knowledge representation and reasoning, any automated and semantics based coordination of services appears hard to achieve without any well-founded logic specification of service semantics upon which intelligent agents could deliberately reason. Unfortunately, contemporary XML-based Web services are lacking formal semantics.

Meanwhile it is common knowledge that this problem can at least be partially solved by exploiting Semantic Web technologies (cf. chapter 2). Key idea of the Semantic Web is to add more meaning to Web resources by semantically annotating them with concepts and rules from agreed-upon or aligned domain models or ontologies which semantics are formally specified in an appropriate logic theory. For example, the standard ontology language OWL is grounded in description logics, while the rule language WSML-Rule relies on well-founded logic programming.

In particular, Web service semantics can be encoded in such a way that intelligent agents can indeed "understand" the meaning of individual and composed services by means of logical reasoning. Different Semantic Web service description frameworks exist for this purpose such as OWL-S, WSML, SWSL, and the recently announced W3C standard SAWSDL, which is, however, only semi-formal in the sense that it refers to formal semantics of service annotations and their handling just outside the given framework (cf. chapter 3).

Both Semantic Web and service-oriented computing use intelligent agents to proactively discover, compose, and execute relevant Web services on demand, individually or in joint cooperation with other agents. In environments with contested services, they need to enter service negotiations in order to resolve conflicting goals, and to maximize the individual or social welfare of the respective multi-agent system. Other issues of agent-based semantic service coordination include safety, privacy, and trust, as well as the dynamics, and scalability of applied coordination means.

Such a synthesis between Semantic Web, services, and agents, may eventually lead to a significant improvement of the way and quality of task oriented Web service provision to the human user [31]. In fact, the convergence of Semantic Web, Web 2.0, and Web services to the so-called service Web 3.0 populated with personal agents is commonly expected to be the next step in the evolution of the Web. However, despite recent advances in related fields, only little is known about the characteristics, potential, and limits of semantic service coordination.

This work provides innovative solutions to following open problems of the field: What is the trading off between quality and costs of Semantic Web service discovery by means of logic-based, approximated, or hybrid forms of semantic matching? How can we efficiently retrieve Semantic Web services in unstructured peer-to-peer service networks without any central coordination means and no prior knowledge about the environment? If the search for relevant atomic services fails, how to exploit AI planning techniques for dynamically composing a complex service that eventually satisfies the given goal? This is a real challenge in open, decentralized and competitive business service environments in which central composition planning with interleaved execution of non-local services (or conditional planning-based fault-tolerant service composition) is prohibited for reasons of autonomy, costs, and efficiency.

We shall also address the problem how cooperative and rational providers can best form coalitions to maximize their individual profits by sharing and com-

binning services on demand: How to negotiate these coalitions with reasonable degree of data privacy, anonymity, and safety against fraud by potential trading partners? How to form stable coalitions with imperfect knowledge of joint profits and individual payoffs? How can resource-bounded service providers limit the financial risk of joining a coalition caused by its possible failure of providing a composed service within a given deadline? How can they efficiently react to dynamic changes in the set of trading partners without restarting the whole negotiation process? What are reasonable options for managing the relationships between all of these service coordination activities?

Finally, the rapid progress made in the building of quantum computing and communication devices may lead to sophisticated quantum computer networks become part of the Internet beyond 2020. This so-called "Quantum Internet" may be populated with new forms of intelligent agents that are capable of both classical and quantum computing. How can quantum agents and multi-agent systems be realized with computational benefits in general, and future prospects of service coordination in particular?

### **Organisation of this work**

This work covers main results of my research on agent-based semantic service coordination in the period of 1998 to 2007 at different institutions (DFKI in Saarbrücken, Carnegie Mellon University in Pittsburgh, Free University of Amsterdam), and is structured into the following six parts.

Part I briefly introduces the reader to the basics of service-oriented computing and agents, the Semantic Web, and Semantic Web services. Readers familiar with these basics, or parts of them, can skip all, or parts of it. In subsequent parts, this foundational part is complemented with brief introductions to semantic service discovery, composition planning, and negotiation, as well as quantum information processing required to understand my own contributions to these fields.

Part II is dedicated to innovative means of Semantic Web service discovery. The contributions provide an operational analysis of the potential and limits of logic-based and hybrid semantic matching of Semantic Web services in OWL-S and WSML. In particular, we present first approaches to hybrid service matchmaking that are shown to outperform existing solutions under certain constraints. Further, we provide an efficient solution to the problem of semantic service discovery in unstructured peer-to-peer networks. Proof-of-concept implementations of these contributions are available as ready-to-use tools most of which we used, for example, in military and e-health service applications.

Part III is devoted to Semantic Web service composition with particular focus on services written in OWL-S. We present and discuss a first solution to the

problem by means of dynamic service composition planning as opposed to reactive or contingency planning, its experimental evaluation, and applications in selected complex use case scenarios in the e-health domain.

Part IV complements these results with solutions to the problem of negotiating pay per use services. The contributions are original game-theoretic negotiation protocols that allow provider and consumer agents of user-charging services to form safe and individually profitable coalitions under uncertainty and with trust.

Part V provides a selection of prototypically implemented systems of agent-coordinated business application services in different domains that use some of the contributions of the previous parts. In particular, a layered architecture for agent-based coordination of Semantic Web services and its application to the e-health service domain is presented. This work was mainly done in the context of the European research project CASCOS and the national project SCALLOPS.

Part VI concludes with contributions to agent-based service coordination in the future Internet including networked hybrid quantum computers. In particular, we propose a classification of quantum agents, and a generic hybrid architecture of quantum agents. Further, we develop quantum matchmaking and search agents, and show that they can outperform their classical counterparts and are feasible to implement on hybrid quantum computers.

### **Acknowledgements**

The work presented in this thesis has been funded in part by the European Commission in the CASCOS project (FP6-IST-511632), the DARPA DAML grant F-30601-00-2-0592, DARPA grant F-30602-98-2-0138, the Office of Naval Research grant N-00014-96-16-1-1222, the German Ministry for Education and Research (BMB+F) in the projects SEMAS (2001 - 2003) and SCALLOPS (01-IW-D02, 2004 - 2007), and the Saarland Ministry for Environment in the projects CASA (2000 - 2002), and AGRICOLA (2002 - 2003). Personally, I am particularly indebted to Jörg Siekmann for his continuous strong support of my research activities in general, and this work in particular. Cordial thanks also go to my colleagues and friends at different research institutions and universities, in particular to those at the multi-agent systems group, and my research team on intelligent information systems and agents at DFKI for a very fruitful collaboration over the past years in several joint projects related to this work. Last but not least, I am particularly thankful to my wife, Barbara, for her enduring patience and encouragement during the preparation of this work.

Saarbrücken, March 2008

*Matthias Klusch*

**Foundations**





## Service-Oriented Computing and Agents

A service can be defined as a kind of action, performance, or promise that is exchanged for value between provider and client. In other words, it is a provider-client interaction that creates and captures value for all parties involved [345, 343]. One guiding principle of the global economy since the 1980s is that, in order to survive, every business should become a service business with long-term customer relationship management instead of relying on mere transactional marketing and focussing on mass production of goods only [315]. Web services and their agent-based coordination are the latest development and hype of this evolution toward a global service economy. In the following, we introduce the reader to the basic terms, and characteristics of service-oriented computing, Web services, and agents. References to further readings on the subject are given throughout and at the end of the chapter.

### 1.1 Service-Oriented Architectures

In the late 1990s, the paradigm of service-oriented computing (SOC) and architectures (SOA) emerged mainly as a response to the way how enterprises conduct their business. Fully integrated enterprises are being replaced by loosely coupled business networks and systems in which each participant provides the others with specialized services. One key issue of this approach is that potentially heterogeneous, geographically dispersed modular business resources are advertised as services to consumer applications using self-contained, standardized machine-interpretable descriptions, and then provisioned dynamically on demand, often as part of complex business process workflows within and across enterprises.

#### 1.1.1 SOA Principle and Relation to Business Process Modeling

There is no common agreement on the notion of service-oriented system architecture (SOA) yet. At its most basic, any SOA is an architectural pattern

of a collection of services on a network that communicate with one another (Datz, 2004)[84]. More concretely, the principle of any SOA, also referred to as the SOA principle, is that existing enterprise application resources are published as a group of self-contained enterprise software components each of which providing one or multiple services with standard based interfaces that can be reused for the engineering of customer tailored business processes. For example, verifying a credit card transaction or processing a purchase order. In fact, the services can be composed into one or multiple business processes or applications, and communicate with each other via event-driven messages. Services are loosely coupled, meaning that any business service application doesn't have to know the technical details of another application in order to talk to it, have well-defined, platform-independent interfaces, and are reusable. In this sense, the SOA principle advocates a higher level of application development (also referred to as coarse granularity) that, by focusing on business processes and using standard interfaces, helps mask the underlying technical complexity of the enterprise IT environment. Although this principle was established before Web services came along, and the concept of a service within a SOA is per se independent of the concept of a Web service, the majority of service-oriented business systems in practice employ them since they naturally implement the SOA principle by using lightweight protocols based on widely accepted standards such as those of the Web service standard stack of the Web consortium (cf. section 1.2).

How does the SOA principle relate to business process management in general? Business processes are usually modeled in a concrete process modeling language such as workflow nets, event-driven process chains, YAWL (Yet Another Workflow Language), or the BPM (Business Process Modeling) notation. Concrete architectures to enact business processes include workflow management architectures, case handling architectures, and - service-oriented architectures. In the latter case, one or multiple services commonly implemented as Web services and described in WSDL (Web service description language) with their choreography and orchestration modeled in BPEL (business process execution language) are (re-)used to realize one or multiple business processes within or across enterprises.

We will discuss Web services and the corresponding conceptual SOA model in section 1.2. For a more comprehensive and accessible treatment of business process modeling and service-oriented architectures, we refer the reader to, for example, the volume of Weske (2007)[382].

### 1.1.2 SOA Example

As mentioned above, the notion of SOA still means different, sometimes even conflicting things to different people. While business executives and consultants often sell a SOA as a set of services that a business wants to expose to their customers, or other parts of their organization, system architects consider SOA as an architectural style, that is a set of architectural principles,

patterns and criteria which address characteristics such as modularity, encapsulation, loose coupling, separation of concerns, reuse, and composability that is enabled by standards, tools, and technologies such as Web services.

As one consequence, a variety of different proprietary architectures and models exists calling for a standardisation effort to allow for fast and widespread adoption of the SOC paradigm across all organisations developing or using services. In August 2006, the organization for the advancement of structured information standards (OASIS) eventually approved the OASIS Reference Model for Service Oriented Architecture V 1.0<sup>1</sup>. An example of a layered SOA that is compliant with this reference model as proposed by the OASIS member IBM is shown in figure 1.1.

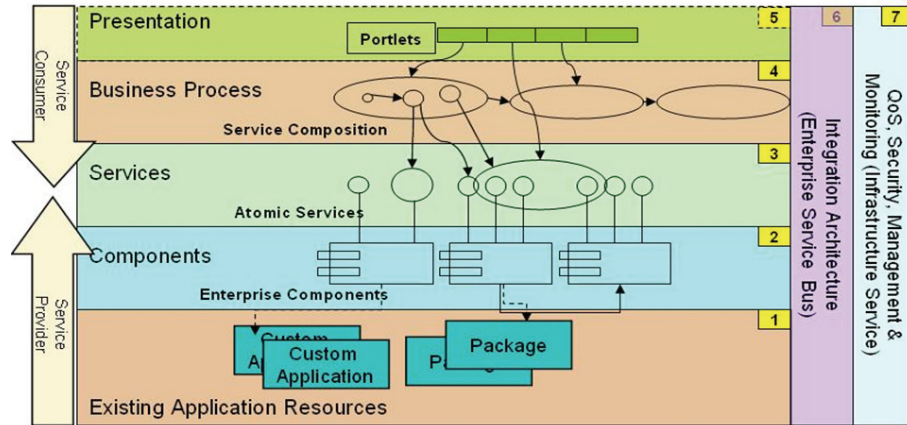


Fig. 1.1. Example of a SOA proposed by IBM (2006)

At the core of this SOA proposal is the service model that defines services and their underlying components of existing business application resources together with the enterprise service bus (ESB). The ESB is a refined broker pattern for decentralized, standards based and asynchronous communication between loosely coupled heterogeneous software (services) running on different platforms and devices. It does not implement a SOA per se but provides the cross-cutting concerns and features with which one can be implemented: Support of messaging (synchronous, asynchronous, point-to-point, publish-subscribe), standards-based adapters for integration with legacy systems, support for service orchestration and choreography (cf. section 1.2), and intelligent content-based routing services. The functionalities of enterprise components are separately offered by means of atomic services. Service

<sup>1</sup> [www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf](http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf)

compositions implement different business application processes each of which functionality is available to the customer at the enterprise portal in the Web.

### 1.1.3 Technical and Business Drivers

The main technical driver of promoting SOA and respective software support by major business stakeholders like BEA Systems, IBM (with WebSphere), Oracle, Microsoft (with .NET), SAP (with NetWeaver) and Sun is the identified need to replace monolithic and proprietary business applications, custom middleware and tools with loosely coupled, logically layered and physically distributed vendor components by means of standards-based services [393, 281].

*Technical drivers: Loose coupling and interoperability*

In practice, XML standard-based Web services are commonly considered the main basic and reusable building blocks of SOAs out of which new and interoperable, loosely coupled Web-based business process applications and systems can be created for added value client-provider interaction. This has been early demonstrated by, for example, Credit Suisse and IBM.

Web service enabled service-oriented architectures, in contrast to traditional enterprise application integration (EAI) models, offer the advantage that (a) business functionality in a SOA can be reused to a higher degree than in more tightly coupled SOAs realized with, for example, CORBA (Common Object Request Broker Architecture); (b) relying on standards it provides a highly flexible and adaptable implementation; and (c) it becomes eventually possible to switch from a particular service to a different one without adaptations.

In other words, the SOA principle represents an evolution from traditional means of tightly coupled applications including CORBA to loosely coupled applications by means of Web services. Tight coupling makes it hard for applications to adapt to changing business requirements, as each modification to one application may force developers to make changes in other connected applications. Also, object-oriented development as in CORBA uses a finer level of granularity - objects can be defined at the level of employee or customer order. In an SOA, a service is defined at a more abstract level, that is a business process such as generating a phone bill.

That is, the SOA principle provides a loosely coupled modular solution to enterprise application landscapes avoiding the central point of integration (star topology), often a bottleneck in traditional EAI solutions. Furthermore, it still reduces the number of point-to-point adapters since every interface is based on standard WSDL and can communicate with every other WSDL enabled interface (cf. section 1.2). However, what the SOA principle does not solve is the problem of documenting the semantics of these service interfaces which we discuss later.

*Business driver: Productivity cost reduction and multi-channel sales management*

Expected benefits of such replacement include the flexible share of interoperable application building blocks across lines of heterogeneous business and projects related to, for example, business-to-business (B2B) applications, and enterprise application integration (EAI). The latter is traditionally realized by a monolithic stack in a hub-and-spoke architecture. These benefits determine the currently main business drivers of SOAs which are a significant cost reduction in productivity and an improved multi-channel order management support of wholesaler-retailer business. This is achieved through (a) automated service composition on demand, and (b) better customer focussed initiatives involving multiple sales channels with more clear separation of concerns such as presentation and application business logic.

Examples from the B2B, and the e-government domain are the realization of flexible services-based least cost supply chain management in the automotive and telecommunications industry, and public Web-based governmental services provided to the citizens for administration purposes. Fast and automated composition of Web services selected best in terms of price and quality for specific product bundles would allow service providers to extend their business through wholesaling of required components and resources on demand. This is supporting providers to stay competitive in the global service economy.

*Business driver: New customer relationships by Web 2.0 services*

Another business driver are the expected huge profits from business-to-consumer services in the Web 2.0. In particular, both Web services and Web 2.0 technologies such as Ajax for asynchronous Java scripting, XML-based microformats for limited semantic content annotation, and RSS (Real Simple Syndication) feeds, enable developers to build more sophisticated, customer tailored business application services in the Web, and to mash up their content via XML-based interfaces. Prominent examples of such "social" (networking) services in the Web 2.0 with user-generated content are Google's YouTube, flickr, NewsCorp's MySpace, facebook, and Ebay's Skype. Together with so-called social software for developing blogs and wikis this type of services is considered making the Web more social to the user, though only partially attractive to business as well<sup>2</sup>.

Main reason for this unprecedented win-win situation is that community building is a key feature of the Web 2.0: Service-oriented tightly knit user communities jointly create and share the precious content of provided Web services

---

<sup>2</sup> Google purchased YouTube for 1.65 billion USD, NewsCorp bought MySpace for 580 million USD, and Ebay swallowed Skype at the cost of 2.6 billion USD. It is notable, however, that according to a recent study (published in April 2007) by McKinsey market analysts, Web 2.0 technologies, except Wikis and blogs, such as RSS, podcasts, news and (socially) tagged bookmarks are not considered important by major business stakeholders for B2B applications yet.

to pursue various tasks of everyday life. As a consequence, customer-provider relationships are inherently formed which for the provider come at no costs but the promise of a new and stable source of monetary gain. Major factor of securing these business profits is the size of these user communities which strongly depends on the availability, maintenance, and extension of customized high quality service data for sharing and reusing. That can be achieved, for example, by means of distributed and topic-oriented service compositions on demand in rational strategic coalitions of relevant providers.

#### 1.1.4 Developing Service-Oriented Applications

To date, there is no commonly agreed upon general purpose methodology for the development of service-oriented business applications and systems. The international OSOA (Open Service Oriented Architecture) consortium of major business stakeholders is currently working on a so-called service component architecture (SCA) <sup>3</sup> which rather aims at simple and platform independent ways of developing SOA-based applications.

##### *Service Component Architecture (SCA)*

The SCA (Service Component Architecture) promises to support SOA implementations in one of many object-oriented or procedural programming languages such as Java, PHP, C++, COBOL, XML-centric languages such as BPEL and XSLT, and also declarative languages such as SQL and XQuery. It also allows to build SOA-based business applications in a range of programming styles, including asynchronous and message-oriented styles, in addition to the synchronous call-and-return (RPC) style. Different service access mechanisms can be used such as Web services (cf. section 1.2), messaging systems and CORBA IIOP; the respective bindings and infrastructure facilities like security, and transactions are handled declaratively and are independent of the implementation code.

##### *Service Data Objects (SDO)*

Related efforts on developing means to uniformly access and manipulate data from heterogeneous data sources, including relational databases, XML data sources, Web services, and enterprise information systems resulted in the service data object (SDO) architecture. SDO is based on the language neutral concept of disconnected data graphs in which a data graph is a collection of tree-structured or graph-structured data objects. Using SDO APIs a client retrieves a data graph from a data source, mutates the data graph, and can then apply the data graph changes back to the data source. The task of connecting applications to data sources is performed by data mediator services.

---

<sup>3</sup> [www.osoa.org/display/Main/Service+Component+Architecture+Specifications](http://www.osoa.org/display/Main/Service+Component+Architecture+Specifications)

Both, SCA and SDO specifications are open source, supported by a range of tools, but are still under development. Currently, there exist SDO specifications for C++ and Java. However, it is not yet clear, whether this combined approach of SCA and SDO will eventually comply with the OASIS reference model for SOAs, and be adopted in practice, once it has been finalized.

#### *Middleware technologies*

Established technologies for realising SOAs include object-oriented middleware (OOM) for distributed computing like OMG CORBA (Common Object Request Broker Architecture), Sun's JINI, Microsoft DCOM, Java's light-weight component model EJB3, and message-oriented middleware (MOM) like IBM's MQseries and TIB/Rendezvous. For example, in contrast to Web services which are mostly based on SOAP/HTTP, application services in CORBA typically communicate via the IIOP (Internet Inter-ORB Protocol). Second, CORBA services are tightly coupled compared to the loose coupling between Web services: In CORBA, objects are shared between components, whereas Web services communicate primarily over messages.

In MOM, unlike OOM, messages are generally untyped, asynchronous, and the internal structure of messages is the responsibility of the application. This is typically designed as publish-subscribe systems. Though some vendors like IONA use CORBA to realize SOAs, these middleware platforms developed in the 1990s have not been adopted to quite the same extent by industry than it currently appears to become the case with Web services. In fact, the most prominent of contemporary approaches to realize SOAs is the Web services framework of the Web Consortium (W3C) which we will briefly present next, after some rather general critics on the SOC.

#### **1.1.5 Critique**

Main criticism of the SOC paradigm is that in general none of its characteristics and techniques is really new. In essence, SOC is adopted by industry from other domains like OOP (object-oriented programming), AOP (aspect-oriented programming), component-based and distributed computing in order to synthesize a standards-based solution to, for example, the enterprise application integration problem.

But what is wrong with that? OOP and SOA, for example, share indeed many characteristics like encapsulation, information hiding through interfaces, but their differences include the remote objects and call stack vs. document-centric messaging, and object name and type versus service bindings and contracts (service level agreements) as linking elements. Hence, OOP is a general purpose programming paradigm on a micro-level whereas SOA can be seen as an architecture style for the problem of enterprise application integration on the macro level.

However, some critics of SOA are questioning more specifically the effectivity and efficiency of maintaining as well as verifying safety and security of large

scale distributed systems of business application services. These issues have to be further investigated by the SOC community.

## 1.2 Web Services

As mentioned above, Web services technology can be used to realize SOAs. According to the W3C, a Web service is a software system designed to support interoperable machine-to-machine interaction over a network. Despite the often interchangeably used terms of Web service and SOA, they are, in principle, not the same thing. As mentioned above, SOAs may not primarily base on Web services but can be built on object-oriented (OOM) or message-oriented middleware (MOM). On the other hand, not every deployed Web service-based system embraces each of the guiding principles of SOAs like the one of loose coupling.

### 1.2.1 The W3C Web Services Framework

The W3C Web services framework is manifested by a set of technical specifications such as WSDL (Web Service Description Language) and SOAP (Simple Object Access Protocol) that codify mechanisms for XML-based interoperability between heterogeneous business services. A Web service exposes operations which consume inputs and produce outputs both encoded in XMLS (XML Schema), and can be communicated with over HTTP and SOAP messaging. These elements of a service interface are described in the machine-readable W3C standard language WSDL. Web service operations define the way in which messages are handled, that is, for example, whether an operation is a one-way, request-response, solicit-response or notification operation.

#### *Stateless Web services*

Please note that Web service interaction, however, is not restricted to SOAP but can take the form of, for example, the simple REST (REpresentational State Transfer) style interaction using lightweight XML messages over (stateless) HTTP based on protocol primitives like HTTP-POST/GET/PUT. In essence, Web services are stateless message processors that accept request messages, process them in some fashion, and (usually) formulate a response to return to the requestor. They are typically implemented by stateless components such as Java servlets or EJB (Enterprise Java Beans) stateless session software components. Therefore, a Web service (a stateless entity) is separate from any persistent state (of a Web server) that it might need in order to complete the processing of request messages. A stateful service requires the service providing Web server to maintain the states of service interaction sessions which is not possible with stateless protocols like HTTP and SOAP.



However, Web services can be made to act as if they were stateful by arranging for the Web server to send the state (or some representative of the state like in the REST protocol) to the client, and for the client to send it back again next time to remind the server of the state. Apart from REST, other approaches to make stateless HTTP-based Web service interactions or sessions appear stateful include Cookies<sup>4</sup>, URL rewriting<sup>5</sup>, hidden fields in forms<sup>6</sup>, and session variables<sup>7</sup>.

### *Web service standards*

The process of Web service coordination encompasses all activities related to service discovery, composition, negotiation, and execution; we discuss each of them in more detail later. Relevant work from the W3C on the subject include the W3C service interaction life cycle for finding WSDL services in registries like UDDI via SOAP API, the standards WS-Coordination, WS-Agreement, and WS-Transactions. Other Web and Web services related standards issued by the W3C are summarized in figure 1.2.

In subsequent sections, we shall briefly present WSDL and SOAP, summarize current approaches to Web service coordination, discuss the relationship between Web services and rational agents, argue for agent-based service coordination, and conclude this chapter with major criticism of Web services. For a more comprehensive treatment of Web service technologies, related standards, and applications, we refer to, for example, the excellent volumes [6, 292], and the proceedings of major conferences in the field such as the European conference on Web services (ECOWS), and the international conferences on Web services (ICWS) and service-oriented computing (ICSOC).

## **1.2.2 Web Service Description and Interaction**

The standard language WSDL allows to specify self-describing service descriptions to enable automated discovery of, access to, and to ease the maintenance

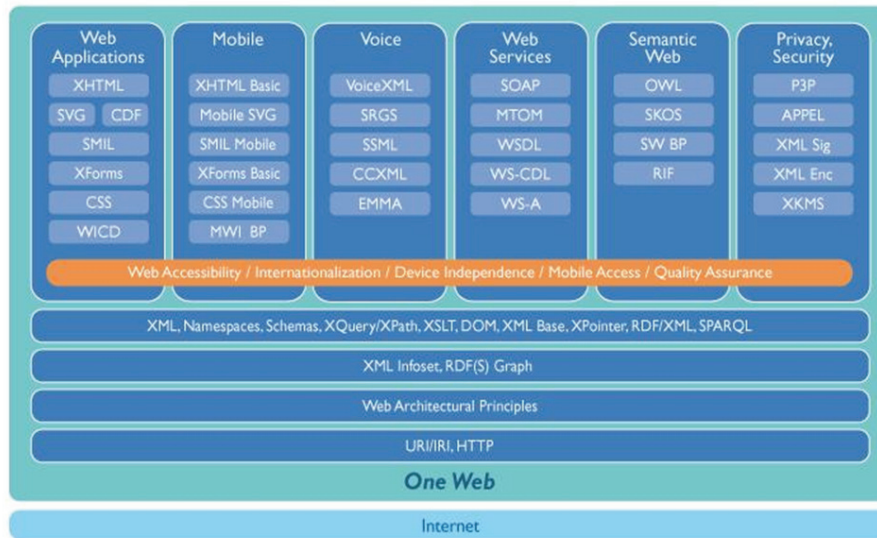
---

<sup>4</sup> Server (Web site) sends the client a unique reference (session identifier) to a session state (maintained at the server). These cookies are persistently stored at the client site (cookie cache) and automatically sent back to the server in HTTP-headers when the client next visits the site such that the server can uniquely restore past sessions with the client.

<sup>5</sup> The session state is sent as part of the response and returned as part of the request URI.

<sup>6</sup> The state (data) is sent from the server to the client in a hidden field as part of the response, and returned by the client to the server as part of a form's data (which can be in the request URI or the POST body, depending on the form's method).

<sup>7</sup> Server itself can store and maintain data (in files) keyed by a session variable (session id) which the client sends back in hidden field or cookie; so nothing is transmitted to the client (except for a session name and id). It is therefore not possible for the client to view or edit this session data.



**Fig. 1.2.** Current Web and Web service technology standards

of service code. WSDL defines services as collections of network endpoints or ports. The abstract definition of endpoints and messages is separated from their concrete network deployment or data format bindings. Figure 1.3 outlines the structure of WSDL 1.1 service descriptions.

The data type part of a WSDL service description references required namespaces.

The message part contains protocol independent messages contained within the requester's query and the services response (body in SOAP). A typical transaction consists of two messages, though several messages may be defined for different transactions. The I/O interface of the service ("portType") contains pre-service operation(s), input messages for service parameters, and output messages for return values of the service; if necessary, error messages describing the set of error conditions are also included. However, a WSDL description of a service does not include the code of the service but the pointer to the hosting site. In fact, the implementation part specifies the service interface in terms of the binding of the interface to specific protocols for transport and messaging (binds operations to physical URLs and ports), and the network location (URI) where the interface has been implemented, i.e., the actual location or endpoint at which the service can be invoked. WSDL does not allow to define stateful service descriptions in terms of preconditions and effects.

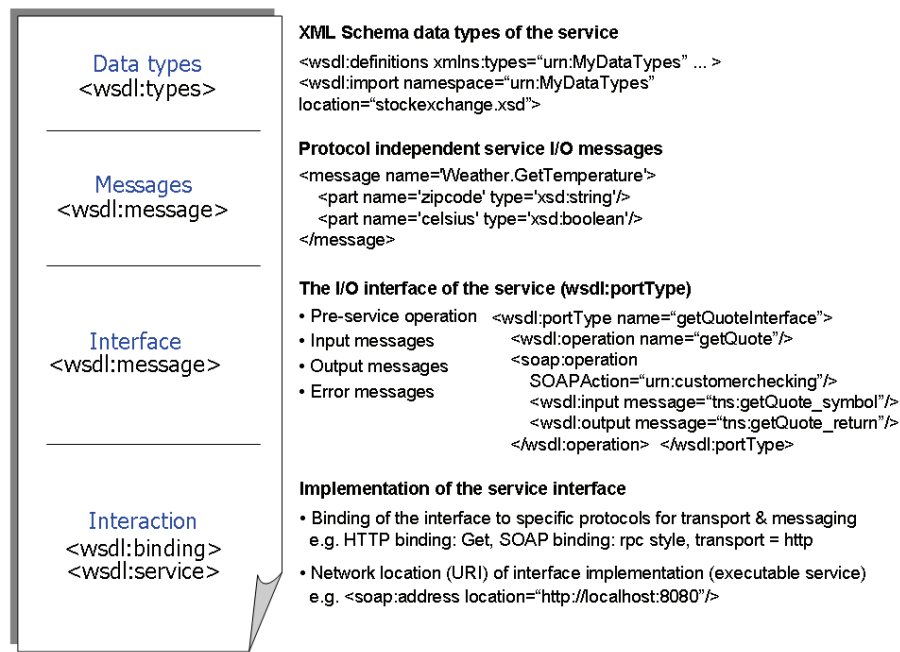


Fig. 1.3. WSDL 1.1 service description structure

### Service interaction via SOAP

Other systems can interact with a given Web service in a manner prescribed by its interface description using messages specified in SOAP (Simple Object Access Protocol). SOAP is a XML-based communication protocol for messaging defined on top of the ISO/OSI TCP/IP transport and network layers. More concrete, it is a message layout specification that defines a uniform way of passing XML-encoded data. SOAP is a W3C recommendation since June 2003. Data types of SOAP messages are defined in XMLS documents referenced by respective (XLS) namespaces declared in the header. Each SOAP message can also reference a method (SOAPAction) the HTTP server has to execute before decoding the rest of the message which can be used to, for example, pre-filter unsolicited requests. SOAP allows for both RPC-style and document-style messaging via HTTP with an XML serialization.

For an accessible account of how to program Web services in Java with detailed examples of WSDL service descriptions and SOAP messaging, we refer the interested reader to, for example, [63]. Prominent Web service development frameworks include Java WSDP/Expresso, IBM WebSphere, and Microsoft ASP.Net-WSE.

### 1.2.3 Web Service Discovery

Service discovery aims at coordinating the ultimate service requester with the ultimate service provider agent. This coordination problem can be solved by means of either assisted mediation through specialized software agents, so-called middle agents, such as matchmakers, brokers and mediators, or in a peer-to-peer fashion, or a combination of both [212].

#### *Web service interaction life cycle*

A typical instance of assisted service mediation through matchmaking is the W3C standard Web service interaction life cycle for service-oriented architectures (also referred to as the Web service role model, or the SOA model for Web services) between consumer, registry, and provider of a Web service as shown in figure 1.4. The Web service role model describes a specific service

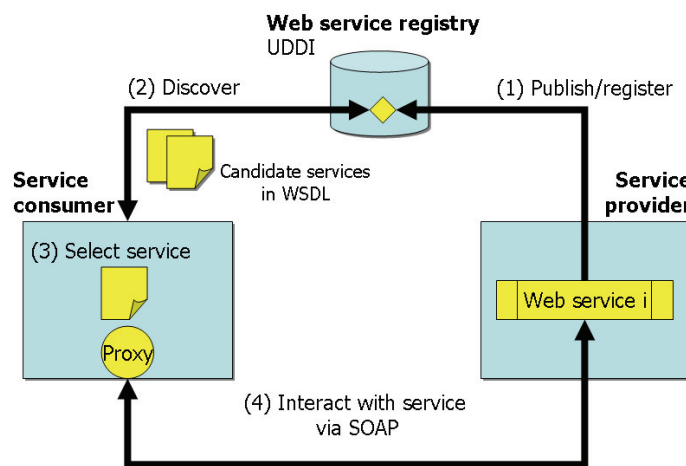


Fig. 1.4. Web service interaction life cycle (SOA model for Web services)

coordination scenario consisting of three core entities:

- A service registry acts as an intermediary between providers and requesters. Most of these directory services categorise services in taxonomies such as UDDI registries.
- A service provider defines a service description and publishes it to the service registry.
- A service requester can use the directory services search capabilities to find (discover) relevant service descriptions and their respective providers.

A service requester agent searches for relevant Web services that are registered (published) by service provider agents at one or multiple Web service registries such as UDDI registries. Upon request, the registry returns (references to WSDL descriptions of) potentially relevant services to the requester agent that retrieves the corresponding WSDL descriptions and selects the most appropriate service. In the last step, the service requester invokes or initiates an interaction with the selected service at runtime via SOAP using the binding details in the WSDL service description to locate, contact and invoke the service. Service selection includes the negotiation of each individual candidate service resulting in the selection of one service and commitment to terms of service delivery (so-called service level agreement).

#### *Web service registries*

Publishing and locating conventional Web services is commonly done by use of UDDI service registries such as Systinet's WASP UDDI v2, where service registries are installed as servlet under Apache Tomcat 3.2.3, WebLogicServer6.1, IBM WebSphere 4.0, or as Java-based UDDI client API. Another alternative is the utilization of IBM's UDDI for Java (UDDI4J, v2 beta) and Apache SOAP 2.1. Please note that, unlike service repositories, service registries do not provide the service code itself but a reference to its interface description only.

The OASIS standard for Web service registries is the Universal Description, Discovery and Integration (UDDI) specification<sup>8</sup>. An UDDI business registry (UBR) is a hierarchically structured, XML-based registry of services, and bindings provided by business entities. It allows to publish services, browse through, and query the set of registered services via APIs using SOAP. Published Web services are uniquely identified, and their types are registered in the UBR as a unique tModel. Each entry in the UBR provides information on the relevant business entity (white pages), business service entity (yellow pages), and technical data on the service (green pages).

One can use two different APIs to access UDDI servers which communicate using SOAP: PublishSOAP for service providers (publish/advertise), and InquireSOAP for service requesters (request). The InquireSOAP interface offers 10 query operations for searching the UDDI registry. Queries to an UBR are regular expressions with keywords using standard taxonomies such as NAICS (North American Industrial Classification System), and SIC (Standard Industrial Classification).

#### *How to use the triplet UDDI/WSDL/SOAP in practice?*

In order to locate relevant services, an agent may search the UDDI registry for relevant services by means of keyword matching through the InquireSOAP API in order to obtain the references to WSDL service descriptions of relevant

---

<sup>8</sup> [www.uddi.org](http://www.uddi.org)

services. It then retrieves and inspects those WSDL description files to know about how to interact with the targeted services using SOAP messaging or RPC calls, and finally invokes the service implementation at the respective provider site.

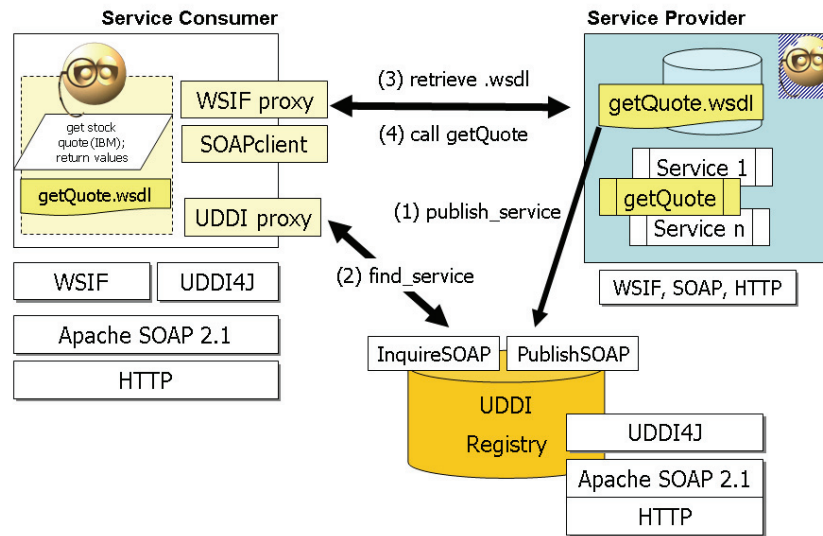


Fig. 1.5. Example of Web service interaction life cycle

An example of implementing this scenario is sketched in figure 1.5. It refers to the discovery, retrieval, and invocation of a Web service in WSDL named `getQuote.wsdl` that returns actual values of given stock quotes from the NYSE (New York Stock Exchange) upon request by a personal agent. This is realized by means of appropriate proxies of software components implementing the basic infrastructure of the W3C interaction life cycle such as proxies of the WSIF (Web Service Invocation Framework), UDDI4J (UDDI for Java), Apache SOAP, and HTTP. These proxies are properly integrated into the agent code that is deliberately initialising and handling the whole interaction, and further processing of returned results.

In particular, the service agent performs a keyword-based search for services capable of returning NYSE stock quotes via the `InquireSOAP` API of the nearest UDDI service registry, selects the service `getQuote.wsdl` from the set of returned results, and retrieves its WSDL description from the relevant provider. By inspection of the WSDL file, the agent gets to know not only about the physical location of the service (URL) at the provider site but the data types of the service I/O messages in XMLS required to interact with the service via SOAP messaging. The personal agent code can be developed

with any software agent development environment such as JADE, FIPA-OS, Tryllian ADK, or JACK.

#### 1.2.4 Web Service Composition

Web service composition refers to the process of combining a number of Web services into one processing entity at a higher abstract level to provide some new functionality. In practice, this task is done manually by assigned experts, so-called business service operators (orchestrators) on the basis of agreed-upon conditions of utilising contracted services provided by known business partners.

Nevertheless, automated service composition has been subject of a few research projects such as the Ninja project [137], the SAHARA project [307], and the OWSBI (Ontology-based Web Services for Business Integration) project from IBM. The latter project implemented a proof-of-concept demonstration for the industrial sector that shows semi-automatic service discovery, composition, and business process transformation<sup>9</sup>. In general, we can distinguish between so-called functional-level composition, that is service signature-oriented ("black-box" composition), and so-called process-level composition, that is based on the control and data flow ("glass-box" composition).

##### *Functional-level Web service composition*

Functional-level WSDL service composition can be achieved by sequentially composing services with properly matching input and output message type structures in XMLS that produce the desired output for a given input. For this purpose, means of XML graph matching with, for example, XPath, XQuery and variants, or statistical means from the domain of information retrieval can be exploited. The same holds, of course, for matching the whole WSDL service description with a given query which can be complemented by non-functional (QoS) parameter matching. Tools that support type matching-based WSDL service matching are rare; one example is the WSDLANalyzer developed in the project ATHENA. Alternatively, WSDL service descriptions can be intuitively mapped to process languages with formal operational semantics. For example, in [254, 150], WSDL services are translated to equivalent coloured Petri nets which composition allows to correctly route and manipulate I/O messages of composed services from one organization to another.

##### *Process-level Web service composition*

Process-level Web service composition is usually split into the so-called orchestration and choreography of WSDL services [292]. A service orchestration model describes both (a) the signature-based ("black-box") interaction between the given set of services, and (b) the control and data flow of their

---

<sup>9</sup> [www.alphaworks.ibm.com/tech/owsbi](http://www.alphaworks.ibm.com/tech/owsbi)

internal ("glass-box") processing. In contrast, a service choreography model delivers only the first part but from a global perspective. It describes the message-oriented communication between services including some control and data flow dependencies, message correlations, time constraints, and transactional dependencies - but no description of internal functionalities of atomic or composite services including binding of service input and output variables. For example, Berardi et al. (2005)[29, 30] propose a process-oriented composition of WSDL services where both the externally observed and the internal process execution behavior of each service are represented in terms of (deterministic) finite state machines - which are then automatically composed to comply with a given service execution scheme.

#### *Service orchestration with BPEL*

Orchestration of Web services in WSDL combines existing Web services by adding a central coordinator (so-called orchestrator) who is responsible of invoking atomic Web services according to a set of abstract control flow patterns. This yields a global process description that can be executed by an orchestration engine. State of the art language for specifying such Web service orchestration is the XML-based business process execution language for Web services (BPEL4WS, in short: BPEL) <sup>10</sup> designed by OASIS members IBM, BEA Systems, Microsoft, SAP AG, and Siebel Systems. BPEL allows to specify the control logic required to aggregate atomic Web services in WSDL participating in a given business process workflow. It combines WSFL (Web Service Flow Language) from IBM and BEA System's WSCI (Web Services Choreography Interface) with Microsoft's XLANG specification.

In BPEL, the result of a Web service composition is called a process, participating services are partners, and message exchange or intermediate result transformation is called an activity. Every BPEL process is exposed as a Web service using WSDL which describes the public entry and exit points for the process, and can be used to reference external services required by the process. XML-based WSDL data types are used within a BPEL process to describe the information that passes between requests, that is a BPEL process interacts with external partner services through a WSDL interface.

Abstract BPEL specifications are interpreted by an orchestration engine such as IBM's BPWS4J and Collaxa ([www.collaxa.com](http://www.collaxa.com)). For this purpose, it is assumed that the control and data-flow dependencies of a composite Web service are maintained and executed by a distinguished node that acts as a central scheduler (usually one of the participating parties). The engine coordinates the various activities in the process, and compensates the system when errors occur. BPEL is essentially a layer on top of WSDL, with WSDL defining the specific operations allowed and BPEL defining how the operations can be executed in sequence or in parallel (cf. figure 1.6).

---

<sup>10</sup> [www-128.ibm.com/developerworks/library/specification/ws-bpel/](http://www-128.ibm.com/developerworks/library/specification/ws-bpel/)



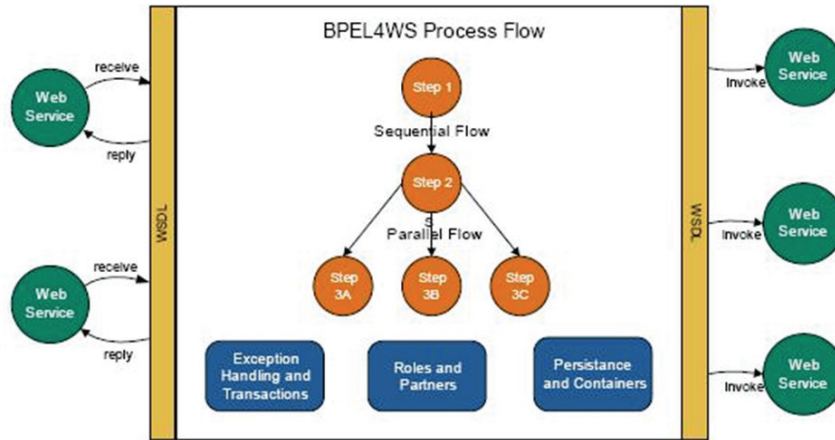


Fig. 1.6. BPEL business service process flow overview [292]

We refrain from providing an example of Web service orchestration in BPEL by means of a concrete (executable) BPEL file, referring to [190] instead.

#### *Service choreography with BPEL or WSCI*

As mentioned above, Web service choreography concerns the description of the observable or visible behavior between Web services involved in the orchestration in terms of their interfacing only. In this sense, it can be both complementary to or even part of the orchestration depending on the level of details of the latter. In practice, orchestration languages like WSCI and WSCDL [25], BPEL is also often used to define the corresponding choreography. The WSCI by SAP, Sun, and Intalio is an XML-based language for defining a purely message-based service collaboration, that is service choreography. It supports message correlation, sequencing rules, exception handling, transactions, and dynamic collaboration. WSCI does not address the definition of executable business processes as defined by BPEL. Furthermore, a single WSCI document only describes one partner's participation in a message exchange. A WSCI choreography includes a set of WSCI documents, one for each partner in the considered interaction, but there is no single controlling process managing the interaction. WSCI is considered part of BPML (business process modelling language) that defines the business processes behind each service. However, to date, BPEL has been adopted by the majority of business stakeholders for both orchestration and choreography (mostly specified in one con-

crete BPEL file in practice) while the alternative WSCI/BPML still has a share in the academic world.

### *Verifying Web service composition in BPEL*

One problem with service orchestration with BPEL in the large is the automated verification of its correctness, that is the reliability of composite Web services. The majority of recently proposed solutions to this problem are based on state-action models (e.g. labelled state-transition systems, several variants of FSM and ASM [22, 311], Petri nets [370, 390]), or process models (e.g.  $\pi$  calculus, calculus of communicating systems, LOTOS, Finite state process FSP, BPE calculus) [322].

Basic idea of verifying BPEL processes at design time is to translate them into one of the aforementioned formal models, and then to use a tool (such as a model checker or workflow analyzer) that verifies standard properties for this specific formal model, or any other desired property expressed in the logic supported by the tool. For example, the prominent model checkers <sup>11</sup> SPIN, SMV, and NuSMV for state machines, and Petri net-based workflow analyzers like LoLA (HU Berlin), Wolfan, or ProM (TU Eindhoven) are used for formal specification and verification of asynchronously communicating Web services[122], and the study and detection of information leakage in Web service compositions [270]. Further, the Concurrency Workbench, CADP (Construction and Analysis of Distributed Processes) [322], and LTSA (Labelled Transition System Analyzer) have been used to verify the formal process models of BPEL, that is to check whether a Web service orchestration designed in BPEL satisfies a given composition process, and vice versa.

Verifying orchestrated Web services at runtime addresses the problem of checking and quantifying how much the actual behavior of a running composite service, as recorded in message logs, conforms to the expected behavior as specified in the underlying process model in BPEL. For example, (van der Aalst et al., 2007)[371] translate BPEL process definitions into Petri nets and apply Petri net-based conformance checking techniques to derive two complementary indicators of conformance (fitness and appropriateness) by means of their toolset for business process analysis and mining, namely ProM, tested in an environment comprising multiple Oracle BPEL servers.

### *Dynamic binding of Web services*

As mentioned above, static orchestration of Web services in WSDL at design time refers to a specification of an executable business process workflow in BPEL. This BPEL script is then typically enacted by a single workflow engine of one of the business parties involved that manages the respective

---

<sup>11</sup> A model checker verifies if a given system model satisfies a desirable property. If the property does not hold, it returns a counter-example of an execution where the property fails.

communication and data flow between the orchestrated services. These patterns of interactions between Web services at the I/O message level usually result in a long-lived, transactional multi-step process model with prescribed execution order within or across enterprise boundaries.

In general, however, inter-organisational service orchestration differs from classical workflow management [388]: Web services need to be dynamically linked to the current process enactment at runtime which overall control is dynamically distributed among business partners involved. In particular, Web services technology promises to enable a service requester to discover, select, and invoke a Web service not only at design time (static) as supported by BPEL but even at runtime, which is often referred to as the process of dynamic binding or integration. This is at the core of most WSDL service composition approaches that employ service registries such as UDDI, or some form of discovery agencies to manage the binding of subsequent Web service interfaces<sup>12</sup>. However, as mentioned above, this requires, in particular, the matching of XMLS service message types with respect to the underlying intended semantics of heterogeneous Web services without knowing these and the order of their invocation precisely and in advance - which goes far beyond BPEL. Recent solutions to this problem rely on the use of appropriate semantic metadata [70], automated semantic reasoning, and rational agents (cf. sect 1.3.4).

### 1.2.5 Service Negotiation and Contracting

In competitive environments, service consumers may get charged for service usage according to a given pricing model of the respective provider. The terms and conditions of service usage are negotiated in so-called service level agreements (SLA) between consumer and providers during a service negotiation phase. Such agreements specify guarantees for (a) the delivery of certain functionalities of configurable services, and (b) the non-functional service qualities (QoS) like the duration of provision, throughput, response time and latency bounds based on mutually agreed measures, the service privacy policies and pricing for the consumer<sup>13</sup>. Any mutually enforceable SLA signed by the selected service providers and consumer is called a legal service contract which subsequently gets initialized and executed.

Automated Web service negotiation in business-to-business protocol standards is only addressed in a very limited way to date. One option is to create SLAs by use of the OASIS standard ebXML Collaboration Protocol Agreement (CPA) from the Collaboration Protocol Profiles (CPPs) of two prospective trading business partners. Another option is to exploit the W3C standard WS-Agreement, that is a XML-based language for representing contracts and

---

<sup>12</sup> Web Services Architecture, 2004: [www.w3.org/TR/ws-arch](http://www.w3.org/TR/ws-arch)

<sup>13</sup> Note that in computer networking literature, the traffic-engineering term SLA is restricted to non-functional service quality guarantees for network service consumers (such as for streaming multimedia applications and videoconferencing).

agreements on one or multiple services between provider and consumer in terms of service level objective (SLO, defines bounds usually over QoS concepts such as response time, fault rate or cost), qualifying conditions (which must exist in order for the SLO to be satisfied) and business values (which represent the strength of commitment by stating penalties, rewards and importance). However, WS-Agreement suffers not only from the lack of an interaction protocol that would allow business partners to form more complex conversations than through a simple 2-step offer-accept message sequence (such as in alternating-offers negotiation) but from vague and unclear semantics of the specification. Oldham et al. (2006)[279] propose an approach to semantic WS-Agreement based matchmaking of providers and consumers. This is achieved by using declarative domain specific (predicate) rules to match syntactically heterogeneous but semantically same SLOs guaranteed by providers and requested by consumers: The SLO of the guarantee should meet or exceed the SLO of the requirement.

In practice, however, most Web service negotiation approaches just return an acceptance or a simple rejection of a service request with desired QoS: Service consumers are provided with the QoS that can be supported at the time the request is made, but there is no adaptation to changing QoS conditions in stochastic, dynamic environments. Prominent service negotiation options are service item auctions, and comparative bargaining (based on, for example, multi-attribute utility theory and recommendation) to select services with the best ratio between performance and price in both short and long term. Alternatively, cooperative and rational provider agents can form coalitions in order to maximize their individual profits by sharing and combining user-charging services on demand. For more details, we refer to part four of this work.

Web services are not supposed to negotiate any terms of their usage. Thus, most service (QoS) negotiation protocols like [214, 60] use agents for this purpose. In the next section, we take a closer look at the nature of the relationship between Web services and rational agents.

### 1.3 Web Services and Rational Agents

According to Singh and Huhns (2005)[335], rational software agents play a major role in service-oriented computing. In fact, agents are considered key for intelligent coordination and provision of complex Web services to the individual user in a given environment and application context.

#### 1.3.1 Rational Agency

In this work, we do not introduce a new definition of the term "rational agent" but refer to those available in the fields of AI (Russell & Norvig, 2002)[313], and autonomous agents and multi-agent systems (Weiss, 1999;

Wooldridge, 2000)[380, 385] each of which emphasizing different aspects of rational agency. In general, an intelligent software agent is said to be rational if it chooses to perform actions that are in its own best interests given the beliefs it has about the world or environment it is situated in. Such rational agents should have the properties of autonomy; pro-activeness; reactivity; and social ability (Wooldridge & Jennings, 1995)[384]. That is, rational agents are supposed to be capable of making independent, rational decisions, identifying and taking goal-driven initiatives, adapting to changes in the environment, and collaborating with other agents to accomplish its individual or joint goals, if required.

Logic-based qualitative approaches to rational decision making encompass the processes of deliberation and means-end reasoning about what, respectively, how an agent can achieve its states of affairs, such as exemplified in the BDI logic framework by Rao and Georgeff. Alternatively, decision and game theory are normative and quantitative theories of rational action that define a rational agent as one that maximizes expected utility of actions it performs without actually saying how to obtain the respective utility function or probability distribution. However, game-theoretic approaches for strategic negotiation between multiple rational agents with conflicting individual goals have found many applications in multiagent systems research. We present game-theoretic protocols that allow rational service provider agents to form profitable coalitions in different environments in part four of this work.

### 1.3.2 Relation Between Service and Agent

Unfortunately, the terms "agent" and "service" are still often used interchangeably in the literature (Huhns, 2002)[169]. We argue that services are not agents: Web services remain passive until invoked, whereas rational agents are not. In particular, Web services are not supposed to take any kind of initiative to deliberately deviate from their hard coded application-centric functionality, neither individually, nor in joint collaboration with other services - whereas agents can do.

Autonomous agents are capable of pro-actively searching for, composing, and negotiating services on behalf of its user, individually or in cooperation with other agents. Besides, unlike agents, Web services are typically not supposed to negotiate their usage terms with potential customers, nor to flexibly adapt to stochastic changes in the environment apart from some kind of trivial user profiling. On the other hand, what service-oriented computing adds to the multiagent systems world is the ability to build on conventional IT and do so in a standardized manner to facilitate the practical development of large-scale, interoperable systems (Singh & Huhns, 2005)[335].

To sum up, what benefits can we expect from distinguishing between services and agents? The utility of passive Web services can be extended with autonomous control, reactiveness, and proactiveness which are essential characteristics of rational agents. Service providing agents can deliberately and

pro-actively respond to changes in the application environment. This promises to leverage personalization and effectivity of service provisioning to the human user. Some of the design choices for integrating agents and Web services are reviewed in (Dickinson & Wooldridge, 2005)[98] together with an approach to control Web service invocation by BDI agents using reactive planning.

### 1.3.3 Types of Service-Agent Interaction

Given that both services and agents are different in nature, how can they interact with each other? We distinguish between the following basic types of service-agent interaction as indicated in figure 1.7. A similar classification is proposed in (Müller et al., 2005)[267].

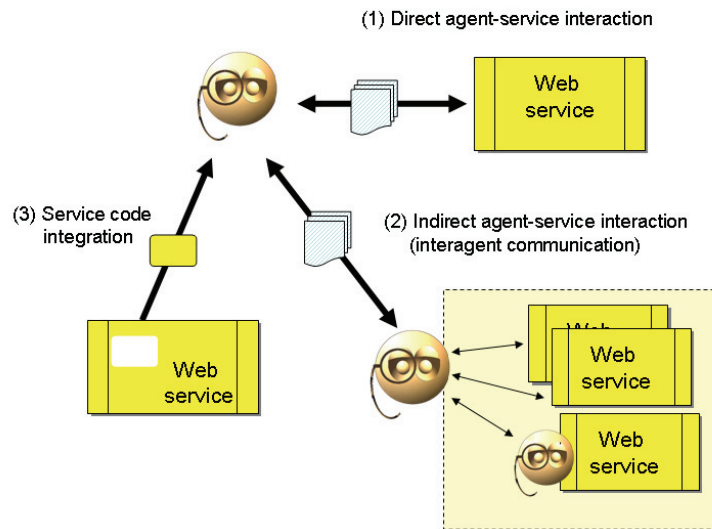


Fig. 1.7. Basic types of service-agent interaction

#### *Direct service-agent interaction*

In the first scenario, any service requester agent directly interacts with one or more Web services by means of appropriate SOAP messaging. The agent invokes the service and processes the returned results. The opposite way of an agent being called by a Web service violates, in principle, the agent autonomy as its exposed capabilities would be required to represent fixed, deterministic behaviours to the service. Besides, the communication idioms of

Web services and FIPA-standard agents are distinct: (a) Procedure calling Vs. plain message-passing, (b) asynchronous Vs. asynchronous/synchronous, (c) stateless Vs. stateful interaction. In any case, unlike Web service interaction, FIPA-standard based inter-agent communication allows a clear separation of the communicative intent (request, assertion) from the application-oriented content of messages which, in particular, eases query-response tracking and adaptation of messaging to changes of the domain.

In order to properly interact with any Web service, an agent has to know the semantics of exchanged message content returned by the service, as well as about exception handling, dealing with lost messages, messages out of sequence, time-outs, and so on. Such knowledge can be acquired by the agent either by means of prior hand-shake coordination with the service, or during its interaction with the service based on a minimal shared ontology and appropriate means of ontology matching, ontology learning, and information integration. Any use of different service-agent interaction patterns that an agent wants to follow such as temporary subscription to a service are not supported by Web service related standards and protocols, hence have to be agreed upon between the requesting consumer agent and the service provider in prior.

Similarly, no agent can flexibly negotiate service usage terms for its users with a passive service that exclusively follows a restrictive set of constraints, acceptable choices, and policies given by the provider once and for all. Partial solutions are, for example, auction services like ebay where the service itself constitutes the negotiation platform for mediation between providers and customers, or more complex virtual e-market platforms in the B2B or B2C domain that allow customer agents to negotiate access to relevant agent-based business services which would imply a different basic type of service-agent interaction as follows.

#### *Indirect service-agent interaction*

In the second scenario of service-agent interaction, one or more Web services are encapsulated (or wrapped) and presented to the external world by a special type of information agent, a so-called service agent<sup>14</sup>. As a consequence, any service requester agent has to converse with relevant service provider agents in an agreed-upon ACL with content language, domain ontology, and following a given coordination protocol. This, in particular, complies to the paradigm of agent-based cooperative information systems (Papazoglou & Schlageter, 1998)[288] such that existing solutions to semantic interoperability, and fault-tolerant, distributed service transactions can be reused for developing agent-based Web service systems with this type of interaction.

---

<sup>14</sup> We use the term service agent interchangeably for both terms service requester agent and service provider agent, as well as for the terms rational service agent and rational agent if the semantics in the given context are clear.

Prominent example of a gateway between a Web service and a FIPA agent is the WSDL2JADE tool [274] that allows to translate XML-based SOAP messages of WSDL services via Java (Axis JAX-RPC and JADE ontology package) to ACL (for content language SL). The Web service (operation) interaction patterns request-response, solicit-response and subscribe-notification are mapped to the FIPA request, respectively subscribe interaction protocol. WSDL2JADE is used by the WSDL2Agent tool [369] to generate a proxy agent that represents the WSDL service as a kind of syntactic "wrapper agent" to other rational agents via FIPA ACL messaging interface only. These simple wrappers can be extended to rational agents depending on the agent-based service application at hand.

#### *Service-agent integration*

In the extreme, intelligent agents may even integrate relevant (parts of) available Web service code into their own agent code. This would inherently change their behaviour at any time as proposed in (Bryson, 2003)[51]. In the remainder of this work, however, we presume the first two types of service-agent interactions.

#### *Examples of agent-based Web services*

There are quite a few agent-based Web or semantic Web service applications available, though most of them are not accessible outside private research project repositories. Some of them are following in part the SOA paradigm like AgentSteel for agent-based steel production control deployed and running at the steel manufacturer Saerstahl AG since 2005[176], or iPARK for inter-enterprise services integration [70]. In part five of this work, we present selected examples of agent coordinated business service systems in general, and agent-based semantic Web service coordination for the e-health domain in particular.

### **1.3.4 Agent-Based Web Service Coordination in Brief**

Why are intelligent agents particularly well suited to perform automated Web service coordination? The general answer to this question is that passive Web services simply cannot automatically coordinate themselves but agents can do, in a proactive way. In other words, agent-based service coordination aims at flexibly ensuring the collective functionality and expected higher end-to-end quality of services proactively provisioned to the human user in dynamic environments.

The building blocks of agent-based service coordination are service discovery, service composition, service negotiation, and service execution as shown in figure 1.8. The typical sequence of coordination activities (in compliance with the SOA model) starts with the discovery and composition of candidate services that are relevant to a given request, the final selection of services by negotiating their terms and conditions with respective providers, and ends with the execution of contracted services.



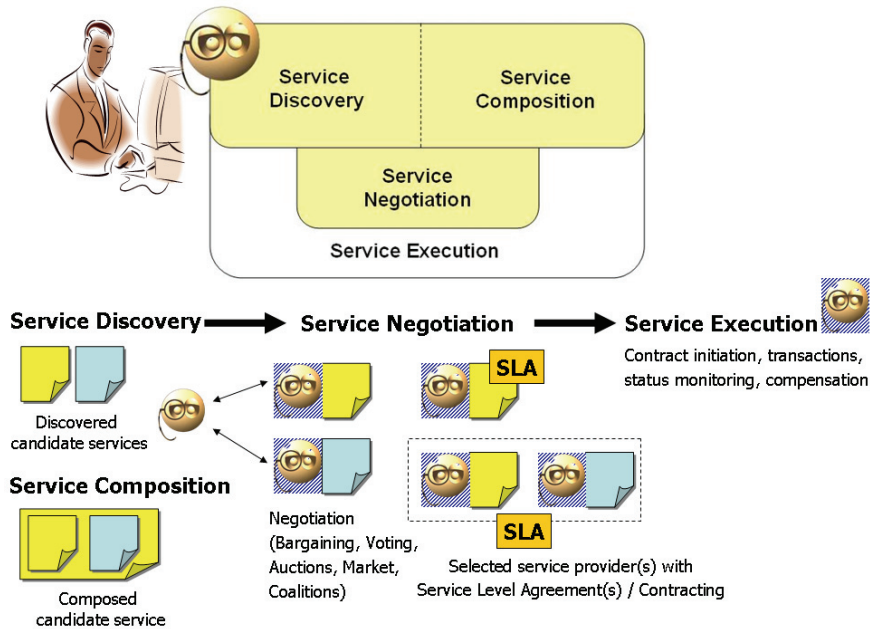


Fig. 1.8. Building blocks of agent-based service coordination.

*Agent-based Web service discovery and composition*

Like intermediaries in the physical economy, so-called middle-agents, are able to solve the coordination or connection problem in the Internet, that is to connect the ultimate service requester with the ultimate service provider, in different ways based on the declarative characterization of the capabilities of both. Prominent types of middle-agents are broker, matchmaker, and mediator agents (Klusch & Sycara, 2001)[212].

In particular, the W3C Web service interaction life cycle or conceptual SOA model (cf. figure 1.4) corresponds to the classical matchmaking process. In fact, like a registry, a matchmaker agent returns a ranked list of registered services that semantically match a given query to the requester, whereas a broker agent additionally handles service engagement (negotiation and legal contracting) as well as transactions of service value which, in most cases, bases on subscription models (with differentiation-based or flat-fee pricing).

Mediator agent-based SOAs like the one proposed in (Cong, Hunt & Dittrich, 2006)[70] draw upon the notion of mediators introduced by Wiederhold and his colleagues in the domain of multidatabase systems. That is, a central service mediator agent is capable of semi-automated integration (composition) and enactment of Web services based on appropriate service interface schema alignment, and coordinated execution. However, as mentioned above, automated service composition (orchestration in BPEL) is still unusual in the

business domain in practice. An approach to agent supported composition and reliable execution of Web services is proposed, for example, in (Binder et al., 2004)[36]. Other approaches are proposed in (Chakraborty & Joshi, 2001)[61] for agent-based Web service composition in wired-line networks, and in (Chen et al., 2001)[64] for mobile service networks.

#### *Agent-based Web service negotiation*

As mentioned above, in commercialized and competitive settings, services may not be available for free but pay per use. Service requester agents could be charged, for example, for every single invocation of services at discovery or planning time according to selected flat fee or differentiation-based pricing models. In most cases, service pricing for both configurable and non-configurable services based on quality of service parameters such as latency, delivery time, warranty, and the opting out of privacy policy parts, is subject to negotiation. Besides, service pricing is often private which makes it hard, if not infeasible, for any requester agent to calculate the total expenses of its coordinated service value provision to its user in advance.

Self-interested requester and provider agents can (semi-)automatically negotiate the terms and conditions of using individual or compound candidate services. The candidate services have been selected according to their semantic relevance to the given query (service selection as part of service discovery). There is a wide range of negotiation protocols (mechanisms) that can be used by agents for service negotiation such as voting, contract nets, auctions, bargaining, general equilibrium market mechanisms, and coalition forming (Rosenschein et al., 1994; Sandholm & Lesser, 1995; Kraus et al., 2001)[319, 219, 327]. In any case, all agents have to agree in advance on the used negotiation protocol, procedure of contracting and its scope of enforcement including terms and condition of penalty payments, the general payment scheme, XML message templates and vocabulary (XMLS namespaces) for (semantically) interoperable interaction.

After reaching a service level agreement at the end of the negotiation phase, the SLA gets transformed into a legally binding contract which has to be signed by both service providers and requesters (contracting). This contracting phase is separated from but often considered being part of service negotiation. In practice, Web service orchestration is done manually based on SLAs that have been negotiated and contracted with business partners even before or during service orchestration.

Representative examples of agent-based service negotiation are listed in the introduction to part four of this work. There we also present our game-theoretic solutions to the particular problem of privacy preserving service negotiation in n-agent coalitions. In particular, these coalition algorithms enable agents to negotiate individually rational distributions of joint payoffs obtained by means of joint sales of service items in a coalition. Hence, the application of these algorithms is agnostic to any (semantic) Web service description format and the common understanding between all negotiating parties is assumed.

## 1.4 Critique

There are quite a few basic problems with Web services which we briefly summarize in the following.

*What are the semantics of WSDL services?*

Although Web service technologies can simplify SOAs drastically, one major barrier of reaching the full potential of automated and meaningful service coordination is that they are exclusively syntactical and lack any formal semantics. In fact, the use of XML allows to uniformly represent all kinds of data in the Web including unstructured text and video streams, structured data like account records, and mixed data like annotated text. Together with the separation of XML data (.xml) and its possibly multifolded interpretation over corresponding type schemas (.dtd), and agreed upon namespaces, XML supports business data interoperability and software mass customization in practice. However, the crucial point is that XML has no formal logic-based semantics, much less XML-based service descriptions in WSDL.

*How to meaningfully reason on WSDL services?*

As one consequence, from the hard core (symbolic) AI perspective, rational agents cannot meaningfully reason about Web services while pursuing their tasks in the Web. That is, the semantics of Web services are not specified in any logic typically used by deliberative agents to represent and reason upon its knowledge about the world including services. There are lots of agreed upon but no formally grounded (domain) namespaces in XML available for service descriptions in WSDL.

More concrete, a service description in WSDL specifies how to technically interact with the service but hardly reveals what it does, nor in what order its operations have to be invoked to achieve certain functionalities. The only way the current W3C Web services framework allows to cope with these problems is to look at respective entries, or informal comments of the WSDL description, or in the service registry which are provided, if at all, in natural language text by humans. UDDI service registries only allow for simple keyword-based searching of relevant services; the same holds for Java-based services in JINI service networks. Though agents can easily search UDDI registries through the respective SOAPInquireAPI, humans still must be involved to semantically interpret the entries returned.

*Scalable Web service orchestration?*

Since heterogeneous WSDL services are encapsulated in BPEL orchestrations, data mismatches must be addressed through a process that transforms data types and models without loss in service semantics. On a large scale, to model these processes as services could dramatically affect the throughput of the

overall system. An account for not applying service concepts to runtime structures but applying high-performance transaction system design criteria that optimize runtime properties instead is given in [52]. There are no experimental evaluation results for BPEL orchestrations available so far which makes it hard to judge even on its scalability.

#### *Automated planning for Web service composition?*

Automating Web service composition could be based on existing AI planning techniques, although these methods were developed for problems where the number of operators is relatively small, but may lead to complex plans. In contrast, Web service composition for large scale pervasive computing environments requires planning methods that can deal with the very large number of possible services, but plans are not likely to become very complex. In any case, composition planning based on AI planning techniques requires a formal logic-based grounding of service semantics which is lacking for WSDL services. As mentioned above, most approaches to WSDL service composition, in essence, base on XML message type compatibility checking.

The vision of agent-based semantically enriched SOA, in particular semantic Web service coordination promises to overcome most of the above problems by exploiting semantic Web technology. In the next chapters, we provide a brief introduction to the semantic Web, semantic Web services, and their automated coordination. The following parts of this work are then specifically dedicated to agent-based semantic Web service discovery, composition planning, and negotiation.

## 1.5 Further Readings

For a more comprehensive treatment of service-oriented computing and architectures, we refer to, for example, the excellent book of Singh and Huhns (2005)[335], the proceedings of major conferences in the field such as the international conference on service-oriented computing (ICSOC), and relevant research projects like SUPER<sup>15</sup> and ATHENA<sup>16</sup>. A most recent survey of service-oriented computing is provided in (Papazoglou et al., 2007)[289]. For more complete coverage of agent-based SOAs, we refer to relevant research projects like the European integrated project ATHENA<sup>17</sup>, and the proceedings of international workshops on Web services, or service-oriented computing and agent-based engineering (WSABE, SOCABE)<sup>18</sup>. (Omicini et

---

<sup>15</sup> [www.ip-super.org](http://www.ip-super.org)

<sup>16</sup> [www.athena-ip.org](http://www.athena-ip.org)

<sup>17</sup> [www.athena-ip.org](http://www.athena-ip.org)

<sup>18</sup> [www.ict.swin.edu.au/conferences/socabe2006/](http://www.ict.swin.edu.au/conferences/socabe2006/)

al., 2001)[280] provide an accessible account of agent-based coordination technologies including a special treatment of middle agents such as brokers and matchmakers (Klusch & Sycara, 2001)[212].



## Semantic Web

The vision of the Semantic Web is that of an extension of the current Web in which the individual meaning of uniquely identified resources is "understandable" not only by human users but also by machines or software agents.<sup>1</sup> The basic idea to realize this vision is to annotate Web resources with machine-interpretable meta-data that is exclusively based on shared vocabularies or ontologies written in a logic-based ontology language. This way, intelligent agents in the Semantic Web can automatically reason about the resource semantics, and, as one consequence, improve upon the quality and the way of how they pursue their tasks on behalf of their users.

In particular, the tremendous amount of information available in the Web should be processed by agents based on formal semantics attached to it. That is, the semantics of any Web resource as seen by the human user shall be encoded in such a way that any agent should not only be able to process, but "understand" it by drawing similar conclusions on it through appropriate kinds of logic reasoning than their human users would do in a given context as good as it gets. This is in perfect line with the weak, if not strong AI paradigm based on symbolic knowledge representation. The Semantic Web community has developed a number of W3C standard languages (RDF, RDF Schema, OWL) that deploy logic for this purpose. It is expected that proactive agents in the Semantic Web will significantly improve on both the way and quality of task-oriented data and information discovery, composition, navigation, and provision to their human user - though this is still out of reach for the common user of the Web to date, despite encouraging progress made in the field and impressive single project results in the past decade.

In the following, we briefly introduce main concepts and selected issues of Semantic Web technology, deployed Semantic Web applications, and provide references to further readings throughout and at the end of this chapter.

---

<sup>1</sup> This vision has been advocated first by the director of the Web consortium (W3C), Sir Tim Berners-Lee, at the XML conference in 2000.

## 2.1 Architecture

The key constituents of the Semantic Web are formal ontologies and rules, logic-based reasoning, and proactive agents. Formal ontologies and rules are used to represent static declarative knowledge for semantic annotations of uniquely identifiable Web resources. Rational agents are supposed to proactively reason upon these resources with formal semantics in due course of accomplishing their tasks individually, or in joint collaboration with other agents. The layered functional architecture of the Semantic Web (also called the Semantic Web layer cake) currently proposed by the Web consortium W3C<sup>2</sup> is shown in figure 2.1.

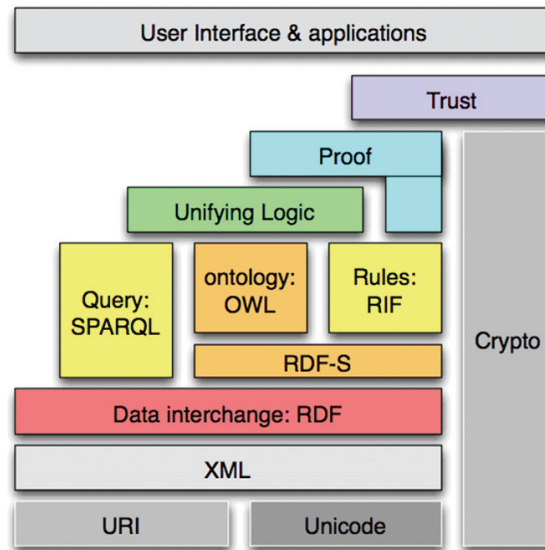


Fig. 2.1. Layered Semantic Web architecture

The Semantic Web architecture grounds itself on standards for referring to entities (URI) and encoding of character symbols (Unicode), and reuses existing Web technologies like XML for syntactic purposes. In particular, its core layers concern the use of standards for improved Web data interoperability and semantic annotation through RDF, respectively, formal ontology languages like RDFS and OWL, and logic-based reasoning and querying.

<sup>2</sup> Original architecture: <http://www.w3.org/2005/Talks/0511-keynote-tbl/>



### *Ontology layer*

The ontology layer focuses on formal knowledge representation and reasoning about the concepts of a domain of discourse and their relationships (ontology) used to semantically annotate the content of Web resources. The standard ontology languages RDFS and OWL (Lite and DL) are decidable subsets of monotonic FOL for which sound and complete proof systems exist. In particular, OWL bases on description logics with reasonable trade off between computational complexity and expressivity required for resource annotation (cf. sections 2.3, 2.4.2).

### *Rules layer*

The original intention of the rules layer is to leverage both the expressivity of ontology languages and practical reasoning upon large scale ontologies by the integrated use of (monotonic and nonmonotonic) logical rules, in particular the efficient proof systems for rule bases in logic programming (LP). As pointed out by Grosz (2003), building rules specified in rule markup languages like RuleML variants [43] on top of ontologies would enable the rule base to have access to ontological definitions for vocabulary primitives (e.g., predicates and individual constants) used by these rules. On the other hand, building ontologies on top of rules enables ontological definitions to be supplemented by rules, or imported into these definitions from rules.

In the current Semantic Web architecture both ontology and rules layer are separated with safe integration either under first-order semantics like in the decidable FOL fragments with monotonic (function- and negation-free) Horn rules, that are DLP (cf. section 2.5.4), Horn-SHIQ (cf. section 2.5.6) and DL-safe SWRL (cf. section 2.5.5), or under nonmonotonic (stable model, answer set) semantics of full logic programming with default negation like in WSMRule (cf. section 2.4.3), DL+log and dl-programs (cf. section 2.5.8).

### *Unifying logic*

An unifying logic on top of the ontology and rules layer is supposed to combine both nonmonotonic and monotonic features in the same language like in (auto-)epistemic extensions of first-order description logics with local closed-world assumption (e.g., FOAEL, ALCK, MKNF [263], OWL-Flight). The local closed-world assumption allows to specify complete knowledge for a certain predicate (DL concept or role) in queries and definitions. For any locally closed and default negated predicate it is up to the implementation to decide which type of nonmonotonic negation to use (e.g. stratified negation, negation under well-founded or stable model semantics).

### *Proof and trust layers*

The final top layers of the Semantic Web cake ensure the secure and trusted exchange of such information between intelligent agents. Basic idea is to let

the agents validate the formal proofs of received logic-based statements on Web resources made in a local or shared context of representation and reasoning. Both, proof and context, are assumed to be provided by the creators of these statements - which authentication is ensured by means of, for example, digital signatures. Once any pair of conflicting statements can be deduced from received information signed with a given key, then anything can be deduced such that the key can be considered broken. However, both layers are just starting to be addressed by Semantic Web research.

According to the W3C, in particular its current director Sir Tim Berners-Lee, reasoning in the open-ended Semantic Web shall be monotonic, though there are strong arguments from the side of practical applications for the need of nonmonotonic (closed-world) reasoning. In fact, the widely accepted compromise is to allow for a kind of local closed-world reasoning (cf. section 2.5).

## 2.2 Ontologies

Logic-based ontologies form the backbone of the Semantic Web and are commonly considered as the silver bullet for many different application areas such as knowledge management, enterprise application integration, e-commerce and e-government systems, Semantic Web service coordination, or social networks in the Web 2.0. This section introduces to the field of ontologies only in very brief with selected references for further readings.

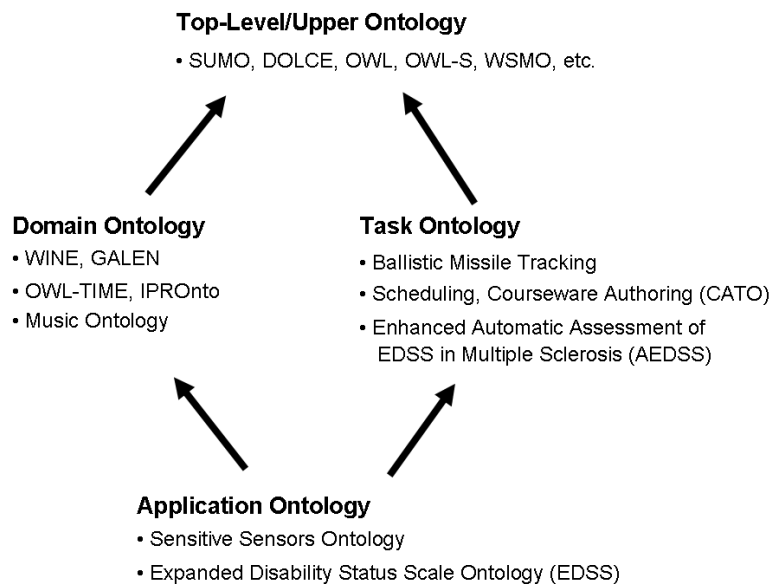
### 2.2.1 Classification

An often cited definition of an ontology by Tom Gruber (1993) [144] defines an ontology as a formal explicit specification of a shared conceptualization of a domain of interest. That is, an ontology explicitly represents a formal model of a shared domain in the real world such that several parties can agree on and reuse it in their own applications.

This definition is not bound to any particular formalism. In fact, there exist many different types of ontologies with varying degrees of formal knowledge representation such as the generic thesaurus WordNet for structured, natural-language descriptions of the semantics of English terms, the vast CyC ontology capturing commonsense knowledge in logic axioms, and other domain-specific ontologies like the UNSPC for product classification scheme for vendors.

#### *Types of ontologies*

According to Guarino (1997)[145], any ontology can be classified with respect to its subject of conceptualization as indicated by the ontology type inclusion hierarchy in figure 2.2.



**Fig. 2.2.** Types of ontologies with examples ([145])

Top-level ontologies describe very abstract and general concepts to be shared across multiple domains and are as such not directly used in applications but for other (domain, task, application) ontologies to be aligned with. Domain ontologies capture (task independent) knowledge within a specific domain of discourse such as genetics, medicine, biology, music, or geography, while task ontologies explicitly refer to a specific task such as diagnosing, configuring, or pricing described neutrally with respect to a domain. Application ontologies are most narrow in scope by providing a vocabulary required to describe a specific task in a specific domain in a particular application; it typically makes use of relevant domain and task ontologies.

#### *How to build ontologies?*

The building (or acquisition) of ontologies is known to be hard in practice, since it requires to reach a consensus among different domain experts, developers, and users on the modeling of the considered task, application domain and/or commonsense knowledge to be effectively shared and reused in a given context. Ontology acquisition can be partially supported by automated means of ontology learning.

#### *Semantic Web ontology languages*

Prominent ontology languages for the Semantic Web are the standards RDF/RDF Schema, OWL (Ontology Web Language), and the non-standard WSML lan-

guage variants each of which having a formal semantics that allows to automatically reason upon the respective ontologies. The same holds for ontologies that are directly specified in first-order logic (FOL), and KIF or any LP variant. We present the standard ontology languages in more detail later.

#### *Examples of Semantic Web ontologies*

Prominent examples of ontologies are the top-level ontologies

- SUMO<sup>3</sup> in FOL/KIF and OWL,
- DOLCE<sup>4</sup> in KIF, RDFS and OWL-Lite (and alignment with WordNet noun sets) for linguistics and cognitive sciences,

and the domain ontologies

- WINE<sup>5</sup> for winery (in OWL),
- GALEN<sup>6</sup> for biomedics (in OWL),
- IPRonto<sup>7</sup> for intellectual property rights (in OWL),
- OWL-TIME<sup>8</sup> and its extension OWL-TIMELINE<sup>9</sup> for temporal concepts (in OWL), and
- Music Ontology<sup>10</sup> for music concepts (in RDFS).

Examples of task and application ontologies are

- the courseware authoring task ontology CATO, specific scheduling task ontologies,
- the expanded disability status scale ontology (EDSS) used by the task ontology for the automatic assessment of EDSS in Multiple Sclerosis (AEDSS) in OCML (Operational Conceptual Modeling Language), and
- the non-public sensitive sensor application ontology (in Prolog) used by the ballistic missile tracking task ontology (in Prolog) of the US Missile Defense Agency.
- the emergency medical assistance ontology used in the research projects CASCOM and SCALLOPS,
- the wealth of other special ontologies developed for application use case demonstrators like in SmartWeb, and the projects of the ESSI cluster.

---

<sup>3</sup> [ontology.teknowledge.com/](http://ontology.teknowledge.com/)

<sup>4</sup> [www.loa-cnr.it/DOLCE.html](http://www.loa-cnr.it/DOLCE.html)

<sup>5</sup> [www.schemaweb.info/schema/SchemaDetails.aspx?id=62](http://www.schemaweb.info/schema/SchemaDetails.aspx?id=62); refers to [ontolingua.stanford.edu/doc/chimaera/ontologies/wines.daml](http://ontolingua.stanford.edu/doc/chimaera/ontologies/wines.daml).

<sup>6</sup> [www.cs.man.ac.uk/-rector/ontologies/simple-top-bnio/](http://www.cs.man.ac.uk/-rector/ontologies/simple-top-bnio/)

<sup>7</sup> [dmag.upf.edu/ontologies/ipronto/](http://dmag.upf.edu/ontologies/ipronto/).

<sup>8</sup> [www.w3.org/TR/owl-time/](http://www.w3.org/TR/owl-time/).

<sup>9</sup> [purl.org/NET/c4dm/timeline.owl](http://purl.org/NET/c4dm/timeline.owl)

<sup>10</sup> [pingthesemanticweb.com/ontology/mo/musicontology.rdfs](http://pingthesemanticweb.com/ontology/mo/musicontology.rdfs).

### 2.2.2 Ontology Alignment

Ontology-based semantic interoperation among heterogeneous resources including services facilitates agents to successfully pursue a variety of different tasks such as service discovery and composition in the Semantic Web. This can be achieved through the common share, reuse or dynamic alignment or matching of relevant parts of different ontologies and rules that are used to describe the semantics of these resources including services.

Many of the current systems for semi-automated ontology alignment like PROMPT, FCA-Merge, FOAM, SAMBO, Chimaera and ODEMerge are based on the computation of similarity values between entites in the given pair of source ontologies, and can be seen as instantiations of the alignment framework defined in (Lambrich & Tan, 2006)[226] which is shown in figure 2.3.

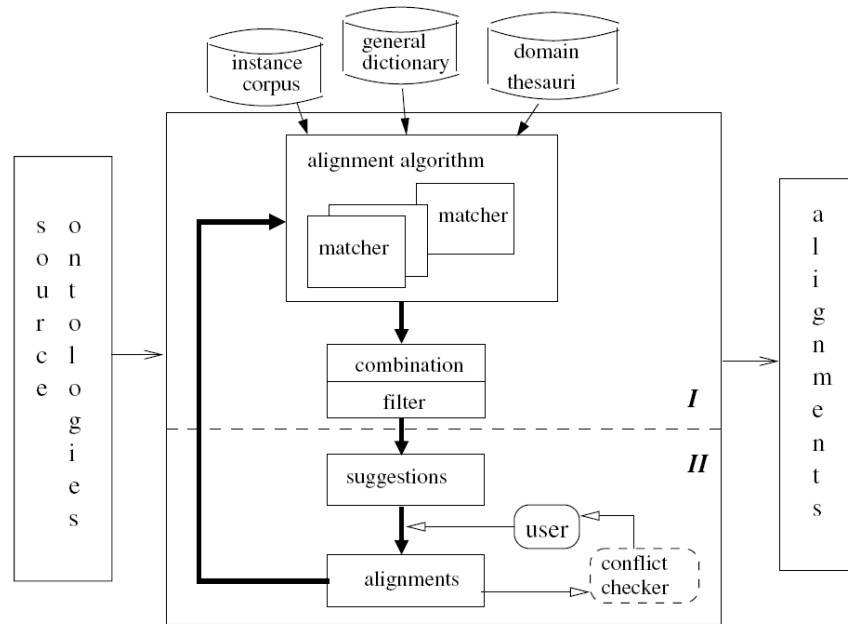


Fig. 2.3. Ontology alignment framework (Tan & Lambrich, 2006)

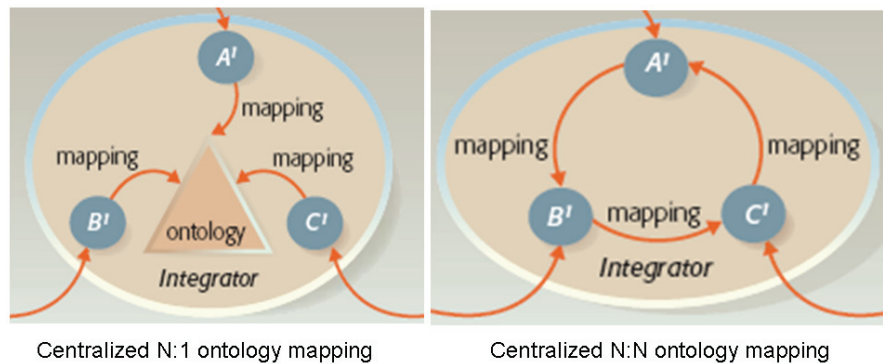
The first part of this framework computes alignment suggestions while the second part interacts with the user to decide on the final alignments. An alignment algorithm receives as input two source ontologies and can invoke several ontology matchers to resolve potential semantic heterogeneities between considered entities. There are many different ontology matching approaches available from relevant fields like knowledge management, databases

[303], and AI. Each of these matching techniques determines semantic correspondences between entities in a given pair of ontologies based on linguistic (syntactic) concept similarity, structure-based matching, logical constraint or instance-based matching strategies, or a combination of these.

Alignment suggestions are then determined by combining and filtering the results generated by one or more matchers. Different ontology alignment strategies are obtained by using different matchers, combining and filtering their results in different ways. The alignment suggestions are presented to the user who accepts or rejects them, which may influence further suggestions. Finally, a conflict checker is used to avoid conflicts introduced by the alignment relationships. The output of the alignment algorithm is a set of alignment relationships between terms from the source ontologies.

### 2.2.3 Ontology Aignment Scenarios

Scenarios of ontology alignment can be classified with respect to the degree of centralization required to implement them (cf. figure 2.4, 2.5) [373] ranging from fully centralized to peer to peer computing environments with respective local interontology mappings.



**Fig. 2.4.** Centralized ontology alignment scenarios [373]

Of course, it depends on the application context which of these ontology alignment scenarios and strategies is most appropriate. A method for recommending ontology alignment strategies with four linguistic matchers for given application is proposed in (Tan & Lambrix, 2007)[227]

#### *Further readings*

A comprehensive introduction to the field of ontologies is provided in the excellent volume [347] and the book chapter [142]. For a survey of ontology match-

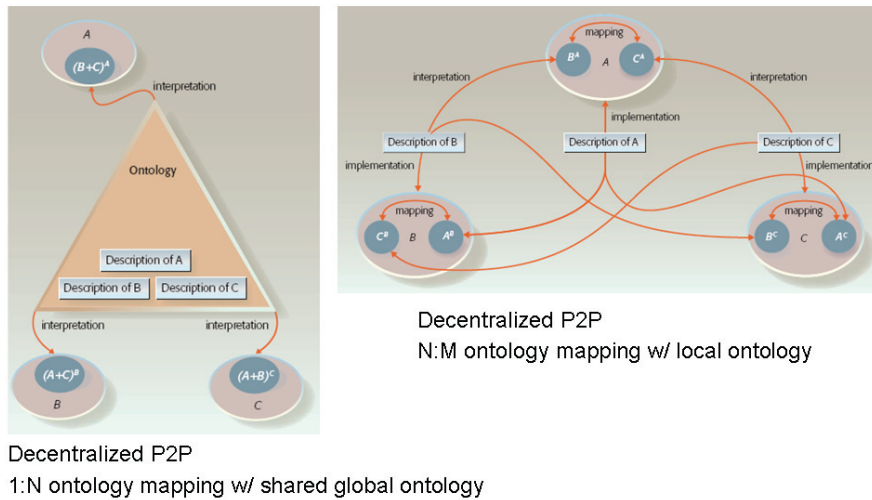


Fig. 2.5. Decentralized ontology alignment scenarios [373]

ing techniques we refer to, for example, [334]. The portal [www.ontologymatching.org](http://www.ontologymatching.org) provides a good source of further references on the same subject.

The majority of ontology languages for the Semantic Web bases on description logics. Therefore, we shall provide a brief introduction to this field in the next section, and then discuss the standard RDFS and OWL family together with the non-standard WSML language variants, and their integration with rules in subsequent sections.

## 2.3 Description Logics

Description logics (DL) are a family of decidable fragments of undecidable first-order logic (FOL)[68] whose trade-offs between expressivity and computational complexity are well known. In this section, we only introduce the main notions and concepts of DL in very brief.

### 2.3.1 Syntax and Semantics

DLs distinguish between atomic concepts or classes, representing sets of objects, and roles or properties, representing relationships between objects, and individuals as specific objects. Atomic concepts (and roles) are canonically constructed to complex concepts (and roles) by means of given concept (and role) constructors of the terminological knowledge representation language

<sup>11</sup>, and a set of primitive components which semantics are assumed common knowledge, hence are not defined in a DL knowledge base.

A DL knowledge base  $W = (T, A)$  consists of a terminology (TBox  $T$ ), and its conservative extension (ABox  $A$ ). The terminology describes the structure of the considered part of the real world by means of both concept and role axioms, while its instantiation is a set of object assertions. To this end,  $T$  can be seen as database schema, while  $A$  would correspond to one database state with respect to  $T$ . There are many linear state-equivalent translation procedures for conceptual data models (like the extended entity-relationship data model [191]) available <sup>12</sup>.

Usually, the semantics of a DL is given by a model theory through its translation to 2-variables FOL fragment with equality and set interpretation over a given domain of discourse. Figure 2.3.1 shows syntax and model-theoretic semantics of some classical DL operations and axioms.

### *Shared minimal vocabularies*

Any complex concept and role between concepts in the TBox of a DL-based ontology is canonically defined from a given set of primitive concept and role components which semantics are assumed to be common knowledge, hence are left undefined. This set of primitive components of an ontology  $T$  is called the basic minimal vocabulary (BMV) of  $T$ .

The common sharing of BMVs out of which different domain ontologies can be canonically built (in the same ontology language) enables any receiver of some concept  $C$  that has been terminologically unfolded in the local ontology of the sender to compute the concept subsumption relation of  $C$  with any concept unfolded in its own ontology. This form of basic semantic interoperability has already been applied to, for example, semi-automated recognition of inter-database dependencies and database schema integration in the early 1990s. However, nothing falls from heaven. Even if we assume such kind of shared BMVs to be exchanged they still have to be aligned to each other (by means of ontology matching techniques), as we cannot presume the existence of one global BMV.

### **2.3.2 Relation to FOL**

The correspondence between FOL and its decidable fragment of function-free DL is indicated in figure 2.7.

---

<sup>11</sup> In the following, the terms role and property, and class and concept are used interchangeably.

<sup>12</sup> However, the open-world semantics of DLs imply that  $A$  contains incomplete information such that reasoning over  $W$  must take all of its models into account whereas closed-world reasoning like in databases exclusively relies on the specific model of  $A$  wrt.  $T$ .



Name	DL Syntax	FOL Semantics	Abbr.
Thing, Nothing	$\top, \perp$	$\Delta^I, \emptyset$	
Concept (Class)	$A$	$A^I$	
Role (Property)	$R$	$R^I$	
Concept Intersection	$C \sqcap D$	$C^I \cap D^I$	S
Concept Disjunction	$C \sqcup D$	$C^I \cup D^I$	
Concept Complement	$\neg C$	$\Delta^I \setminus C^I$	
Concept Inclusion	$C \sqsubseteq D$	$C^I \subseteq D^I$	
Concept Equivalence	$C \equiv D$	$C^I = D^I$	
Universal Role-Value Restriction	$\forall R.C$	$\{x \in \Delta^I : \forall y \in \Delta^I. (x, y) \in R^I \rightarrow y \in C^I\}$	
Existential Role-Value Restriction	$\exists R.C$	$\{x \in \Delta^I : \exists y \in \Delta^I. (x, y) \in R^I \wedge y \in C^I\}$	
Trans. Role Closure	$R^+ \sqsubseteq R$	$(R^I)^+$	
Role Inclusion	$R \sqsubseteq P$	$R^I \subseteq P^I$	H
Inverse Role	$\exists R^-.C$	$\{x \in \Delta^I : \exists y \in \Delta^I. (y, x) \in R^I \wedge y \in C^I\}$	I
Non-Qualified Role (Cardinality) Restr.	$(\leq nR)$ $(\geq nR)$ $(= nR)$	$\{x \in \Delta^I :  \{y \in \Delta^I : (x, y) \in R^I\}  \leq n\}$ $\{x \in \Delta^I :  \{y \in \Delta^I : (x, y) \in R^I\}  \geq n\}$ $\{x \in \Delta^I :  \{y \in \Delta^I : (x, y) \in R^I\}  = n\}$	N
Functional Role	$(\leq 1R)$	$\{x \in \Delta^I :  \{y \in \Delta^I : (x, y) \in R^I\}  \leq 1\}$	F
Qualified Role (Cardinality) Restr.	$(\leq nR.C)$ $(\geq nR.C)$ $(= nR.C)$	$\{x \in \Delta^I :  \{y : (x, y) \in R^I \wedge y \in C^I\}  \leq n\}$ $\{x \in \Delta^I :  \{y : (x, y) \in R^I \wedge y \in C^I\}  \geq n\}$ $\{x \in \Delta^I :  \{y : (x, y) \in R^I \wedge y \in C^I\}  = n\}$	Q
Nominals	$\{o_1, \dots, o_n\}$ $\exists R.\{o\}$	$\{o_1^I, \dots, o_n^I\}$ $\{x \in \Delta^I : \exists y \in \Delta^I. (x, y) \in R^I \wedge y^I \in \{o\}\}$	O
Concept Membership	$a : C$	$a \in C^I$	
Role Membership	$(a, b) : R$	$(a, b) \in R^I$	

Fig. 2.6. Description Logics: Syntax and Semantics.

Atomic DL concepts correspond to unary FOL predicates, and complex concepts correspond to FOL formulae with one free variable. Roles are binary predicates, complex role expressions are equivalent to FOL formulae with one free variable guarded by the role predicate like in  $\exists y.R(x, y) \wedge C(y)$ , and individuals (nominals) are equivalent to FOL constants. Standard class (and role) inclusion axioms (e.g.  $A \sqsubseteq B$ ) correspond to standard first-order material implications with universal variable quantification (e.g.  $\forall x.A(x) \rightarrow B(x)$ ). DL axioms of the form  $a : C$  and  $(a, b) : P$  with named individuals (nominals)  $a, b$  correspond to ground atoms (facts)  $C(a)$  and  $P(a, b)$ . Concepts (or classes) can be explicitly defined by means of nominals (so-called enumerated classes). Finally, an interpretation over given domain of discourse satisfies a DL knowledge base (also called DL-based ontology)  $W$  iff it satisfies every axiom and fact in  $W$ ;  $W$  is consistent iff it is satisfied by at least one interpretation. A DL-based ontology  $W_2$  is entailed by another DL-based ontology  $W_1$  iff every interpretation that satisfies  $W_1$  also satisfies  $W_2$ .

Name	DL Syntax	FOL Syntax
Thing, Nothing	$\top, \perp$	$\top, \perp$
Concept (Class)	$A$	$A(x)$
Role (Property)	$R$	$R(x, y)$
Concept Intersection	$C \sqcap D$	$C(x) \wedge D(x)$
Concept Disjunction	$C \sqcup D$	$C(x) \vee D(x)$
Concept Complement	$\neg C$	$\neg C(x)$
Concept Inclusion Axiom	$C \sqsubseteq D$	$\forall x. C(x) \rightarrow D(x)$
Concept Equivalence Axiom	$C \equiv D$	$\forall x. C(x) \leftrightarrow D(x)$
Universal Role Value Restr.	$\forall R.C$	$\forall y. (R(x, y) \rightarrow C(y))$
Existential Role Value Restr.	$\exists R.C$	$\exists y. R(x, y) \wedge C(y)$
Transitive Role Closure	$R^+ \sqsubseteq R$	$\forall x, y, z. (R(x, y) \wedge R(y, z)) \rightarrow R(x, z)$
Functional Role	$(\leq 1R)$	$\forall x, y, z. (R(x, y) \wedge R(x, z)) \rightarrow y = z$
Non-Qualified Role (Cardinality) Restr.	$(\leq nR)$ $(\geq nR)$	$\forall_{i=1}^{n+1} y_i. \bigwedge_{j=1}^n R(x, y_i) \wedge \bigwedge_{i \neq j} y_i \neq y_j$ $\forall_{i=1}^{n+1} y_i. \bigwedge_{j=1}^n R(x, y_i) \rightarrow \bigvee_{i \neq j} y_i = y_j$
Qualified Role (Cardinality) Restr.	$(\leq nR.C)$	$\forall y_1 \dots y_n. \bigwedge_{1 \leq i \leq n} (R(x, y_i) \wedge C(y_i)) \rightarrow \bigwedge_{1 \leq i < n, i < j \leq n} y_i \neq y_j$
Nominals (Value Restr.)	$\{o_1, \dots, o_n\}$ $\exists R.\{o\}$	$x = o_1 \vee \dots \vee x = o_n$ $R(x, o)$
Concept Instance Assertion	$a : C$	$C(a)$
Role Instance Assertion	$(a, b) : R$	$R(a, b)$

Fig. 2.7. Correspondence between DL and FOL.

### 2.3.3 Reasoning and Complexity

The classical DL reasoning tasks for a given knowledge base (or ontology)  $W = (T, A)$  are as follows.

- *Knowledge base consistency*: To check whether  $W$  has at least one model (satisfiability of  $W$ ). In other words, checking the consistency of ABox  $A$  with respect to the TBox  $T$ , or the consistency of the terminology  $T$  (Abox  $A$ ) itself can be reduced to checking the unsatisfiability of  $W$ .
- *Concept satisfiability*: Given  $W$  and concept  $C$ , we verify whether there is a model of  $W$  in which the interpretation of  $C$  is a non-empty set ( $C^I \neq \emptyset$ ).
- *Concept subsumption*: Given  $W$  and two concepts  $C, D$ , verify whether the interpretation of  $C$  is a subset of the interpretation of  $D$  in every model of  $W$ :  $W \models C \sqsubseteq_T D$  iff  $W \cup \{C \sqcap \neg D(a)\}$  unsatisfiable. Similarly for subsumption checking without a TBox: Is  $C$  interpreted as subset of  $D$  for all FO interpretations  $I$  ( $C^I \subseteq D^I$ ).
- *Instance checking and retrieval*: Given  $W$ , a named individual  $o$  and a concept  $C$ , verify whether  $o$  is an instance of  $C$  in every model of  $W$ :  $W \models C(o)$  (iff  $A$  entails  $o$  wrt  $T$  iff  $W' = A \cup \{\neg C(o)\}$  is inconsistent), or whether the same holds for pairs of objects defined in  $A$  as role fillers of some role defined in  $T$ . Instance retrieval returns the set of all individuals  $o$  in the Abox  $A$  that are instances of a given concept description  $C$  in every model of  $W$ .

Query answering is equivalent to deciding the entailment of query  $q$  by  $W$  ( $W \models q$ ) [165, 131]. In conjunctive query answering (CQA), the answers to a given conjunctive query  $q$  with respect to  $W$  is returned. Apart from the KAON2 system, all state-of-the-art DL reasoners implement the tableaux calculus to provide automated reasoning support for all of the above logical inference tasks. In DLs that are propositionally closed, i.e. that provide, either implicitly or explicitly, conjunction, union and negation of class descriptions such as OWL-DL, OWL-Lite and OWL 1.1 the problems of knowledge base consistency, concept satisfiability, concept subsumption and instance checking can be reduced to each other in polynomial time.

### *Types of complexity*

For practical purposes, when evaluating the complexity of reasoning upon DL knowledge bases, it is commonly distinguished between the following kinds of complexity.

- *Data complexity* is the complexity measured with respect to the number of facts (assertions) in the knowledge base.
- *Taxonomic complexity* is the complexity measured with respect to the size of the axioms in the knowledge base.
- *Query complexity* is the complexity measured with respect to the number of conjuncts (single subqueries) of a conjunctive query.
- *Combined complexity* is the complexity measured with respect to both the size of the axioms and the number of facts. In case of conjunctive query answering, the combined complexity also includes the query complexity.<sup>13</sup>

Decision procedures for conjunctive query entailment in SHIQ and SHOQ are known only recently [131]. Most CQA techniques are seen as problematic since they only allow primitive roles in queries, and do not handle nominals at all. In general, the efficiency of query answering in DL is commonly considered insufficient for large scale knowledge bases in practice.

### **Practical Problems**

Main challenges of practical reasoning with DL-based ontologies concern its scalability, in particular the efficient query answering over knowledge bases with large-scale ABoxes, the handling of cyclic class definitions via blocking, inverse and transitive roles, and nominals.

While research on optimization techniques with respect to nominals has just started, optimized techniques for concept subsumption [158] and query answering for tableaux-based DL reasoners are available (and partially implemented) even for the case of changing ABoxes with frequent updates [149, 235, 90].

<sup>13</sup> In the following, we refer to the combined complexity of a given DL, if not stated otherwise.

There are currently two different approaches towards scalable ABox reasoning. One approach is to partition the ABox so that some kinds of reasoning can be performed separately on each partition and trivially combine the results [121, 136]. Another approach is to (a) convert the knowledge base in the (function- and negation-free) FOL fragment Horn-SHIQ to an equisatisfiable disjunctive Datalog program with equality replacement rules and integrity constraints (Datalog <sup>$\forall, IC$</sup> ), and then (b) to reduce this further to plain Datalog (def-Horn) by splitting the disjunctive rule heads in order to use any of the existing Datalog engines for answering ground queries (e.g.  $C(a)$ ,  $a$  object) more efficiently (computing the minimal Herbrand model of the plain Datalog program including the ABox facts) in polynomial data complexity. Though the preceding conversion in disjunctive Datalog, in particular the saturation of the knowledge base with all grounded inferences related to the query is in NEXPTIME (cf. section 2.5.6).

Motik et al. [262] propose a resolution-based algorithm which evaluates also non-grounded queries in one pass so that the efficiency of query answering is further improved. This still requires exponential space in the worst case calling for appropriately partitioning the ABox to cut down memory consumption.

#### *Further readings*

For a more comprehensive treatment of description logics, and their wide range of applications, we refer the interested reader to the excellent handbook of description logics (Baader et al., 2003)[19], and the Web pages of the DL community at <http://dl.kr.org>.

## 2.4 Semantic Web Ontology Languages

The current W3C standards of ontology languages for the Semantic Web include RDFS and the OWL language family. Prominent non-standard examples of the ontology languages landscape are the Semantic Web rule language (SWRL), the WSM language family, and Description logic programming (DLP).

### 2.4.1 RDF and RDFS

The RDF (Resource Description Framework) language<sup>14</sup> and its extension RDF Schema (RDFS) are probably the most adopted Semantic Web languages today.

---

<sup>14</sup> [www.w3.org/RDF/](http://www.w3.org/RDF/)

## Syntax and Semantics of RDF(S)

Driven by the least possible commitment to a particular data model for the Web, RDF uses directed labeled graphs, also called triple-based data model, for representing information. Any RDF graph is a set of directed labeled edges, or statements commonly written as triples of the form (*Subject Predicate Object*). The edge links *Subject* denoting a resource identified by an URI or blank node to another resource, blank node or datatype (or XML) literal *Object* with *Predicate* as label denoting a property of the origin node *Subject*. Blank (anonymous) nodes are used to express incomplete information, or queries. RDF uses XML for its serialization.

One can consider RDF as an first-order assertional logic in which each triple expresses a simple proposition. This imposes a fairly strict monotonic discipline on the language, so that it cannot express closed world assumption, local default preferences, and several other commonly used non-monotonic constructs like multiple inheritance and overriding.

RDF Schema (RDFS) extends RDF in order to express simple taxonomies and hierarchies among resources and properties. RDFS allows to define more complex ontological vocabularies for RDF descriptions in terms of simple taxonomies and hierarchies of classes (concepts) and properties (roles), containers, and property restrictions. RDFS statements are equivalent to DL axioms of the form  $C \sqsubseteq D$ ,  $\top \sqsubseteq \forall P : C$ ,  $\top \sqsubseteq \forall P^- . C$ ,  $P \sqsubseteq Q$ ,  $a : C$  and  $(a, b) : P$ . Thus, the semantics of RDFS can to a large extent be approximated by a set of FOL sentences. For details, we refer the interested reader to [151]<sup>15</sup>

## Support of RDF(S)

Querying RDF knowledge bases is commonly reduced to RDF graph matching which is implemented for most RDF query languages such as the SQL-like and W3C recommendation SPARQL (SPARQL Protocol and RDF Query Language)<sup>16</sup>, one of its predecessors RDQL<sup>17</sup>, and Versa-QL<sup>18</sup>.

Prominent frameworks supporting RDF reasoning are HP's Jena<sup>19</sup>, RacerPro, Sesame, TRIPLE, KAON2 and Corese<sup>20</sup>. TRIPLE<sup>21</sup> offers additional rules reasoning support on top of RDF(S) graphs. Sesame<sup>22</sup> is another open source

---

<sup>15</sup> One alternative is to translate RDFS constructor semantics to function-free, negation-free def-Horn rules and consider RDF statements as facts in the resulting plain Datalog program of which its minimal Herbrand model then is a correct, smallest possible interpretation that satisfies the given RDF graph.

<sup>16</sup> [www.w3.org/TR/rdf-sparql-query/](http://www.w3.org/TR/rdf-sparql-query/)

<sup>17</sup> [www.w3.org/Submission/RDQL/](http://www.w3.org/Submission/RDQL/)

<sup>18</sup> [www.xml.com/pub/a/2005/07/20/versa.html](http://www.xml.com/pub/a/2005/07/20/versa.html)

<sup>19</sup> [www.hpl.hp.com/semweb/tools.htm#jena](http://www.hpl.hp.com/semweb/tools.htm#jena)

<sup>20</sup> [www-sop.inria.fr/acacia/soft/corese/](http://www-sop.inria.fr/acacia/soft/corese/)

<sup>21</sup> [triple.semanticweb.org/](http://triple.semanticweb.org/)

<sup>22</sup> [www.openrdf.org/](http://www.openrdf.org/)

framework for storage, inferencing and querying of RDF data. The Corese platform<sup>23</sup> implements an RDF/RDFS processor based on the conceptual graph model with a special graph matching algorithm and SPARQL. The W3C Metalog system<sup>24</sup> allows querying of RDF resources in a NL-oriented language (PNL) interfaced with an underlying logical extension of the RDF semantics and model. The Oracle 11g RDF database provides full RDFS support for querying RDF and a subset of OWL-DL inside Oracle's relational database management system<sup>25</sup>.

For more information on the practical use of RDF(S), and its relation to other knowledge modeling approaches like UML, XML, and XML-TopicMaps, we refer to the respective W3C site<sup>26</sup>.

### 2.4.2 OWL

The standard ontology language for the Semantic Web is OWL (Ontology Web Language)[71, 160]. It has its roots in the joint initiative DAML+OIL of researchers from the US and Europe in 2000 to develop a formal annotation or mark-up language for the Web. Only three years later, OWL became a W3C recommendation and has been widely adopted by both industry and academics since then. The current version of OWL is OWL 1.1<sup>27</sup>.

#### Variants

OWL comes in several variants, that are OWL-Full, OWL-DL, and OWL-Lite. Each variant corresponds to a DL of different expressivity and complexity. OWL-Lite and OWL-DL are an abstract syntactic form of the description logic SHIF(D), respectively, SHOIN(D), whereas OWL-Full corresponds to the description logic SHOIQ(D)\*. For syntax and model-theoretic semantics of these description logics, we refer to figure 2.3.1 in section 2.3.1.

**OWL-Full.** The most expressive but undecidable variant OWL-Full provides full compatibility with RDFS and covers the expressivity of the description logic SHOIQ(D)\* which offers not only simple data types (D) but inverse roles (I), roles as subroles (a role hierarchy H), role transitivity (S) and qualified role cardinality restrictions (Q), as well as derived classes (classes used as individuals) together with non-primitive roles (cf. figure 2.3.1, section 2.3.1). Since OWL-Full allows in particular non-primitive roles (which can either be transitive or have transitive subroles) in role cardinality restrictions (S\*), it is undecidable (while SHOIQ(D) is not) [164].

---

<sup>23</sup> [www-sop.inria.fr/acacia/soft/corese/](http://www-sop.inria.fr/acacia/soft/corese/)

<sup>24</sup> [www.w3.org/RDF/Metalog/](http://www.w3.org/RDF/Metalog/)

<sup>25</sup> [www.oracle.com/technology/tech/semantic\\_technologies/](http://www.oracle.com/technology/tech/semantic_technologies/)

<sup>26</sup> [www.w3.org/RDF/developers](http://www.w3.org/RDF/developers)

<sup>27</sup> [www.w3.org/Submission/2006/10/](http://www.w3.org/Submission/2006/10/)

**OWL-DL.** Unlike OWL-Full, the less expressive variant OWL-DL (SHOIN(D)) allows only for unqualified number (role cardinality) restrictions, and does not permit to state that a role  $P$  is transitive or the inverse of another role  $Q \neq P$ . In particular, OWL-DL does not include relationships between (transitive) role chains which would cause its undecidability. That is, in role number restrictions, only simple roles which are neither transitive nor have transitive subroles are allowed; otherwise we gain undecidability even in SHN [164]. OWL-DL also does not allow classes to be used as individuals (derived classes), or to impose cardinality constraints on subclasses.

**OWL-Lite.** The variant OWL-Lite (SHIF(D)) is even less expressive than OWL-DL. It prohibits unions and complements of classes, does not allow the use of individuals in class descriptions (enumerated classes, nominals O), and limits role cardinalities to 0 or 1 (F). However, it is possible to capture all OWL-DL class descriptions except those containing either individuals or role cardinalities greater than 1 by properly exploiting the implicit negations introduced by disjointness axioms, and introducing new class names [161]. In role cardinality restrictions, only simple roles are allowed; however, it is unknown whether SHF or SHIF becomes undecidable without this restriction [164].

The syntactic transformation from OWL-Lite and OWL-DL ontologies to corresponding DL knowledge bases is of polynomial complexity. What makes OWL a Semantic Web language is not its semantics (which are quite standard for a DL) but the use of URI references for names, the use of XMLS datatypes for data values, and the ability to connect to documents in the Web. The abstract syntax of OWL can be mapped to the normative syntax of RDF.

### Relation to RDFS

OWL adds constructors to RDFS for building more complex class (concept) and property (role) descriptions with model-theoretic semantics. For example, the use of intersection (union) within (sub-)class descriptions, or universal/existential quantifications within super-/subclasses in OWL is not possible in RDFS[161]. However, the variants OWL-DL and OWL-Lite are extensions of a restricted use of RDFS whereas OWL-Full is fully upward compatible with RDFS. As mentioned above, OWL-DL and OWL-Lite do not allow classes to be used as individuals, or to impose cardinality constraints on subclasses, and the language constructors cannot be applied to the language itself - which is possible in OWL-Full and RDFS.

It has been shown only recently in [283] that the formal semantics of a sub-language of RDFS is compatible with that of the corresponding fragment of OWL-DL such that RDFS could indeed serve as a foundational language of the Semantic Web layer stack. Though checking whether a RDF graph is an OWL ontology and upgrading from RDFS to OWL remains hard in practice,

and is topic of ongoing research. For a detailed treatment of this subject, we refer to [131].

## Complexity

As mentioned above, logical entailment or concept subsumption reduced to concept satisfiability is decidable for OWL-Lite and OWL-DL in EXPTIME (-complete), respectively, NEXPTIME (-complete) [159, 359]. Noticeably, concept satisfiability in SHOIQ(D) is intractably co-NEXPTIME-hard [359], while its variant SHOIQ(D)\*, hence OWL-Full, allowing non-primitive transitive roles to occur in role cardinality restrictions (S\*) is undecidable. Undecidability of SHOIQ\* has been proven by reduction to the Domino problem [71].

Reasoning with data types and values (D) for all OWL variants can be separated from reasoning with concepts and individuals by allowing the DL reasoner to access a kind of datatype oracle that answers simple questions with respect to data types and values; this way, the language remains decidable if data type and value reasoning is decidable, i.e., if the oracle can guarantee to answer all questions of the relevant kind for supported datatypes. Figure 2.8 shows the relation of OWL to other prominent (polynomially reducible) tractable DL subsets like EL++, Horn-SHIQ, and DLPs (cf. section 2.5.4) together with related complexity results (cf. section 2.3.3) [55].

Efficient query answering over DL knowledge bases with large ABoxes (instance stores) and fixed TBoxes is of particular interest in practice. Unfortunately, OWL can be considered insufficient for this purpose in general: Conjunctive query answering for SHIQ and SHIF underlying OWL-Lite is decidable but only in time exponential in the size of the knowledge base (taxonomic complexity) and double exponential in the size of the query [131] (query and combined complexity); the CQA complexity for OWL-DL is unknown. For both OWL-DL (SHOIN(D)) and OWL-Lite SHIF(D) knowledge base (DL-based ontology) entailment checking<sup>28</sup> is decidable.

## Critique

The main criticism of the standard Semantic Web ontology language OWL is that it only allows for static declarative knowledge representation of limited expressivity and reasoning support. For example, OWL does not allow temporal or spatial reasoning nor integrity constraints. Approaches to overcome these limitations of OWL by means of adding monotonic or nonmonotonic rules from logic programming are briefly discussed in section 2.5. The following section summarizes existing tools for reasoning on OWL ontologies.

---

<sup>28</sup>  $W_1$  entails  $W_2$ , i.e.,  $W_1 \models W_2$  if and only if every model of  $W_1$  also satisfies  $W_2$ .



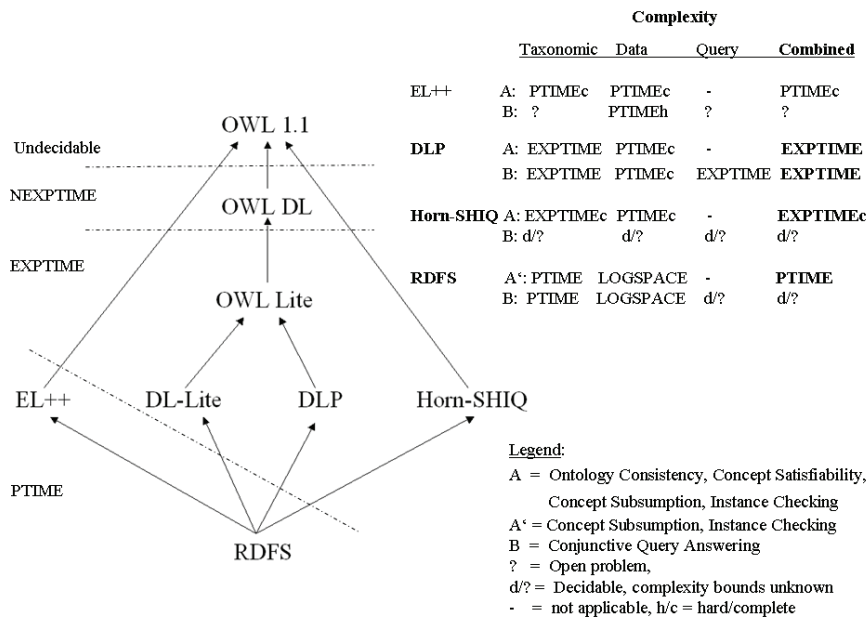


Fig. 2.8. Tractable fragments of OWL ([55])

## OWL Support

There are many implemented OWL reasoners available for both OWL-DL (SHOIN(D)) and OWL-Lite SHIF(D) that cover all major DL inferencing tasks. Each of these reasoners provide access to their functionality either through proprietary APIs that conform to their implementation language, or support the standard DIG interface[27] for handling DL elements in an XML format.

State-of-the-art OWL reasoners include QuOnto for OWL-Lite, FaCT++<sup>29</sup> and Pellet for OWL-DL [163] as part of the OWL-API, Racer/RacerPro<sup>30</sup> for OWL-Lite and OWL-DL with approximations for nominals. Another prominent framework for reasoning with OWL are KAON1 and KAON2 for SHIQ extended with the decidable DL-safe fragment of SWRL, i.e. for OWL-DL with qualified role cardinality restrictions but without nominals (cf. section 2.5.6).

The above reasoners are claimed to be sound and complete, though a proof has been published for KAON2 only. Besides, they are tableaux-based DL reasoners, except KAON2 which reduces a OWL-DL knowledge base  $W$  to a

<sup>29</sup> owl.man.ac.uk/factplusplus/

<sup>30</sup> www.racer-systems.com

disjunctive Datalog program  $P$  [172] (in EXPTIME) - such that  $W$  and  $P$  entail the same set of ground facts - and then efficiently solves  $P$  by means of a disjunctive Datalog engine through magic set transformation [78] (bottom-up evaluation of  $P$ ) and resolution-based theorem proving [276]. Besides, KAON2 uses an equisatisfiable semantics of SHOIN(D) defined through translation into a multi-sorted (for concrete domains  $D$ ) first-order logic [164]. This enables optimized assertional DL reasoning (OWL-DL ABoxes, F-Logic instance bases, RDF triple stores) for (approximate) query answer set computation over large ABoxes.

Since OWL-DL is a decidable FOL fragment one can also translate OWL-DL knowledge bases into a collection of equivalent FOL formulas and use FOL theorem provers for consistency checking. For example, Hoolet<sup>31</sup> is an OWL-DL reasoner that uses the FOL theorem prover Vampire [363].

Prominent OWL query languages are OWL-QL[119]<sup>32</sup>, SAIQL (Schema And Instance Query Language)<sup>33</sup>, and GLOO (Graphical Query Language for OWL Ontologies)[112]. For example, OWL-QL[119] is a formal language and protocol for query-answering dialogues among Semantic Web computational agents. An OWL-QL query is basically a OWL knowledge base with a specification which of the URIs referred to in the query pattern are to be interpreted as variables. Variables come in three forms: must-bind, may-bind, and do-not bind variables. OWL-QL uses the standard notion of logical entailment: query answers can be seen as logically entailed sentences of the queried knowledge base. OWL-QL allows not only extensional queries but also structural queries such as "retrieve the subsuming concept names of the concept name father". The use of the RDF query language SPARQL for querying OWL-DL knowledge bases via the interaction between the Protege and the Jena2 toolkit is problematic for the following reason. SPARQL does not support the model-theoretic interpretation of RDFS or OWL ontologies. That is, there is neither one single query graph like in RDF but many possible interpretations (models) of an OWL ontology, nor a finite query graph since each of these models can be infinite. On the other hand, conjunctive queries over OWL-DL knowledge bases could be considered as SPARQL query graphs of conjunctive RDF triples. However, the treatment of variables differs in both approaches such that the extension of SPARQL to OWL-DL can only be approximative in principle ([154], pp 235).

### 2.4.3 WSMO and WSML

In this section, we informally introduce the reader to the basic elements of semantic service description in the Web service modeling language (WSML).

---

<sup>31</sup> owl.man.ac.uk/hoolet/

<sup>32</sup> projects.semwebcentral.org/projects/owl-ql/

<sup>33</sup> www.uni-koblenz.de/FB4/Institutes/IFI/AGStaab/Research/SAIQL

#### 2.4.4 WSMO Framework

The WSMO (Web Service Modelling Ontology) framework<sup>34</sup> provides a conceptual model and a formal language WSML (Web Service Modeling Language)<sup>35</sup> for the semantic markup of Web services together with a reference implementation WSMX (Web Service Execution Environment). Historically, WSMO evolved from the Web Service Modeling Framework (WSMF) as a result of several European Commission funded research projects in the domain of Semantic Web Services like DIP, ASG, Super, TripCom, KnowledgeWeb and SEKT in the ESSI (European Semantic Systems Initiative) project cluster<sup>36</sup>.

WSMO offers four key components to model different aspects of Semantic Web services in WSML (Web Service Modeling Language): Ontologies, goals, services, and mediators. Goals in goal repositories specify objectives that a client might have when searching for a relevant Web service. WSMO ontologies provide the formal logic-based grounding of information used by all other modeling components. Mediators bypass interoperability problems that appear between all these components at data (mediation of data structures), protocol (mediation of message exchange protocols), and process level (mediation of business logics) to "allow for loose coupling between Web services, goals (requests), and ontologies". Each of these components, called top-level elements of the WSMO conceptual model, can be assigned non-functional properties to be taken from the Dublin Core metadata standard by recommendation.

#### WSML Variants

The Web service modeling language WSML allows to describe a Semantic Web service in terms of its functionality (service capability), imported ontologies, and the interface through which it can be accessed for orchestration and choreography. The syntax of WSML is mainly derived from F-Logic extended with more verbose keywords (e.g., "hasValue" for  $- >$ , "p memberOf T" for  $p:T$  etc.), and has a normative human-readable syntax, as well as an XML and RDF syntax for exchange between machines. WSML comes in five variants with respect to the logical expressions allowed to describe the semantics of service and goal description elements. In the following, we informally introduce F-Logic and the WSML variants in very brief.

**F-Logic.** F-Logic is an object-oriented extension of first-order predicate logic with objects of complex internal structure, class hierarchies and inheritance, typing, and encapsulation in order to serve as a basis for object-oriented logic

---

<sup>34</sup> <http://www.wsmo.org/TR/d2/v1.4/20061106>

<sup>35</sup> <http://www.wsmo.org/TR/d16/d16.1/v0.21/20051005/>

<sup>36</sup> <http://www.sdkcluster.org/>

programming and knowledge representation. For modeling ontologies, it allows to define, for example, is-a object class (or type) hierarchies through subclass relationships like `person::human` denoting class "person" as a subclass of "human, a class of objects with structured properties (object type signature) like `person[name * $\Rightarrow$  string, children * $\Rightarrow$  person]`, and instances of classes (typed objects) like `john:person` as well as rules like (`R:region :- R1:region, R::R1.`) and (`L:location :- L:R, R:region.`) denoting that every subclass "R" of an object class "R1" of type "region" is a region and that every member L of a region "R" is also a location. Rules may also be used to define virtual classes like the rule (`X:redcar :- X:car, X[color  $\rightarrow$  red].`) defining the virtual class "redcar".

F-Logic comes in two flavors with respective variants: A first-order F-Logic variant (F-Logic(FO)) that includes an (OWL-DL/WSML-DL) description logic subset of classical predicate logic, and a full logic programming (LP) variant (F-Logic(LP)) that is LP extended with procedural built-ins (functions), and nonmonotonic default inheritance and negation-as-(finite)-failure<sup>37</sup>. Nonmonotonic (default) inheritance of F-Logic(LP) allows to override default property values of classes inherited by subclasses. For example, a class `Elephant[color * $\rightarrow$  grey]` with default value "grey" of property "color" has a subclass `royalElephant[color * $\rightarrow$  white]` for which objects this default value of inherited property "color" is overridden by (default) value "white". Hence, one can assert object `fred[color  $\rightarrow$  grey]` as member of class "Elephant" (but not "royalElephant"), and `clyde[color  $\rightarrow$  white]` as member of both classes. Semantics of F-Logic(LP) are derived from Van Gelder's well-founded (fix-point-based, minimal model) semantics of the nonmonotonic part of logic programming [368]. F-Logic(LP) is more commonly used than F-Logic(FO) like in the LP-reasoners `OntoBroker`, `Flora-2` and `Florid`. For more details on the syntax and semantics of F-Logic, we refer to [8, 187, 389].

**WSML variants.** The formal semantics of WSML service description elements are specified as logical axioms and constraints in ontologies using one of five WSML variants: `WSML-Core`, `WSML-DL`, `WSML-Flight`, `WSML-Rule` and `WSML-Full` (cf. Figure 2.9).

Though WSML has a special focus on annotating Semantic Web services like `OWL-S` it tries to cover more representational aspects from knowledge representation and reasoning under both classical FOL and nonmonotonic LP semantics. For example, `WSML-DL` is a decidable variant of F-Logic(FO) with expressivity close to the description logic `SHOIN(D)`, that is the variant `OWL-DL` of the standard ontology Web language `OWL`. `WSML-Flight`

<sup>37</sup> In nonmonotonic LP, like semi-decidable `PROLOG` and F-Logic(LP), the default negation of fact  $p$  (**not**  $p$ ) means " $p$  is true if  $p$  cannot be proven in a given knowledge base  $KB$  in finite time" (under closed-world assumption). This is nonmonotonic, i.e., truth values of asserted and implied knowledge in  $KB$  do not grow monotonically:  $(KB \models p)$  does *not* imply  $(KB \cup \{q\} \models p)$ , e.g.,  $KB = \{(p :- \mathbf{not} \ q)\}$  implies  $p$  true ( $KB \models p$ ), but  $KB^* = \{q, (p :- \mathbf{not} \ q)\}$  implies  $p$  false.

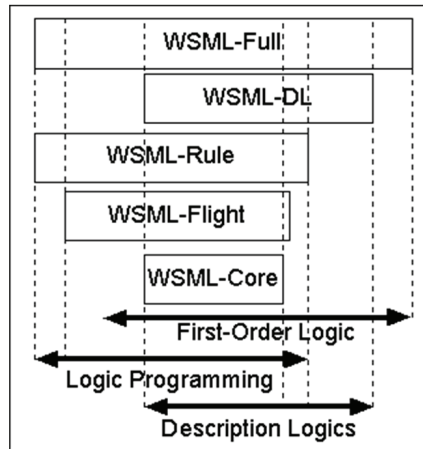


Fig. 2.9. WSML language variants.

is a decidable Datalog variant of F-Logic(LP) (function-free, non-recursive and DL-safe Datalog rules) with (nonmonotonic) default negation under perfect model semantics [302] of locally stratified F-Logic programs with ground entailment. WSML-Rule is a fully-fledged logic programming language with function symbols, arbitrary rules with inequality and nonmonotonic negation, and meta-modeling elements such as treating concepts as instances, but does not feature existentials, strict (monotonic) negation, and equality reasoning. The semantics of WSML-Rule is defined through a mapping to undecidable (nonmonotonic, recursive) F-Logic(LP) variant with inequality and default negation under well-founded semantics [368]. WSML-Full shall unify the DL and LP paradigms as a superset of FOL with non-monotonic extensions to support nonmonotonic negation of WSML-Rule via Default Logic, Circumscription or Autoepistemic Logic. However, neither syntax nor semantics of WSML-Full have been completely defined yet.

The description of Semantic Web services in WSML will be treated in more detail in the following chapter.

### Mediators in WSMO

As mentioned above, mediators are responsible to resolve semantic heterogeneities between ontologies in WSMO. Currently the WSMO specification covers four different types of mediators.

- ooMediators import the target ontology into the source ontology by resolving all the representation mismatches between source and target;
- ggMediators connect goals that are in a relation of refinement and resolve mismatches between those;

- wgMediators link Web services to goals and resolve mismatches;
- wwMediators connect several Web services for collaboration.

Mediators of type ggMediator, wgMediator, wwMediator shall use ooMediators in order to align different imported domain ontologies (or shared basic vocabularies out of which different domain ontologies can be canonically built, cf. Section 2.3) that are used for describing goals and Web service capabilities.

## Critique

From the WSMO specification it remains completely unclear how the proposed mediators shall be implemented apart from pointing either to a goal that declaratively describes the required "magic" mapping between the description elements, or to a Web service that actually implements this mapping, or to a wwMediator that just links to such a service. In any case, the solution to the problem is shifted completely outside the framework. Other critics of WSML with respect to its use for Semantic Web services are presented in the following chapter (section 2.4.3).

## WSMO Support

Implemented approaches to build and query ontologies in non-standard WSML include the WSMO studio<sup>38</sup> with WSML validator, the WSMO4J API<sup>39</sup>, and reasoners for WSML-Core, WSML-DL and WSML-Rule. Note that reasoning on ontologies in WSML, in opposite to OWL, comes in two different flavors: Reasoning on WSML-DL is monotonic, while it is nonmonotonic for WSML-Flight and WSML-Rule with negation by default. For more information on reasoning support of WSML, we refer to respective deliverables of the WSMO initiative ([www.wsmo.org](http://www.wsmo.org)).

## 2.5 Semantic Web Ontologies and Rules

Semantic Web research considers both ontologies and rules as separate "stacks" of the Semantic Web architecture (cf. section 2.1) with an unifying logic on top of them. The standardized markup, publishing and interchange of rules between different systems and tools is expected to benefit many business applications over the Web.

### 2.5.1 Motivation

Main motivation of combining rules with OWL ontologies are to overcome

---

<sup>38</sup> <http://www.wsmostudio.org/download.html>

<sup>39</sup> <http://wsmo4j.sourceforge.net/>

- (a) the limited expressivity of description logics underlying OWL for practical applications, and
- (b) the scalability problem of extensional querying OWL ontologies with large ABoxes (cf. section 2.3.3) by use of optimized LP rule reasoning engines like SWI-Prolog, XSB, dl<sub>v</sub> and smodels.

A normal disjunctive logic program  $P$  consists of a finite set of rules

$$A_1 \vee \dots \vee A_k \leftarrow B_1 \wedge \dots \wedge B_m \wedge \text{not } B_{m+1} \wedge \dots \wedge \text{not } B_n$$

with  $k \geq 0, n \geq m \geq 0$ , and facts. In the following, we call a rule disjunctive normal if  $k > 0$ , definite if  $k = 1$  (no disjunction), positive if negation-as-failure (*not*) does not appear ( $m = n$ ), disjunctive if it is positive and  $k > 0$ , Horn if it is definite and positive ( $H \leftarrow B_1 \wedge \dots \wedge B_m$ ), and Datalog if it is function-free. Definite logic programming (definite LP) with definite, positive rules is related to the undecidable Horn FOL, while its subset def-LP relates to the decidable function-free Horn FOL (called def-Horn, plain Datalog).

*Overcoming limited expressivity of first-order description logics*

**Relational knowledge.** Description logics are very limited in representing relational knowledge. The reasons are that (a) they are designed as decidable subsets of undecidable 2-variable FOL fragment with one free variable only, and (b) they are restricted to unary and binary predicates only. In particular, the decidability of most DLs is due to their having the so-called tree-model property, i.e., any model of a given DL knowledge base  $kb$  defines a finite, tree-shaped directed graph with depth and branching factor bounded by the size of  $kb$ .

This restricts the way variables and quantifiers can be used in DLs. In fact, each quantified variable must occur in a role along with the only free variable (e.g.  $\exists y.R(x, y) \wedge C(y)$ ). As a consequence, it is impossible to describe concepts whose instances are related to other (anonymous) instances via different role paths. One example is the well-known relational concept *uncleOf* that is not possible to express in DL but as the function-free Horn rule  $\text{uncleOf}(Z, Y) \leftarrow \text{fatherOf}(X, Y), \text{brotherOf}(X, Z)$  that has a triangular-shaped model<sup>40</sup>.

**Integrity constraints.** DLs do not support integrity constraints that are used to check the consistency of a knowledge base without deriving new facts. In deductive databases such constraints are modeled in Datalog<sup>IC</sup> as def-Horn rules with empty heads, i.e., rules of the form  $\leftarrow B_1, \dots, B_n$  (equivalent to  $\perp \vee \neg B_1 \vee \dots \vee \neg B_n$ ) which asserts that statements  $B_i$  can never become

<sup>40</sup> Similarly, the concept of homeworkers as individuals who live and work at the same location are not expressible in description logics but in def-Horn (e.g.  $\text{HomeWorker}(X) \leftarrow \text{Person}(X), \text{livesAt}(X, Y), \text{worksAt}(X, Y)$ , or  $\text{HomeWorker}(X) \leftarrow \text{worksAt}(X, Y), \text{livesAt}(X, Z), \text{loc}(Y, W), \text{loc}(Z, W)$ ).

true simultaneously<sup>41</sup>.

**Nonmonotonic reasoning.** Unlike normal logic programming with default negation, or autoepistemic description logics with modal operator **K** that allows for local closed-world reasoning, first-order description logics like OWL-DL are restricted to monotonic reasoning under open-world assumption. Since the Semantic Web is open-ended, this is considered appropriate by the W3C, but for many practical Web applications like ticket booking services over finite databases it makes much more sense to perform (local) closed-world reasoning instead. In particular, nonmonotonic negation-as-finite-failure, defeasible rules with priorities, procedural attachments and builtins to compute, for example, aggregations like count, sum, max, avg)<sup>42</sup>, as well as nonmonotonic class inheritance (overriding default property values) are commonly considered useful for data intensive Web applications in practice, but impossible to express with FOL much less description logic.

#### *Rules reasoning support*

Main reasoning tasks of normal logic programming with nonmonotonic (default) negation-as-failure under minimal model (or stable model) semantics concern the inference of factual knowledge (ground entailments, query answering). These inferences are as follows [105]:

- a) Given a logic program  $P$  and ground literals (facts)  $l_1, \dots, l_n$ , decide whether they simultaneously hold in every stable model of  $P$ ;
- b) Given a logic program  $P$  and non-ground literals  $l_1, \dots, l_n$  over variables  $X_1, \dots, X_k$ , return all value assignments  $v$  to  $X_j$  such that  $l_1, \dots, l_n$  evaluate to true.

Prominent examples of implemented frameworks that provide monotonic rule reasoning support include JESS<sup>43</sup> with its own declarative XML rule language, the TRIPLE<sup>44</sup> Horn rules engine (on top of RDF/S, uses external DL reasoners like RACER to cover OWL outside def-Horn), the KAON2 reasoner<sup>45</sup> (covering SWRL, WSML-DL, Horn-SHIQ), and the FUB SWRL engine<sup>46</sup>, and Hoolet covering SWRL rules and the (monotonic layers of)

---

<sup>41</sup> For example, disjointness of OWL concepts  $C, D$  ( $C \sqcap D \equiv \perp$ ) corresponds to the rule  $\leftarrow CD$ . Every model of a logic program  $P$  that satisfies this rule body evaluates the rule to false ( $\perp$ ), hence is considered inconsistent and therefore has to be dropped.

<sup>42</sup> Any aggregation implicitly applies default negation, since it is implicitly assumed that no additional facts have to be taken into account for computing the aggregation.

<sup>43</sup> [jessrules.com/jess/index.shtml](http://jessrules.com/jess/index.shtml)

<sup>44</sup> [triple.semanticweb.org/](http://triple.semanticweb.org/)

<sup>45</sup> [kaon2.semanticweb.org/](http://kaon2.semanticweb.org/)

<sup>46</sup> [www.ag-nbi.de/research/swrlengine/](http://www.ag-nbi.de/research/swrlengine/)



SWSL-Rule<sup>47</sup>. Tools that additionally support nonmonotonic rule reasoning include the KAON2 reasoner, the OntoBroker and the FLORA-2 reasoner (for F-Logic(LP) underlying WSML-Flight and WSML-Rule), and the Bossam Rule/OWL reasoner.

### 2.5.2 Issues of Combining Rules With Ontologies

The problem of combining rules with ontologies is to combine a first-order theory,  $\phi$ , that is a set of logical formulae in a first-order language  $L\phi$  over signature  $\sigma_\phi$  with a logic program (or rule base)  $P$ , that is a set of (monotonic or nonmonotonic) rules in a language  $L_P$  over signature  $\sigma_P$ , in a combined knowledge base  $kb = (\phi, P)$ . There are several issues related to this problem which we only indicate in very brief. For a more elaborated discussion of them with examples, we refer to the excellent readings [105, 103].

*Classical first-order logic vs. logic programming*

**Ground entailments.** As well-known, the function-free Horn subset of classical FOL corresponds to the core of logic programming without negation-as-failure, that is plain Datalog. For entailments of positive facts  $\alpha$  (positive ground entailments), the first-order semantics of  $\phi$  ( $\phi \models \alpha$ ) coincides with the minimal model semantics of  $P$  ( $P \models_c \alpha$ ). This does neither hold for negative ground entailments nor for non-ground entailments in general<sup>48</sup>. Besides, the first-order reading of a positive logic program  $P$  can also allow for additional non-factual inferences that are not entailed by  $P$ .

**Negation-as-failure vs. classical negation.** Negation-as-failure (*not*) in normal logic programs is evaluated under closed-world assumption, hence is nonmonotonic. Negation ( $\neg$ ) in first-order logics is interpreted classically under open-world assumption, hence is monotonic. As one consequence, negative ground entailments that hold in definite programs  $P$  with negation may not hold in the first-order description logics reading of  $P$ <sup>49</sup>.

<sup>47</sup> [www.w3.org/Submission/SWSF-SWSL/](http://www.w3.org/Submission/SWSF-SWSL/)

<sup>48</sup> For example ([262]), in a simple knowledge base  $P$  of one fact or ground atom  $P = \{C(a)\}$ , the classically negated fact  $\alpha = \neg C(o)$  is not true (under OWA) in all first-order models of  $P$ :  $P \not\models \alpha$ . That is, its negation  $C(o)$  cannot be explicitly derived from  $P$ :  $P \models \neg C(o)$  if  $P \cup \{C(o)\}$  is unsatisfiable, but this does not hold through at least the model  $\{C(a), C(o)\}$ . In contrast, the default negated fact  $\alpha = \text{not } C(o)$  is true (under CWA) in the only minimal (w.r.t set inclusion) model  $\{C(a)\}$  of  $P$ :  $P \models_c \alpha$ . Non-ground entailments like concept subsumption  $\alpha = (\forall x.C(x) \rightarrow D(x))$  can be decided in most description logics under first-order semantics but, in general, not in normal logic programming with default negation under stable model semantics. However, as mentioned above, LP is mostly concerned with factual inferences (ground entailments).

<sup>49</sup> In the simple example above, the classically negated fact  $\neg C(o)$  does not hold in  $P$  under OWA such that adding  $C(o)$  later to  $P$  does not falsify any previously

**Unique names assumption.** The unique names assumption (UNA) states that individuals with different names (URI, IRI) are different, i.e., there are no aliases or synonyms. This assumption is not made in classical logic, hence not in RDF and OWL. In OWL knowledge bases it is possible to deduce that two individuals or concepts with different names are the same based on their logical definitions. OWL also offers operators to specify whether named individuals are the same or not (`sameIndividualAs`, `differentFrom`,  $\leq nR(o)$ ). In logic programming, the UNA does hold, hence any equality reasoning by means of some equality predicate in rule heads is not supported.

The UNA for a combined knowledge base  $kb = (\phi, P)$  can be obtained by adding a special binary equality predicate as a classical built-in predicate with the usual equality axioms in  $\phi$ . An example of equality predicate  $eq$  in  $P$  axiomatised through DL-safe function-free Horn rules is as follows:  $eq(X, X)$ ;  $eq(X, Y) \leftarrow eq(Y, X)$ ;  $eq(X, Z) \leftarrow eq(X, Y), eq(Y, Z)$ ;  $C(Y) \leftarrow C(X), eq(X, Y)$  for every concept  $C$  and  $R(C, D) \leftarrow R(A, B), eq(C, A), eq(D, B)$  for every role  $R$  in  $\phi$ . Alternatively, the UNA could be axiomatized by asserting inequality between every set of distinct individuals (constants) of  $kb$ .

#### *Decidability and DL-safe rules*

Another problem of combining first-order ontologies with rules from normal logic programming is that the combination of decidable fragments of both worlds can be undecidable. For example, query answering and satisfiability checking in the decidable description logic ALCNR combined with a decidable Datalog variant in CARIN has been shown undecidable (Halevy and Rousset, 1998) [231]. Similarly, the combination of SHOIN(D) with function-free Horn (def-Horn) in the Semantic Web rule language SWRL is undecidable (cf. section 2.5.5).

The main reason is that the evaluation of existential role constraints (e.g.  $Person \sqsubseteq \exists father.Person$ ) equivalently encoded as a recursive rule (e.g.  $father(X) : Person \leftarrow Person(X)$ ) with function symbols ( $father$ ) for object creation can, for a given query (e.g.  $?XhasFather?Y$ ), create an infinite chain of anonymous individuals (e.g.  $father(father(...(peter))$  for  $?X/peter$ ). The syntactic DL-safety condition for rules avoids this problem. It requires that each variable occurring in a rule must occur in a positive non-DL-atom in the rule body, and may therefore be bound only to constants, that are named (not anonymous) individuals in the ABox of the DL part of the combined knowledge base. For example, the rule  $uncleOf(Z, Y) \leftarrow fatherOf(X, Y), brotherOf(X, Z)$  with OWL-DL roles (DL-atoms)  $brotherOf$ ,  $fatherOf$  is not DL-safe since  $X$  does not occur in the only (positive) non-DL-atom of the rule, that is the rule head  $uncle(Y, Z)$ .

---

inferred knowledge, hence classical negation is monotonic. However, the default negated fact  $not C(o)$  is true in  $P$  under CWA such that adding the fact  $C(o)$  later to  $P \cup \{not C(o)\}$ , the previous inference  $not C(o)$  from  $P$  has to be retracted (belief revision of  $P$ ), hence default negation is nonmonotonic.

Unsafe rules can be made DL-safe by adding positive non-DL-atoms of the form  $O(X)$  (with special rule predicate  $O$  not defined in  $\phi$ ) to the rule body for each variable  $X$  violating the DL-safety condition, and a fact  $O(a)$  for each individual  $a$  in the ABox of  $\phi$ . Regarding the example above, the modified rule  $uncleOf(Z, Y) \leftarrow fatherOf(X, Y), brotherOf(X, Z), O(X)$  is DL-safe. DL-safety enforces evaluation of the rule with known individuals of the given ABox of  $\phi$  only, since the positive non-DL-atoms  $O$  only match with these but no anonymous individual implied by existential constraints.

### 2.5.3 Combination Strategies

As mentioned above, in the current Semantic Web architecture both ontology and rules layer are separated. Approaches to combining first-order description logic-based knowledge bases with rules from logic programming can be classified as follows [103, 105].

- Homogeneous integration (tight coupling) of both components under first-order semantics without any negation in the rules part of the combined knowledge base  $kb = (\phi, P)$ . In this setting, the rules part  $P$  is restricted to function-free Horn rules such that the semantics of  $kb$  can be defined by its translation to classical FOL. Examples are the decidable DLP (Description Logic Programming, cf. section 2.5.4), the Horn-SHIQ (cf. section 2.5.6) and DL-safe SWRL (cf. section 2.5.5), as well as undecidable SWRL.
- Homogeneous integration (tight coupling) of both components under stable model (answer-set) semantics of logic programming with default negation in  $P$  and classical negation in  $\phi$ . The (nonmonotonic) stable models of  $P$  are built with respect to first-order models of  $\phi$ , and constitute the stable models of the combined knowledge model  $kb$ . Examples are decidable DL+log (cf. section 2.5.7) and undecidable WSML-Rule (cf. section 2.4.3).
- Hybrid integration with strict semantic separation of both components (loose coupling) with allowed use of default negation in the rules part. In this setting, both components communicate via a safe interface but do not impose any syntactic restriction on each other. The DL part  $\phi$  is merely dealt with by the rules part  $P$  as an external source of information whose first-order semantics is treated separately which allows a mix of separate closed- and open-world reasoning. That involves special query predicates as positive DL-atoms in rule bodies in order to query the DL knowledge base from  $P$  for additional factual knowledge (ground entailments) such as evaluated concept memberships in (the ABox of)  $\phi$ . Prominent example is dl-programming with default negation under stable model semantics (cf. section 2.5.8).

## 2.5.4 DLP

The basic idea of description logic programming (DLP) introduced by Grosz et al. (2003)[143] is to intersect description logics with def-LP, that is equality-free, plain Datalog ( $H \leftarrow B_1 \wedge \dots \wedge B_m \wedge$ )<sup>50</sup> such that query answering of respectively combined knowledge bases is decidable. This is achieved by mapping as much as possible of OWL into the decidable function-free, definite Horn (def-Horn) which is the syntactic equivalent of def-LP in FOL. In fact, the first-order and minimal (w.r.t set inclusion) model semantics of def-Horn, respectively, def-LP coincide for positive ground entailments<sup>51</sup>.

For example, concept conjunction  $C_1 \sqcap C_2$  on both sides of DL inclusion axioms ( $C_1 \sqcap C_2 \sqsubseteq D$ ,  $D \sqsubseteq C_1 \sqcap C_2$ ) can be either directly or via Lloyd-Topor transformations from its FOL syntax to equivalent (set of) def-Horn rules<sup>52</sup>. Disjunctions on the right-hand-side of concept inclusion axioms cannot be handled within def-Horn, since no disjunction are allowed in Horn rule heads. Similarly, the translation of qualified universal and existential role restrictions into def-Horn is incomplete, while concept negation and role cardinality restrictions cannot be mapped to def-Horn.

The resulting description Horn logic (DHL), or OWL-DLP is a strong subset of not only def-Horn but SHIF (OWL-Lite), hence not very expressive for practical purposes. For example, without disjunctions or existential in the rule head, DLP cannot define a subconcept of a complex concept expression which is a disjunction (e.g.  $Human \sqcap Adult \sqsubseteq Man \sqcup Woman$ ), or a subclass of a complex class expression which is an existential (e.g.  $Radio \sqsubseteq \exists hasPart.Tuner$ ).

The semantics of the combined knowledge base  $(\phi, P)$  with  $\phi$  in DHL and  $P$  in def-Horn is defined by translation to the FOL fragment def-Horn. Examples of DLP reasoners are KAON-DLP and OWLIM.

## 2.5.5 SWRL

The Semantic Web rule language SWRL is a monotonic FOL extension of the description logic OWL-DL, i.e. SHOIN(D) with function-free Horn rules [351]. In particular, negation is allowed through OWL-DL axioms only (classical negation) but not in the rules part of the combined knowledge base. In contrast to the overcautious approach of DLP intersecting DL with def-Horn logic,

---

<sup>50</sup> These rules correspond to definite function-free Horn FOL clauses  $(\forall x. \neg B_1(x) \vee \dots \vee \neg B_n(x) \vee H(x))$  with universally quantified variable  $x$ .

<sup>51</sup> That is, the def-LP and the def-Horn rule sets entail exactly the same set of facts, though conclusions of def-Horn are not restricted to be facts but can also be rules (definite Horn clauses). In practical Semantic Web applications, however, often only factual inferences (ground entailments, conclusions in fact-form) are desired.

<sup>52</sup>  $C_1 \sqcap C_2 \sqsubseteq D$  corresponds to FOL  $\forall x. C_1(x) \wedge C_2(x) \rightarrow D(x)$  that is in def-Horn;  $D \sqsubseteq C_1 \sqcap C_2$  corresponds to  $\forall x. C_1(x) \wedge C_2(x) \leftarrow D(x)$  which is converted into set of def-Horn rules  $\{C_1(x) \leftarrow D(x), C_2(x) \leftarrow D(x)\}$ .

SWRL takes the other extreme of proposing the union of both but at the price of decidability.

A combined knowledge base  $kb = (\phi, P)$  in SWRL consists of a set of SHOIN(D) statements ( $\phi$ ), and a set of def-Horn rules ( $P$ ) with a combined signature  $\sigma_{kb}$  as union of the signatures ( $\sigma_\phi, \sigma_P$ ) of both parts. In fact, SWRL allows for full interaction between these parts (tight coupling): Any DL concept  $C$  and role  $R$  can be unrestrictedly used in rules by means of unary, respectively, binary DL atoms  $C(s)$  and  $R(s, t)$  with  $s, t$  constants or variables and predicates  $C$  and  $R$ . Concepts and role predicates (DL-atoms) may occur in the head and the body of rules without any restrictions<sup>53</sup>. There exist various SWRL builtins for numbers, strings, lists, times, etc.

Unlike DLP remaining in def-Horn logic, the combination of SHOIN(D) with function-free Horn rules increases the expressivity of both languages. For example, integrity constraints (Horn rules with empty head) and any statement that requires rules with arbitrarily shaped model such as the triangular model of the role "uncleOf" mentioned above cannot be represented in SHOIN(D) but in SWRL through Horn rules.

On the other hand, SWRL adds expressivity to Horn rules in terms of DL atoms in rules that are defined in SHOIN knowledge bases but are not expressible in Horn. For example, DL statements with concept negation, unqualified role cardinality constraints, and existential role restrictions on the right-hand-side of concept inclusions ( $D \subseteq \exists R.C$ )<sup>54</sup> that allow inferring the existence of anonymous individuals are not possible to express with function-free Horn rules<sup>55</sup>.

The semantics of a SWRL knowledge base  $kb$  is defined by translation of both  $\phi$  in SHOIN(D) and  $P$  in def-Horn (both decidable fragments of FOL) into first-order formulae (FOL clauses). That is, SWRL has a single integrated FO model which is the union of two models, one for the OWL-DL and one for the rules part, which share the same domain. Variables in SWRL rules universally quantify over both named and unnamed individuals in  $\phi$ . In other words, SWRL embeds rules and DL ontologies under the same FOL semantics (tight integration) but is restricted to monotonic (negation-free) Horn rules.

<sup>53</sup> SWRL rules are defined as new axioms in  $kb$  through  $r := \text{Implies}(H, B)$  with head  $H$  and body  $B$  as atoms of the form  $C(x)$ ,  $P(x, y)$ ,  $\text{sameAs}(x, y)$ ,  $\text{dataRange}(x)$ , or  $\text{differentFrom}(x, y)$ ,  $\text{individualvaluedPropertyID}(x, y)$ ,  $\text{datavaluedPropertyID}(x, y)$ ,  $\text{builtIn}()$  where  $C$  is a named OWL class,  $P$  is an OWL property, and  $x, y$  are either variables, OWL individuals or OWL data values from a given OWL ontology.

<sup>54</sup> Like in  $\text{Person} \subseteq \exists \text{father}.\top$ , or the concept of grand childs  $\exists \text{father}.\exists \text{father}.\text{Person} \sqsubseteq \text{GrandChild}$  used in the (DL-unsafe) Horn rule  $\text{BadChild} \leftarrow \text{GRandChild}(x, y), \text{parent}(x, y), \text{parent}(z, y), \text{hates}(x, z)$  [262]

<sup>55</sup> The reading of the latter in FOL, that is  $\forall x.\exists y.R(x, y) \wedge C(y) \leftarrow D(x)$ , cannot be transformed to def-Horn because of the conjunction (not definite) and the existentially quantified variable in the rule head (Horn requires all variables universally quantified at the outer level of a rule).

Main inferences over combined knowledge bases  $kb = (\phi, P)$  in SWRL under first-order semantics are (a) checking the satisfiability (consistency) of  $kb$ , and (b) query answering in terms of ground entailments, i.e. checking whether a ground atom (fact)  $\alpha$  holds in or is entailed by  $kb$  ( $kb \models \alpha$ ). Unfortunately, SWRL extends OWL-DL with unrestricted function-free Horn rules which causes its undecidability [231, 160].

However, (incomplete) reasoning in SWRL can be performed using any general first-order theorem provers like Vampire, as well as any PROLOG or Datalog engine (no default negation in SWRL). For example, Hoolet is an implementation of an OWL-DL reasoner that uses Vampire to support SWRL. Other systems that offer also SWRL support are the KAON2 system, the FUB SWRL engine, and even Pellet, a prominent OWL-DL reasoner.

### 2.5.6 Horn-SHIQ

In (Motik, Sattler & Struder, 2004)[262] the EXPTIME-complete description logic SHIQ [173] is extended with DL-safe disjunctive positive Datalog rules. In these rules, DL-atoms are restricted to concepts and primitive (atomic) roles only. Though function-free Horn is covered by disjunctive Datalog, this extension is called Horn-SHIQ. It lies between both extremes of homogeneously combining ontologies and rules under first-order semantics, that are DLP and SWRL with respect to both expressivity and computational complexity of query answering (w.r.t. positive ground entailment, and non-ground concept subsumption reduced to satisfiability).

Horn-SHIQ is far more expressive than DLP, since SHIQ provides concept constructors that, unlike in DLP, cannot be mapped into def-Horn logic. In addition to DLP (def-Horn), SHIQ supports classical negation ( $\neg$ ), role cardinality restrictions, unrestricted use of existential and universal role (value) restriction, and concept disjunction. Further, Horn and even more disjunctive Datalog covers def-Horn of DLP. As mentioned above, DL-safety does not restrict the expressivity of Horn-SHIQ but only restricts the evaluation of rules with DL predicates to individuals that are explicitly named in the SHIQ knowledge base to retain decidability [262].

In particular, SHIQ with DL-safe def-Horn is a decidable subset of undecidable SWRL\* (SHOIQ and def-Horn)[262]. SHIQ prohibits the use of nominals (O) and unqualified role cardinality restrictions (N) which allows to achieve optimal query answering for a significant portion of decidable DL-safe SWRL (SHOIN and def-Horn) <sup>56</sup>. The motivation of restricting the description of knowledge bases to SHIQ(D) with DL-safe rules is not only to retain decidability - which would have been the case already for more expressive DL-safe SWRL\* - but to provide a more efficient query answering mechanism [131, 222].

---

<sup>56</sup> The combined use of nominals, inverse roles and unqualified role cardinality restrictions cause an increase in complexity from EXPTIME to NEXPTIME [359].

This is achieved by reducing the SHIQ(D) knowledge base  $\phi$  to an equisatisfiable DL-safe disjunctive Datalog program  $DD(\phi)$  over which query answering in terms of positive ground entailments can be performed in polynomial data complexity (PTIME). In particular, the reduction proposed in [262, 260] guarantees that

- (a)  $kb = (\phi, P)$  is satisfiable if and only if  $DD(\phi) \cup P$  is satisfiable, and
- (b)  $kb \models \alpha$  (under FO semantics) if and only if  $DD(\phi) \cup P \models_c \alpha$  (under minimal model semantics) for positive facts (non-negated ground DL-atoms)  $\alpha$  of the form  $R(a, b)$  or  $C(o)$  with  $C$  concept,  $R$  primitive role and  $a, b, o$  named individuals. This query answering is safe, since for such positive ground entailments the first-order and minimal model semantics coincide.

It is noticeable, however, that the conversion of  $\phi$  into disjunctive Datalog, in particular the saturation of  $\phi$  with all grounded consequences related to the query is computed in EXPTIME with respect to the size of the knowledge base [131, 222]. This leads to a combined complexity of Horn-SHIQ in EXPTIME. Any other DL-safe rule in  $P$  with the above restrictions can be appended to this reduction of  $\phi$  to  $DD(\phi)$ .

According to the recent study [320], the naive use of LP reasoners like XSB for query answering transformed SHIQ(D) knowledge bases is infeasible for practical applications, whereas special DL/LP reasoners KAON2, OntoBroker and SCREECH showed reasonable performance. It is not yet clear whether this Horn-SHIQ approach also scales well to knowledge bases of size of several terabytes in practice [113] which is a general problem of all approaches to DL-based and rules reasoning.

### 2.5.7 DL+log

In DL+log (Rosati et al., 2006)[316], both components of a combined knowledge base  $kb = (\phi, P)$  are tightly coupled under nonmonotonic stable model (answer-set) semantics. DL predicates taken from  $\phi$  can arbitrarily occur in DL-safe rules of  $P$  (but not vice versa) with possible default negation in rules, and classical negation in  $\phi$ . Unlike DLP, the approach allows to integrate a knowledge base in an arbitrary DL with safe Datalog<sup>W</sup> rules and, unlike loosely coupled dl-programs, does not require special query atoms in rules for interaction.

The stable model of  $kb$  is of the form  $NM = I \cup M$  where  $I$  is a first-order model of  $\phi$ , and  $M$  a stable model of rules predicates after deletion of classical DL atoms that are satisfied by  $I$  in  $P$ . That is, for a given interpretation  $I$  of  $\phi$ , the rules in  $P$  are grounded and reduced with respect to  $I$  by eliminating (a) non-satisfied (wrt  $I$ ) DL atoms in rule heads, and (b) satisfied DL atoms in rule bodies. The resulting ground program  $P_I$  contains no DL atoms and can be assigned a stable model, if it exists. For an illustrative example of a DL+log knowledge base and a stable model  $NM$  ( $= M$  of  $P_I$ ) for given

interpretation  $I$  of  $\phi$  (there could be no stable models  $M$  for given  $\phi$ , we refer to [103]).

### 2.5.8 Nonmonotonic DLP

Nonmonotonic description logic programs, called dl-programs, are loosely coupled knowledge bases  $(\phi, P)$  that extend function-free normal logic programs  $P$  with queries to description logic knowledge bases  $\phi$ . This is achieved by means of special query atoms (DL atoms) to  $\phi$  in the bodies of rules in  $P$  of the form  $DL[S_1op_1p_1, \dots, S_mop_mp_m; Q](t)$  with query predicate  $Q$  [106, 107]. These special query atoms are used to check for ground entailment of concept membership in  $\phi$ . Query answering in dl-programs is decidable provided that it is decidable in  $\phi$ .

The minimal model semantics of a dl-program  $(\phi, P)$  is defined via grounding all rules in  $P$  with a set of constants in  $P$  and  $\phi$ . That is, the query atom  $DL[C](X)$  fetches the instances of concept  $C$  from the ABox of  $\phi$  by calling an external DL reasoner, and then grounds the rule (via variable  $X$ ) in  $P$  with these instances and those in  $P$ .

For example, the rule  $wineBottle(X) \leftarrow DL[Wine](X)$  gets evaluated over all (stable) models of  $P$  and the extension of the concept  $Wine$  in all first-order models of  $\phi$ . A query  $?-wineBottle(Topkapi)$  evaluates to true in  $P$  if the DL-atom  $DL[Wine](Topkapi)$  of the rule above holds in  $\phi$  ( $\phi \models Wine(Topkapi)$ ). The evaluation of  $DL["WhiteWine(+), mywhite, DryWine(-), semidry; "Wine"](X)$  adds all assertions  $WhiteWine(c)$  and  $\neg DryWine(c)$  to  $\phi$  such that  $mywhite(c)$ , respectively,  $semidry(c)$  holds in  $P$ . For more details on dl-programming with examples, we refer to [103, 106, 107].

#### *Standardized rule exchange format*

Research on Semantic Web rules is rapidly evolving but the recommended choice of a combined ontology rule language or unifying logic is still open. For the rules layer, the W3C issued the rule interchange format (RIF)<sup>57</sup> as a general purpose rule language to enable the sharing of rules across rule systems from different suppliers. RIF includes the RIF-Core language for expressing Horn rules. For more details on RIF, we refer to [44].

#### *Open problems and challenges*

Some open problems of reasoning with Semantic Web ontologies and rules are as follows.

- A general unifying logical framework to formally study the integration of monotonic DLs and nonmonotonic rules (e.g., the analysis of autoepistemic and modal logics for unifying FOL and LP [87], the integration of OWL-DL with DL-safe rules of autoepistemic logic MKNF[263], and embedding of LP in autoepistemic logic FOAEL [88]).

<sup>57</sup> [www.w3.org/2005/rules/](http://www.w3.org/2005/rules/)



- Practical software support for rule-based application building and sharing based on RIF, and field test experiments on the scalability of DL and rules reasoning.

## 2.6 Semantic Web Applications

Current major application domains of Semantic Web technology are knowledge management (KM), enterprise application integration (EAI), CSCW and social networking, infotainment, e-healthcare, life sciences, digital libraries, and astronomy. Though, the exploitation of Semantic Web technologies in each of these domains is different in nature.

For example, in typical KM applications to organize and provide tailored information inside or across enterprise boundaries, the Semantic Web technologies RDF, RDFS and SPARQL are used for document annotation, building of related ontologies, and RDF document querying, whereas the use of OWL and rules is rather rare. In contrast, in EAI or e-commerce applications the standard ontology language OWL is supposed to be heavily used to make the semantics of information explicit in ontologies, and rules are used to describe the relationships between them and/or to convert between different data formats while application database(s) can have RDF wrappers and data is published and exchanged in RDF.

In case of Semantic Web applications made available as a single or composed Web service, we are into the field of Semantic Web services - which is the topic of the next chapter. One popular application example in business is the dynamic classification of, and search for relevant business partners and experts in and across enterprises (Sheth et al., 2006)[332].

### *Deployed applications*

Remarkably, there has been quite a number of Semantic Web applications deployed to date of which the Semantic Web Challenge promotes the most innovative ones at [challenge.semanticweb.org](http://challenge.semanticweb.org). The following short list of Semantic Web application examples is representative but, of course, not exhaustive.

- One of the first and most widely known Semantic Web applications is the Friend-of-a-Friend (FoaF) social network which describes relationships among people in terms of homepage-like profiles in RDF.
- The system [Revyu.com](http://revyu.com) for jointly reviewing and rating literally anything ([revyu.com](http://revyu.com)) tagging the content in RDF was voted Semantic Web Challenge winner in 2007.
- The collaborative semantic music recommender [Foafing-the-music](http://foafing-the-music.iaa.upf.edu)<sup>58</sup>.
- The multimedial e-culture demonstrator<sup>59</sup> was voted Semantic Web Challenge winner in 2006.

<sup>58</sup> [foafing-the-music.iaa.upf.edu](http://foafing-the-music.iaa.upf.edu)

<sup>59</sup> [e-culture.multimedial.nl](http://e-culture.multimedial.nl)

- The TcmGrid<sup>60</sup> that semantically interconnects over 70 heterogeneous relational databases for traditional chinese medicine by use of a shared global OWL ontology with over 70 classes and 800 properties; it also provides semantic query, search and navigation services.
- The BioDASH<sup>61</sup> Semantic Web dashboard for drug development that uses rule-based RDF inferencing to filter and merge data in due course of associating relevant information on disease, drug progression stages, molecular biology and others for a team of users.
- The VSTO 1.0 (Virtual Solar Terrestrial Observatory)<sup>62</sup> astronomical application that provides semantically integrated access to two large, heterogeneous online scientific data repositories in the area of solar- and solar-terrestrial physics. Semantic information integration in VSTO is based on a globally shared special VSTO ontology in OWL for the CEDAR community of approximately 1200 participants.
- The semantic knowledge management platforms KIM and hTechSight KMP.
- The RichNews news agency application by BBC which semantically annotates and summarizes radio and television news broadcasts using resources retrieved from the Web.
- Vodafone's Live Mobile Portal allows users to more efficiently search for relevant RDF annotated resources, such as ringtones, games, and pictures which decreased the page views per download around 50 percent and number of downloaded ringtones up to 20 percent in only two months.

#### *Selected Semantic Web application projects*

From the set of Semantic Web application development projects world wide, we would like to point to, for example, Nepomuk<sup>63</sup>, SmartWeb<sup>64</sup>, NEWS<sup>65</sup>, MESH<sup>66</sup>, Sculpteur<sup>67</sup>, Helsinki IIT application projects<sup>68</sup>, and the relevant European Commission funded projects on the subject. Other projects focussing on semantic Web business data integration include DartGrid<sup>69</sup> from Zhe Jiang University in China, and those launched internally by Boeing, MITRE Corporation, and Elsevier.

For example, the German government funded project SmartWeb coordinated by DFKI was developed context-aware multimodal user interfaces for distributed and composable Semantic Web services on mobile devices. In particular,

<sup>60</sup> [ccnt.zju.edu.cn/projects/dartgrid/tcmgrid.html](http://ccnt.zju.edu.cn/projects/dartgrid/tcmgrid.html)

<sup>61</sup> [www.w3.org/2005/04/swls/BioDash/Demo/](http://www.w3.org/2005/04/swls/BioDash/Demo/)

<sup>62</sup> [www.vsto.org](http://www.vsto.org)

<sup>63</sup> [nepomuk.semanticdesktop.org/xwiki/bin/Main1/](http://nepomuk.semanticdesktop.org/xwiki/bin/Main1/)

<sup>64</sup> [smartweb.dfki.de/main\\_pro\\_en.pl?infotext\\_en.html](http://smartweb.dfki.de/main_pro_en.pl?infotext_en.html)

<sup>65</sup> [www.news-project.com](http://www.news-project.com)

<sup>66</sup> [cordis.europa.eu/ist/kct/fp6\\_mesh.htm](http://cordis.europa.eu/ist/kct/fp6_mesh.htm)

<sup>67</sup> [www.sculpteurweb.org](http://www.sculpteurweb.org)

<sup>68</sup> [www.seco.tkk.fi/applications/](http://www.seco.tkk.fi/applications/)

<sup>69</sup> [ccnt.zju.edu.cn/projects/dartgrid/](http://ccnt.zju.edu.cn/projects/dartgrid/)

advancements of both mobile broadband communication and Semantic Web technology were exploited to realize such services with innovative personalization and localization features. The implemented SmartWeb application demonstrators are (a) the multi-modal dialogue-based personal guide for the 2006 FIFA world cup in Germany which provides a mobile infotainment service to soccer fans on mobile devices such as the MDA pro III/IV, and (b) safety information services for drivers of a Mercedes A-class car and a BMW K1200LT motor bike.

## 2.7 Critique

One major problem of realizing the Semantic Web is to effectively achieve an automated semantic integration of distributed, heterogeneous data, information and services in the Web on demand. From a historical perspective, the Semantic Web can even be seen as a kind of rebirthed idea that has been used to cope with the same problem of semantic interoperation in the domains of federated and multi-database systems, and cooperative information systems already two decades ago.

*What is new with the Semantic Web?*

Provocatively speaking, the principled use of description logics for automated logic-based reasoning about heterogeneous, semantically annotated resources and formal ontologies was deeply investigated in the context of semi-automated database schema integration since the mid 1980s, while the use of intelligent agents to manage and coordinate them is a proclaimed key feature of the paradigm of intelligent cooperative information systems introduced by Papazoglou, Selis and Laufman already in 1992. The tremendous progress in research and development of Semantic Web technologies since the beginning of this century would not have been possible without heavily building on relevant results achieved and practical experiences made in these and other related fields.

*Where shall the critical mass of semantic metadata come from?*

One key to the success of the Semantic Web in practice is the massive production and maintenance of standard-based formal data semantics, ontologies and related instance sets or knowledge bases. However, as both Tom Gruber and Tim Berners-Lee pointed out during their keynote talks at the ISWC 2006 conference, the availability of a cost-free, critical mass of Semantic Web data and ontologies to the common user is still far from being achieved. As a consequence, apart from theoretic research challenges and required funding of relevant science world wide, creating this critical mass is considered to be one major barrier for developing Semantic Web business applications today.

In other words, bringing a true Semantic Web to the world of common Web users still is kind of a chicken-and-egg problem. Until there is enough Web data attached with metadata in standard RDF or OWL to make it meaningful, and without easy to use software support for resource annotation and processing the semantic metadata, apart from academics and research projects, nobody is going to be able to create any interesting services on the fly for everyday tasks. At the ISWC 2007 conference, there has been some vivid discussion on possible incentive schemes to push both the development and widespread use of easy to use tools for semantic annotation of Web resources by the common user but still, in essence, without any common agreement by the community participating in this discussion - so, this essential problem remains to be solved yet.

*Do we really need formal ontologies to share?*

Another main criticism of the Semantic Web even questions its practical feasibility and added value for business applications in general, from both the technical and the human user perspective. For example, the simple Web resource format XHTML or task specific microformats such as hCard, XFN and XOXO for contact information, social relationships, lists and outlines, respectively, are argued for their more ease of creation and use in specific Web 2.0 services and applications like RSS-based newsfeeds and podcasts.

The same goes with individual user-generated tags, so-called folksonomic categories that allow for rather idiosyncratic tagging of collaboratively shared content of social Web 2.0 application services like flickr, mySpace and facebook. As with agreed-upon XML-namespaces for whole application domains like chemistry and mathematics, the use of such popular, informal folksonomies instead of formal logic-based ontologies in RDFS or OWL are often argued to better suit the needs of individual service specific user communities in practice.

*Is logic-based reasoning only appropriate?*

Regarding the original Semantic Web idea of logic-based reasoning on annotated Web resources, it is often argued for an alternative, a so-called statistical Semantic Web. The basic idea of this alternative is to live with semantic annotations in structured XML and text, and perform approximative reasoning upon such metadata by means of, for example, linguistic similarity measurement or probabilistic XML information retrieval. In other words, if one does accept the hypothesis that it will take smart software to produce the markup that the Semantic Web and agents will exploit, then what is the case for believing that it will be ontology-based logical inference engines rather than statistically-based heuristic search engines making use of already widespread XML metadata that people will be using? At the current stage, we cannot tell until solid and publicly available user studies on the subject are conducted.

In fact, the same argument in principle holds for the area of Semantic Web services which we will introduce in the next chapter.

Notably, the original idea to base the Semantic Web on (description) logics only got even questioned by the Semantic Web community itself quite recently. For example, in (van Harmelen & Fensel, 2007)[113] it is acknowledged that the basic underlying assumptions of proposed (crisp) logical, complete and correct reasoning approaches do not seem to match the reality the Web provides. Such limiting assumptions include the restriction of the sets of (ontological/rule) axioms and facts in size, the axioms and facts are static and known in advance, and the inference process must be sound and complete. In fact, Semantic Web research now tends to revise these limiting assumptions of the original vision in favor of approximate reasoning that promises to better scale to the the Web. As mentioned above (cf. section 2.4.2), the discussion on related issues and open problems like belief revision in the Semantic Web, probabilistic notions of entailment, desirable properties of such inferencing (monotonic vs. nonmonotonic, anytime availability, etc.), and appropriate query languages is ongoing.

*What are mature technologies for easy adoption by the common user?*

In practice, the Semantic Web today provides standardized technologies that aim at the easy integration of semantically annotated Web data: It includes a standardized abstract model (RDF) for relational resource graphs, plus efficient means to extract RDF information from XML, XHTML or microformatted pages (GRDDL), plus means to add structured information to XHTML pages (RDFa), plus a query language adapted for RDF graphs (SPARQL), and standardized ontology languages to categorize resources such as RDFS and OWL. Apart from the rule interchange format RIF, reasoning means for formal ontologies with (monotonic or nonmonotonic) rules like for SWRL, DLP, or WSML-Rule are not standardized.

Depending on the complexity required, applications may choose among these technologies ranging from rather simple RDFS to more sophisticated ones like OWL and rules, and reuse existing ontologies that others have produced such as DOLCE, WINE, SUMO, SEMINTEC, GALEN. Though, some of these technologies are stable, others are being under development still. Moreover, as mentioned above, there is even a current major shift in research of the Semantic Web community towards approximate, scalable reasoning on semantically annotated Web resources of which no one knows the results yet.

*What about the future of Semantic Web technologies?*

Taking this and the lack of easy-to-use Semantic Web tools for annotation, reasoning, and application building for the common Web user into account, it comes at no surprise that the uptake of Semantic Web technology world wide appears slow. For example, a recent survey of the Semantic Web performed by use of a specific search engine Swoogle revealed that there merely exist

around 800K RDF annotated resources in the visible Web. This size is of some magnitudes smaller than the one of the Web with its estimated 12 billion resources indexed by the search engine Google alone.

Besides, the technology survey 2006 by market analyst Gartner expects Semantic Web science to take further five to ten years for reaching a level of maturity that is sufficient enough for possible commercial uptake by major business stakeholders beyond RDF and OWL, not to speak about making it to the common Web user broadly. The same survey also predicts the convergence of Semantic Web technologies with peer to peer (P2P) computing (Stuckenschmidt & Staab, 2006)[345], service-oriented computing, and pervasive computing. In fact, first research and development efforts towards the ubiquitous Semantic Web (ako Web 3.0++) by merging Web 2.0 ideas of practical everyday life and social collaborative applications with ambient intelligence and Semantic Web technologies for future (intra- and inter-organizational) service applications anytime, anywhere are already underway.

## 2.8 Further Readings

For a comprehensive and more detailed coverage of the Semantic Web, we refer the interested reader to, for example, the excellent readings on the subject [114, 11]. Advancements in the field are regularly reported in the proceedings of major conferences like the International Semantic Web Conference (ISWC, since 2002), and the European Semantic Web Conference (ESWC, since 2004). Examples of major funded research projects that significantly contributed to the advancement of theory and application of the Semantic Web are KnowledgeWeb<sup>70</sup>, DIP<sup>71</sup>, SEKT<sup>72</sup>, REVERSE<sup>73</sup>, SUPER<sup>74</sup>, and SmartWeb<sup>75</sup>. Besides, the Web sites of relevant W3C working groups <sup>76</sup>, and the DFKI competence center for the Semantic Web (semanticweb.dfki.de) serve as good starting points for obtaining references to topical research and development in the field.

---

<sup>70</sup> [knowledgeweb.semanticweb.org](http://knowledgeweb.semanticweb.org)

<sup>71</sup> [dip.semanticweb.org](http://dip.semanticweb.org)

<sup>72</sup> [www.sekt-project.org](http://www.sekt-project.org)

<sup>73</sup> [reverse.net](http://reverse.net)

<sup>74</sup> [www.ip-super.org](http://www.ip-super.org)

<sup>75</sup> [smartweb.dfki.de/main\\_pro\\_en.pl?infotext\\_en.html](http://smartweb.dfki.de/main_pro_en.pl?infotext_en.html)

<sup>76</sup> [www.w3.org/2001/sw/](http://www.w3.org/2001/sw/)

---

## Semantic Web Services

Industrial standards for XML-based Web services (cf. chapter 1) are designed to represent information about the interfaces of services, how they are deployed, and how to invoke them. They are, however, insufficient to represent the semantics of a Web service, that are its capabilities and requirements in a machine-understandable, logic-based form to enable automated semantic service discovery, composition planning and integration. This major challenge of semantic service coordination carried out by intelligent software agents has been addressed by the Semantic Web community introducing Semantic Web services (SWS) technology as one manifest of the convergence of Semantic Web and service-oriented computing.

This chapter briefly discusses prominent Semantic Web service description frameworks, that are the standard SAWSDL, OWL-S and WSML<sup>1</sup>, and Semantic Web service service coordination activities each of which discussed in more detail in the following parts. This is complemented by main critics of Semantic Web services, and selected references to further readings on the subject. This chapter is an extended version of (Klusch, 2008)[195].

### 3.1 Issues of Semantic Service Description

Each semantic service description framework can be characterised with respect to (a) what kind of service semantics are described, (b) in what language or formalism, (c) allowing for what kind of reasoning upon the abstract service descriptions? Further, we distinguish between an abstract Web service, that is the description of the computational entity of the service, and a concrete service as one of its instances or invocations that provide the actual value to the user (Preist, 2007)[299]. In this sense, abstract service descriptions are considered complete but not necessarily correct: There might be concrete

---

<sup>1</sup> Due to space limitations other description frameworks like SWSL (Semantic Web Service Language) and project specific formats like DIANE are omitted.

service instances that are models of the capability description of the abstract service but can actually not be delivered by the provider.

### **3.1.1 Parts of Service Semantics**

In general, the functionality of a service can be described in terms of what it does, and how it actually works. Both aspects of its functional semantics (or capability) are captured by a service profile, respectively, service process model. The profile describes the signature of the service in terms of its input (I) and output (O) parameters, and its preconditions (P) and effects (E) that are supposed to hold before or after executing the service in a given world state, and some additional provenance information such as the service name, its business domain and provider. The process model of atomic or composite services describes how the service works in terms of the interplay between data and control flow based on a common set of workflow or control constructs like sequence, split+join, choice, and others.

This general distinction between profile and process model semantics is common to structured Web service description frameworks, while differences are in the naming and formal representation of what part of service semantics. We can further differentiate between stateless (IO), respectively, state-based (PE) abstract service descriptions representing the set of its instances, that are concrete services providing value to the user. The non-functional service semantics are usually described with respect to a quality of service (QoS) model including delivery constraints, cost model with rules for pricing, repudiation, availability, and privacy policy.

### **3.1.2 Structured Representation**

A domain-independent and structured representation of service semantics is offered by upper (top-level) service ontologies and languages such as OWL-S and WSML with formal logic groundings, or SAWSDL which comes, in essence, without any formal semantics. Neither OWL-S nor WSML provide any agreed formal but intuitive, standard workflow-based semantics of the service process model (orchestration and choreography). Alternatively, for abstract service descriptions grounded in WSDL, the process model can be intuitively mapped to BPEL orchestrations with certain formal semantics.

### **3.1.3 Monolithic Logic-Based Representation**

The formal specification of service semantics agnostic to any structured service description format can be achieved, for example, by means of a specific set of concept and role axioms in an appropriate logic (cf. section 3.5). Since the service capability is described by means of one single service concept, this representation of service semantics is called monolithic and allows to



determine the semantic relations between service descriptions fully within the underlying logical formalism based on concept satisfaction, subsumption and entailment. However, it does not provide any further information on how the service actually works in terms of the process model nor any description of non-functional semantics.

#### **3.1.4 Data Semantics**

The domain-dependent semantics of service profile parameters (also called data semantics) are described in terms of concepts, roles (and rules) taken from shared domain, task, or application ontologies. These ontologies are defined in a formal Semantic Web language like OWL, WSML or SWRL. If different ontologies are used, agents are supposed to automatically resolve the structural and semantic heterogeneities for interoperation to facilitate better Web service discovery and composition. This process of ontology matching is usually restricted to ontologies specified in the same language, otherwise appropriate inter-ontology mappings have to be provided to the agents.

In subsequent sections, we briefly introduce prominent approaches to both types of service descriptions. For structured semantic service descriptions, we focus on OWL-S, WSML, and SAWSDL, and omit to discuss alternatives like DSD (DIANE service description format) and SWSL (Semantic Web Service Language).

#### **3.1.5 Reasoning on Service Semantics**

The basic idea of formally grounded descriptions of Web services is to allow agents to better understand the functional and non-functional semantics through appropriate logic-based reasoning. For this purpose, it is commonly assumed that the applied type of logic reasoning complies with the underlying semantic service description framework. Further, the concept expressions used to specify the data semantics of service input and output parameters are assumed to build up from basic concepts and roles taken from formal application or domain ontologies which the requester and provider commonly refer to. We survey approaches to non-logic based, logic based, and hybrid reasoning means for Semantic Web service discovery, and composition planning in the introductions to the following parts of this work.

### **3.2 SAWSDL**

The standard language WSDL for Web services operates at the mere syntactic level as it lacks any declarative semantics needed to meaningfully represent and reason upon them by means of logical inferencing. In a first response to this problem, the W3C Working Group on Semantic Annotations for WSDL

and XML Schema (SAWSDL) developed mechanisms with which semantic annotations can be added to WSDL components. The SAWSDL specification became a W3C candidate recommendation on January 26, 2007<sup>2</sup>, and eventually a W3C recommendation on August 28, 2007.

### 3.2.1 Annotating WSDL Components

Unlike OWL or WSML, SAWSDL does not specify a language for representing formal ontologies but provides mechanisms by which ontological concepts that are defined outside WSDL service documents can be referenced to semantically annotate WSDL description elements. Based on its predecessor and W3C member submission WSDL-S<sup>3</sup> in 2005, the key design principles for SAWSDL are that (a) the specification enables semantic annotations of Web services using and building on the existing extensibility framework of WSDL; (b) it is agnostic to semantic (ontology) representation languages; and (c) it enables semantic annotations for Web services not only for discovering Web services but also for invoking them.

Based on these design principles, SAWSDL defines the following three new extensibility attributes to WSDL 2.0 elements for their semantic annotation:

- An extension attribute, named **modelReference**, to specify the association between a WSDL component and a concept in some semantic (domain) model. This **modelReference** attribute is used to annotate XML Schema complex type definitions, simple type definitions, element declarations, and attribute declarations as well as WSDL interfaces, operations, and faults.
- Two extension attributes, named **liftingSchemaMapping** and **loweringSchemaMapping**, that are added to XML Schema element declarations, complex type definitions and simple type definitions for specifying mappings between semantic data in the domain referenced by **modelReference** and XML. These mappings can be used during service invocation.

An example of a SAWSDL service, that is a semantically annotated WSDL service with references to external ontologies describing the semantics of WSDL elements, is given in figure 3.1: The semantics of the service input parameter of type "OrderRequest" is defined by an equally named concept specified in an ontology "purchaseorder" which is referenced (URI) by the element tag "modelReference" attached to "OrderRequest". It is also annotated with a tag A tag "loweringSchemaMapping" which value (URI) points to a data type mapping, in this case an XML document, which shows how the elements of this type can be mapped from the referenced semantic data model (here RDFS) to XMLS used in WSDL.

---

<sup>2</sup> [www.w3.org/2002/ws/sawsdl/](http://www.w3.org/2002/ws/sawsdl/)

<sup>3</sup> [www.w3.org/Submission/WSDL-S/](http://www.w3.org/Submission/WSDL-S/)

```

<wsdl:description targetNamespace="http://www.w3.org/2002/ws/sawSDL/spec/wsdl/order#"
  xmlns="http://www.w3.org/2002/ws/sawSDL/spec/wsdl/order#"
  xmlns:wsdl="http://www.w3.org/ns/wsdl"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:sawSDL="http://www.w3.org/ns/sawSDL">

  <wsdl:types>
    <xs:schema targetNamespace="http://www.w3.org/2002/ws/sawSDL/spec/wsdl/order#" elementFormDefault="qualified">
      <xs:element name="OrderRequest"
        sawSDL:modelReference="http://www.w3.org/2002/ws/sawSDL/spec/ontology/purchaseorder#OrderRequest"
        sawSDL:loweringSchemaMapping="http://www.w3.org/2002/ws/sawSDL/spec/mapping/RDFOnt2Request.xml">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="customerNo" type="xs:integer" />
            <xs:element name="orderItem" type="item" minOccurs="1" maxOccurs="unbounded" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      ...
    </xs:schema>
  </wsdl:types>

  <wsdl:interface name="Order"
    sawSDL:modelReference="http://example.org/categorization/products/electronics">
    <wsdl:operation name="order" pattern=" http://www.w3.org/ns/wsdl/in-out"
      sawSDL:modelReference="http://www.w3.org/2002/ws/sawSDL/spec/ontology/purchaseorder#RequestPurchaseOrder">
      <wsdl:input element="OrderRequest" />
      <wsdl:output element="OrderResponse" />
    </wsdl:operation>
  </wsdl:interface>
</wsdl:description>

```

**Fig. 3.1.** Example of semantic annotation of WSDL elements in SAWSDL.

### 3.2.2 Limitations

Major critique of SAWSDL is that it comes, as a mere syntactic extension of WSDL, without any formal semantics. In opposite to OWL-S and (in part) WSMML, there is no defined formal grounding of neither the XML-based WSDL service components nor the referenced external metadata sources (via model-Reference). Quoting from the SAWSDL specification (section 2.2): "Again, if the XML structures expected by the client and by the service differ, schema mappings can translate the XML structures into the semantic model where any mismatches can be understood and resolved." This makes any form of logic-based discovery and composition of SAWSDL service descriptions in the Semantic Web rather obsolete but calls for "magic" mediators outside the framework to resolve the semantic heterogeneities.

Another problem with SAWSDL today is its very limited software support. Notable exceptions are the implemented SAWSDL service discovery and com-

position planning means of the METEOR-S framework (MWSDI). However, the recent announcement of SAWSDL as a W3C recommendation does not only support a standardized evolution of the W3C Web service framework in principle (rather than a revolutionary technology switch to far more advanced technologies like OWL-S or WSML) but will push software development in support of SAWSDL and reinforce research on refactoring these frameworks with respect to SAWSDL.

### 3.3 OWL-S

OWL-S is an upper ontology used to describe the semantics of services based on the W3C standard ontology OWL and is grounded in WSDL. It has its roots in the DAML Service Ontology (DAML-S) released in 2001, and became a W3C candidate recommendation in 2005. OWL-S builds on top of OWL and consists of three main upper ontologies: the Profile, Process Model, and Grounding (cf. figure 3.2). In the following, we briefly present each of the main



Fig. 3.2. OWL-S service description elements.

elements of OWL-S service descriptions. The underlying standard ontology language OWL has been introduced in the previous chapter.

#### 3.3.1 Service Profile

The OWL-S profile ontology is used to describe what the service does, and is meant to be mainly used for the purpose of service discovery. An OWL-S service profile encompasses its functional parameters, i.e. `hasInput` and `hasOutput`, and precondition and effect (IOPEs), as well as non-functional parameters such as `serviceName`, `serviceCategory`, `qualityRating`, `textDescription`,

and meta-data (actor) about the service provider and other known requesters. Please note that, in contrast to OWL-S 1.0, in OWL-S 1.1 the service IOPE parameters are defined in the process model with unique references to these definitions from the profile (cf. figure 3.3).

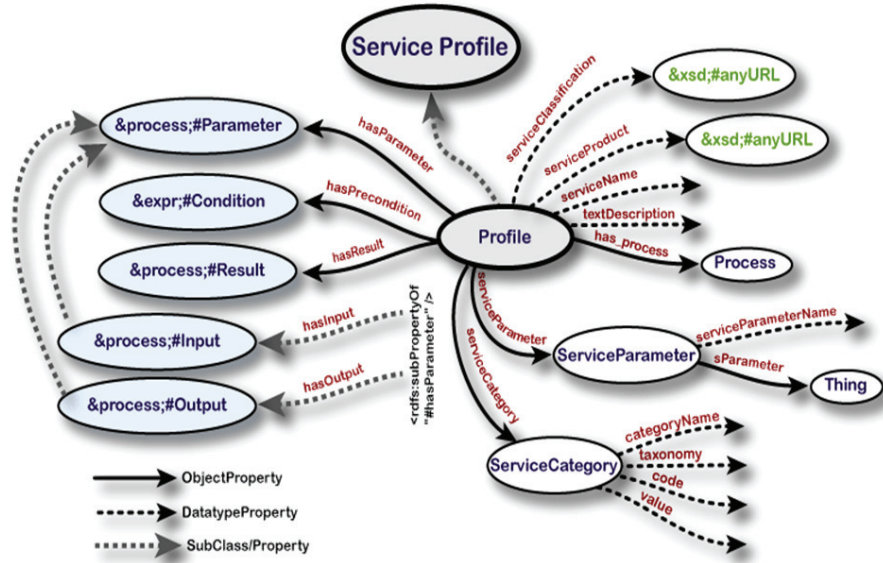


Fig. 3.3. OWL-S service profile structure.

Inputs and outputs relate to data channels, where data flows between processes. Preconditions specify facts of the world (state) that must be asserted in order for an agent to execute a service. Effects characterize facts that become asserted given a successful execution of the service in the physical world (state). Whereas the semantics of each input and output parameter is defined as an OWL concept formally specified in a given ontology, typically in decidable OWL-DL or OWL-Lite, the preconditions and effects can be expressed in any appropriate logic (rule) language such as KIF, PDDL, and SWRL. Besides, the profile class can be subclassed and specialized, thus supporting the creation of profile taxonomies which subsequently describe different classes of services. An example of a Semantic Web service profile in OWL-S 1.1 is given in figure 3.4.

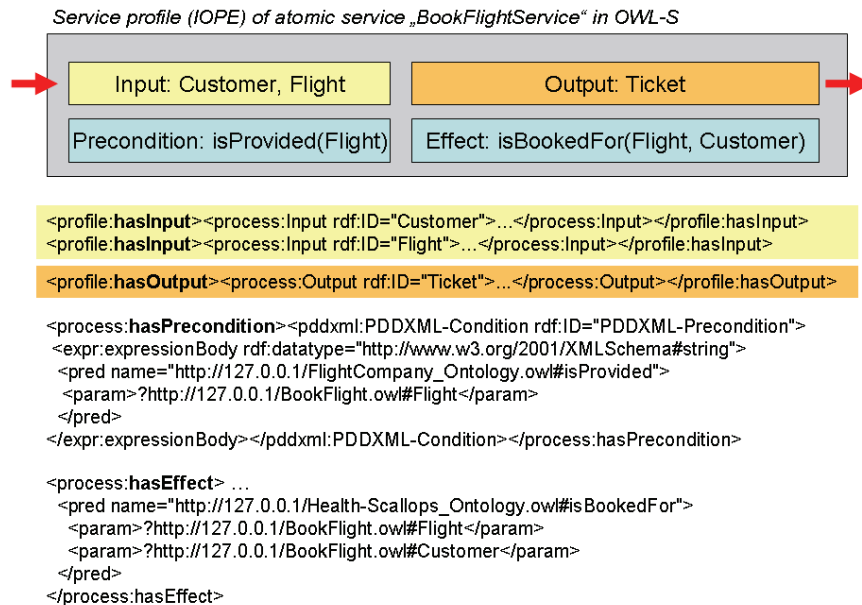


Fig. 3.4. Example of OWL-S 1.1 service profile.

### 3.3.2 Service Process Model

An OWL-S process model describes the composition (choreography and orchestration) of one or more services, that is the controlled enactment of constituent processes with respective communication pattern. In OWL-S this is captured by a common subset of workflow features like split+join, sequence, and choice (cf. figure 3.5). Originally, the process model was not intended for service discovery but the profile by the OWL-S coalition.

More concrete, a process in OWL-S can be atomic, simple, or composite. An atomic process is a single, black-box process description with exposed IOPEs. Simple processes provide a means of describing service or process abstractions which have no specific binding to a physical service, thus have to be realized by an atomic process, e.g. through service discovery and dynamic binding at runtime, or expanded into a composite process. The process model of the example OWL-S service above is provided in figure 3.6.

Composite processes are hierarchically defined workflows, consisting of atomic, simple and other composite processes. These process workflows are constructed using a number of different control flow operators including Sequence, Unordered (lists), Choice, If-then-else, Iterate, Repeat-until, Repeat-while, Split, and Split+Join. In OWL-S 1.1, the process model also specifies the inputs, outputs, preconditions, and effects of all processes that are part

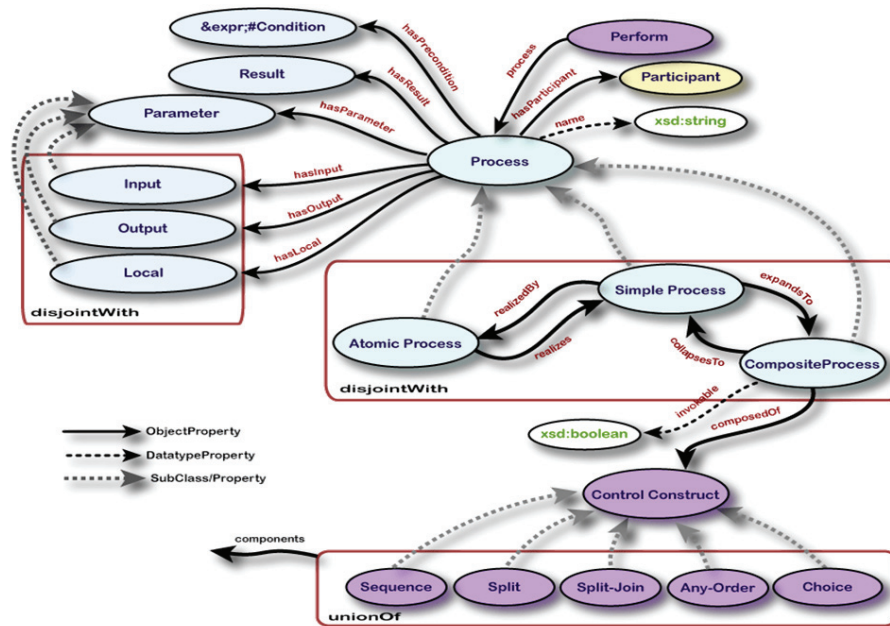


Fig. 3.5. OWL-S service process model.

of a composed service, which are referenced in the profiles of the respective services.<sup>4</sup> An OWL-S process model of a composite service can also specify that its output is equal to some output of one of its subprocesses whenever the composite process gets instantiated. Moreover, for a composite process with a Sequence control construct, the output of one subprocess can be defined to be an input to another subprocess (binding). Finally, OWL-S allows to specify conditional outputs (*inCondition*).

Unfortunately, the semantics of the OWL-S process model are left undefined in the official OWL-S documents. Though there are proposals to specify these semantics in terms of, for example, the situation calculus, and the logic programming language GOLOG based on this calculus [252].

### 3.3.3 Service Grounding

The grounding of a given OWL-S service description provides a pragmatic binding between the logic-based and XMLS-based service definitions for the purpose of facilitating service execution. Such a grounding of OWL-S services can be, in principle, arbitrary but has been exemplified for a grounding in

<sup>4</sup> This is in opposite to OWL-S 1.0, where the IOPES are defined in the profile and referenced in the process model.

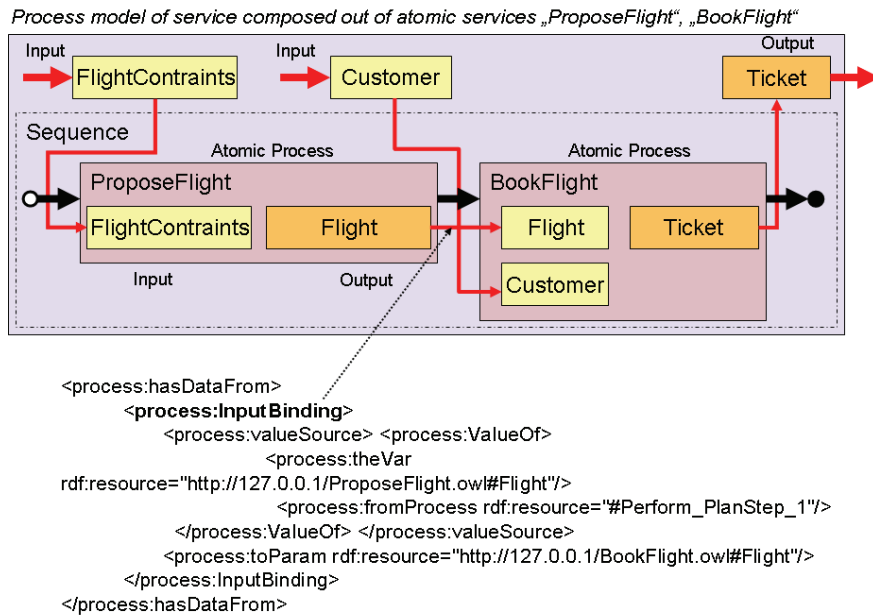


Fig. 3.6. Example of OWL-S service process model.

WSDL to pragmatically connect OWL-S to an existing Web service standard (cf. figure 3.7).

In particular, the OWL-S process model of a service is mapped to a WSDL description through a thin (incomplete) grounding: Each atomic process is mapped to a WSDL operation, and the OWL-S properties used to represent inputs and outputs are grounded in terms of respectively named XML data types of corresponding input and output messages. Unlike OWL-S, WSDL cannot be used to express pre-conditions or effects of executing services. Any atomic or composite OWL-S service with a grounding in WSDL is executable either by direct invocation of the (service) program that is referenced in the WSDL file, or by a BPEL engine that processes the WSDL groundings of simple or orchestrated Semantic Web services.

### 3.3.4 Software Support

One prominent software portal of the Semantic Web community is SemWeb-Central<sup>5</sup> developed by InfoEther and BBN Technologies within the DAML program in 2004 with BBN continuing to maintain it today. As a consequence, it comes at no surprise that this portal offers a large variety of tools for OWL

<sup>5</sup> <http://projects.semwebcentral.org/>



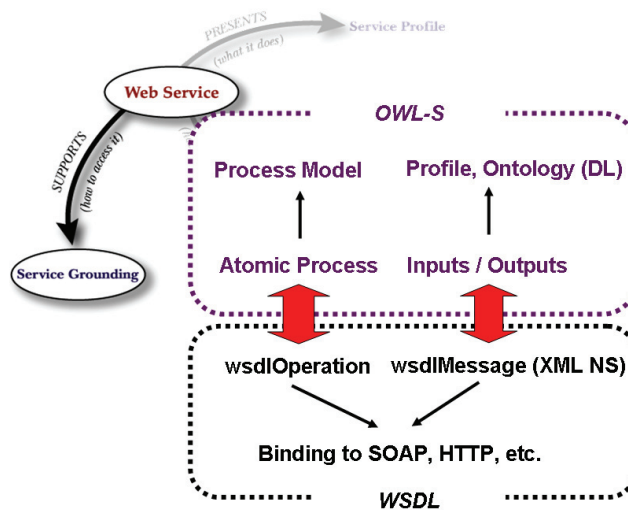


Fig. 3.7. Grounding of OWL-S in WSDL.

and OWL-S service coordination as well as OWL and rule processing. Examples of publicly available software support of developing, searching, and composing OWL-S services are as follows.

- *Development.*  
OWL-S IDE integrated development environment<sup>6</sup>, the OWL-S 1.1 API<sup>7</sup> with the OWL-DL reasoner Pellet<sup>8</sup> and OWL-S editors.
- *Discovery.*  
OWL-S service matchmakers OWLS-UDDI<sup>9</sup>, OWLSM<sup>10</sup> and OWLS-MX<sup>11</sup> with test collection OWLS-TC2.
- *Composition.*  
OWL-S service composition planners OWLS-XPlan<sup>12</sup>, GOAL ([www.smartweb-project.de](http://www.smartweb-project.de)).

### 3.3.5 Limitations

Main critique of OWL-S concern its limited expressiveness of service descriptions in practice which, in fact, corresponds to that of its underlying descrip-

<sup>6</sup> <http://projects.semwebcentral.org/projects/owl-s-ide/>  
<sup>7</sup> <http://projects.semwebcentral.org/projects/owl-s-api/>  
<sup>8</sup> <http://projects.semwebcentral.org/projects/pellet/>  
<sup>9</sup> <http://projects.semwebcentral.org/projects/mm-client/>  
<sup>10</sup> <http://projects.semwebcentral.org/projects/owls-m/>  
<sup>11</sup> <http://projects.semwebcentral.org/projects/owls-mx/>  
<sup>12</sup> <http://projects.semwebcentral.org/projects/owls-xplan/>

tion logic OWL-DL (cf. chapter 2). Only static and deterministic aspects of the world can be described in OWL-DL, since it does not cover any notion of time and change, nor uncertainty. OWL-S allows specifying conditional effects, that are possible effects of the service each of which conditioned by its result (output) but not input. Besides, in contrast to WSDL, an OWL-S process model cannot contain any number of completely unrelated operations. However, OWL-S bases on existing W3C Web standards, in particular the Web services protocol stack: It extends OWL and has a grounding in WSDL. Furthermore, the large set of available tools and applications of OWL-S services, as well as ongoing research on Semantic Web rule languages on top of OWL such as SWRL and variants still support the adoption of OWL-S for Semantic Web services, though this might be endangered by the choice of SAWSDL as a W3C standard just recently.

### 3.4 WSML

The conceptual model WSMO (Web Service Modelling Ontology), in particular the different variants of the formal ontology and Semantic Web service description language WSML has been introduced in the previous chapter. In this section, we briefly describe how goals and services are described in WSML in more detail.

WSML is particularly designed for describing a Semantic Web service in terms of its functionality (service capability), imported ontologies in WSML, and the interface through which it can be accessed for the purpose of orchestration and choreography. The formal semantics of elements within the description of goals and services capabilities (pre- and postconditions) are specified as logical axioms and constraints in ontologies using one of five WSML variants (cf. chapter 2, section 2.4.3). In general, the description of the semantics of a service and request (goal) in WSML is structured into the parts of the service capability, the service interface used for orchestration and choreography, and the shared variables.

#### 3.4.1 Goal

Like in OWL-S, a goal in WSMO represents the desired WSML service which is indicated with a special keyword "goal" instead of "webservice" in front of the service description. A goal refers to a desired state that can be described by help of a (world state) ontology. Such an ontology provides a basic vocabulary for specifying the formal semantics of service parameters and transition rules (TBox), and a set of concept and role instances (ABox) which may change their values from one world state to the other. It also specifies possible read-write access rights to instances and their grounding. A state is the dynamic set of instances of concepts, relations and functions of given state ontology at a certain point of time. The interpretation of a goal (and service) in WSML

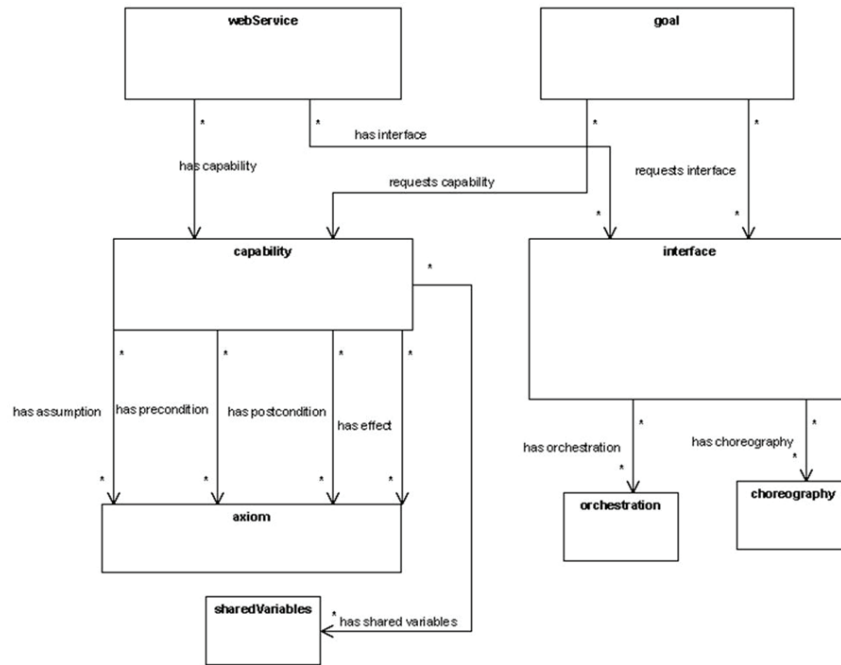


Fig. 3.8. WSM service and goal description.

is not unique: The user may want to express that either all, or only some of the objects that are contained in the described set are requested (Keller et al., 2005) [185].

```

namespace { _"http://example.org/goals#", dc _"http://purl.org/dc/elements/1.1#",
tr _"http://example.org/tripReservationOntology",
wsm _"http://www.wsmo.org/wsm/wsm-syntax#",
loc _"http://www.wsmo.org/ontologies/locationOntology#"}

goal _"http://example.org/havingATicketReservationInnsbruckVenice"
importsOntology { _"http://example.org/tripReservationOntology",
_"http://www.wsmo.org/ontologies/locationOntology"}

capability
postcondition definedBy
?reservation[ reservationHolder hasValue ?reservationHolder,
Item hasValue ?ticket ]
memberOf tr#reservation and
?ticket[ trip hasValue ?trip ] memberOf tr#ticket and
?trip [ origin hasValue loc#innsbruck, destination hasValue loc#venice ]
memberOf tr#trip.

```

Fig. 3.9. Example of a service request (goal) in WSM.

Figure 3.9 gives an example of a goal in WSML to find a service, which as a result of its execution, offers to reserve a ticket for the desired trip. In this case, the only element of the capability the user is interested in, is the postcondition of the desired service.

### 3.4.2 Service Capability

A WSML service capability describes the state-based functionality of a service in terms of its precondition (conditions over the information space), postcondition (result of service execution delivered to the user), assumption (conditions over the world state to met before service execution), and effect (how does the execution change the world state). Roughly speaking, a WSML service capability consists of references to logical expressions in a WSML variant that are named by the scope (precondition, postcondition, assumption, effect, capability) they intend to describe. It also specifies non-functional properties and all-quantified shared variables (with the service capability as scope) for which the logical conjunction of precondition and assumption entails that of the postcondition and the effect.

```

capability BookTicketCapability
sharedVariables {?creditCard, ?initialBalance, ?trip, ?reservationHolder, ?ticket}

precondition
  definedBy
    ?reservationRequest[
      reservationItem hasValue ?trip,
      reservationHolder hasValue ?reservationHolder ]
    memberOf tr##reservationRequest
  and ?trip memberOf tr##tripFromAustria
  and ?creditCard[ balance hasValue ?initialBalance ] memberOf po##creditCard.

assumption
  definedBy po##validCreditCard(?creditCard)
    and ( ?creditCard[ type hasValue "PlasticBuy" ] or
      ?creditCard[ type hasValue "GoldenCard" ] ).

postcondition
  definedBy
    ?reservation memberOf tr##reservation[ reservationItem hasValue ?ticket,
      reservationHolder hasValue ?reservationHolder ]
    and ?ticket[ trip hasValue ?trip ] memberOf tr##ticket.

effect
  definedBy
    ticketPrice(?ticket, "euro", ?ticketPrice)
    and ?finalBalance= (?initialBalance - ?ticketPrice)
    and ?creditCard[ po##balance hasValue ?finalBalance

```

Fig. 3.10. Example of service capability in WSML.

Figure 3.10 provides an example of a Web service capability specified in WSML. This example service offers information about trips starting in Austria and requires the name of the person and credit card details for making

the reservation. The assumption is that the credit card information provided by the requester must designate a valid credit card that should be of type either PlasticBuy or GoldenCard. The postcondition specifies that a reservation containing the details of a ticket for the desired trip and the reservation holder is the result of the successful execution of the Web service. Finally, the effect in the world state is that the credit card is charged with the cost of the ticket.

### 3.4.3 Service Interface

A WSML service interface contains the description of how the overall functionality of the Web service is achieved by means of cooperation of different Web service providers (orchestration) and the description of the communication pattern that allows to one to consume the functionality of the Web service (choreography). A choreography description has two parts: the state and the guarded transitions. As mentioned above, a state is represented by an WSMO ontology, while guarded transitions are if-then rules that specify conditional transitions between states in the abstract state space.

```

interface BookTicketInterface
  importsOntology _http://www.example.org/BookTicketInterfaceOntology
  choreography BookTicketChoreography
  orchestration BookTicketOrchestration

choreography BookTicketChoreography
  state _"http://example.org/BookTicketInterfaceOntology"
  guardedTransitions BookTicketChoreographyTransitionRule

guardedTransitions BookTicketChoreographyTransitionRule
if ( reservationRequestInstance [ reservationItem hasValue ?trip,
                                reservationHolder hasValue ?reservationHolder ]
    memberOf bti#reservationRequest
    and ?trip memberOf tr#tripFromAustria
    and ticketInstance[ trip hasValue ?trip, recordLocatorNumber hasValue ?rln ]
    memberOf tr#ticket )
then temporaryReservationInstance[ reservationItem hasValue ticketInstance,
                                   reservationHolder hasValue ?reservationHolder ]
    memberOf bti#temporaryReservation

```

**Fig. 3.11.** Example of WSML service interface.

Figure 3.11 provides an example of a service interface with choreography, and a guarded transition rule which requires the following to hold: If a reservation request instance exists (it has been already received, since the corresponding concept in the state ontology currently has the mode "in") with the request for a trip starting in Austria, and there exists a ticket instance for the desired

trip in the Web service instance store, then create a temporary reservation for that ticket.

#### 3.4.4 Software Support

The project web site [www.wsmo.org](http://www.wsmo.org) provides, for example, a comprehensive set of links to software tools for developing WSMO oriented services (in WSML) most of which available under open source related licenses at [sourceforge.net](http://sourceforge.net). Examples include the WSMO4J API<sup>13</sup>, the WSMO studio<sup>14</sup> with WSML service editor, WSML-DL and WSML-Rule reasoner, WSML validator, and the WSMX service execution environment<sup>15</sup>.

Remarkably, there are still no implemented semantic WSML service composition planner nor full-fledged WSML service matchmaker available apart from rather simple keyword-based and non-functional (QoS) parameter oriented WSML service discovery engine as part of the WSMX suite, and the hybrid matchmaker WSMO-MX.

#### 3.4.5 Limitations

The WSMO conceptual model and its language WSML is an important step forward in the Semantic Web service domain as it explicitly overcomes some but not all limits of OWL-S. Unfortunately, the development of WSMO and, in particular, WSML has been originally at the cost of its connection to the W3C Web service standard stack at that time. This raised serious concerns by the W3C summarized in its official response to the WSMO submission in 2005 from which we quote<sup>16</sup>: "The submission represents a development, but one which has been done in isolation of the W3C standards. It does not use the RDFS concepts of Class and Property for its ontology, and does not connect to the WSDL definitions of services, or other parts of the Web Services Architecture. These differences are not clearly explained or justified. The notion of choreography in WSMO is obviously very far from the definition and scope presented in WS-CDL. The document only gives little detail about mediators, which seem to be the essential contribution in the submission." To date, however, the connection of WSML with WSDL and SAWSDL (WSDL-S) has been established in part, and is under joint investigation by both WSMO and SAWSDL initiatives in relevant working and incubator groups of the standardisation bodies OASIS and W3C.

Another main critique of WSML concerns the lack of formal semantics of service capabilities in both the WSMO working draft as of 2006, and the WSML specification submitted to the Web consortium W3C in 2005. Recently,

---

<sup>13</sup> <http://wsmo4j.sourceforge.net/>

<sup>14</sup> <http://www.wsmostudio.org/download.html>

<sup>15</sup> <http://sourceforge.net/projects/wsmx/>

<sup>16</sup> <http://www.w3.org/Submission/2005/06/Comment>

this problem has been partly solved by means of a semi-monolithic FOL-based representation of functional service semantics over abstract state spaces and (guarded) state space transitions by service execution traces (Stollberg et al., 2007)[349]. Though, the formal semantics of the WSML service (orchestration and choreography) interface part is still missing - which is not worse than the missing process model semantics of OWL-S.

Further, principled guidelines for developing the proposed types of WSMO mediators for services and goals in concrete terms are missing. Besides, the software support for WSML services provided by the WSMO initiative appears reasonable with a fair number of downloads but is still not comparable to that of OWL-S in terms of both quantity and diversity.

Finally, as with OWL-S, it remains to be shown whether the revolutionary but rather academic WSMO framework will be adopted by major business stakeholders within their service application landscapes in practice. Apart from the announcement of SAWSDL as initial standard, this also relates to the key concern of insufficient scaling of logic-based reasoning to the Web scale in general, as mentioned in the previous chapter.

### 3.5 Monolithic DL-based Service Descriptions

As mentioned above, an alternative to formally specifying the functional semantics of a Web service agnostic to any structured service description formats like OWL-S, SAWSDL, or WSML, is the pure DL-based approach: The abstract service semantics is defined through an appropriate set of concept and role axioms in a given description logic. Any instantiation of this service concept corresponds to a concrete service with concrete service properties. That is, the extension  $S^I$  of a service concept  $S$  representing the abstract service to be described in an interpretation  $I$  of the concept over a given domain contains all service instances the provider of  $S$  is willing to accept for contracting with a potential requester of  $S$ . An example of a monolithic DL-based description of an abstract service and possible service instances is shown in figure 3.12 ([139]).

In this example, the functional semantics or capability of the abstract Web service  $S$  is described by a set  $D_S$  of two DL concept axioms: The service concept  $S$  for the shipping of items with a weight less than or equal to 50kg from cities in the UK to cities in Germany; the concept *Shipping* (used to define  $S$ ) which assures that instances of  $S$  specify exactly one location for origin and destination of the shipping. Semantic relations between such monolithically described service semantics can be determined fully within the underlying logical formalism, that is by DL-based inferencing. For a more detailed treatment of this topic, we refer to (Grimm, 2007) [139].

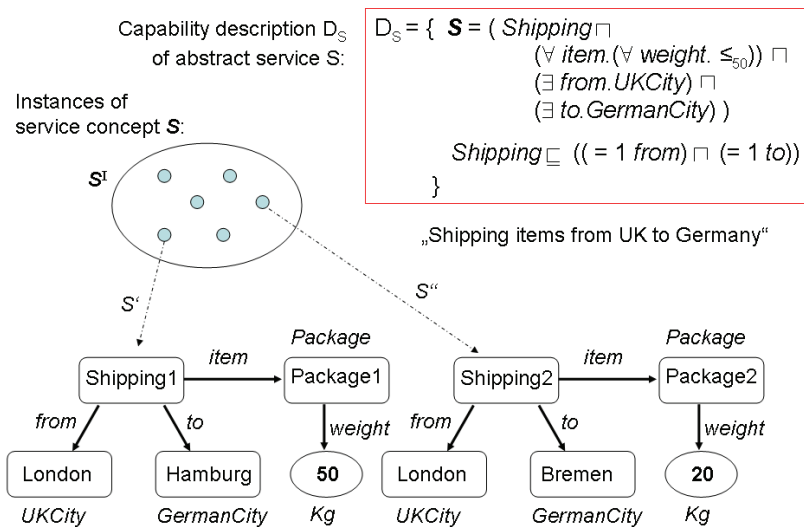


Fig. 3.12. Example of a monolithic DL-based semantic service description [139]

### 3.6 Semantic Web Service Coordination

Semantic service coordination aims at the coherent and efficient discovery, composition, negotiation, and execution of Semantic Web services in a given environment and application context. Each of these coordination activities is an active field of research in itself, and mainly treated separately in the literature. In general, the basic process of Web service coordination and the potential relations between service discovery, composition, negotiation and execution also hold for Semantic Web services (cf. chapter 2, section 1.3.4). However, what makes agent-based coordination of services in the Semantic Web different from its counterpart in the Web is its far more advanced degree of automation and, from the perspective of strong AI, meaning. This is achieved through means of logic-based reasoning on heterogeneous semantically annotated Web services by appropriate software agents with respective background theories (ontologies).

The following three parts of this work provide introductions to the fields of Semantic Web service discovery, composition planning, and negotiation. In these introductions, we also comment on the principled relationships between these coordination activities, and provide representative examples of their interleaved coordination, if available.

In summary, the feasibility of logic-based and hybrid semantic discovery and composition of Web services has been demonstrated (for different formats) in several major funded research projects, though its benefit in practice seems not yet bullet proved. In fact, current proof-of-concept implementations of Seman-



tic Web service matchmakers and composition planners are lacking sufficient scalability for the vast, open ended Web.

Apart from other open problems like privacy preserving coordination of Web services in the Semantic Web with its inherently higher potential of automated semantic inference attacks, research on the potential impact of exploiting Semantic Web technology for Web service negotiation, is in its very infancies yet. Though one would intuitively expect that a better understanding of service semantics should allow for a more informed and focussed decision making and negotiation, this has not been sufficiently investigated yet, neither in theory nor in practice. This concerns the process of automatically interleaving discovery, composition and negotiation as well as its expected revenues compared to Web service negotiation models. A brief account of the few approaches to agent-based semantic service negotiation and open problems are given in the introduction to part four of this work.

### 3.7 Semantic Web Service Applications

Despite the tremendous project support and progress made in the domain of Semantic Web services, the number of publicly deployed real world applications using this technology appears almost negligible compared to the large number of available Web services. We present selected applications of agent-based semantic services in part five of this work.

Prominent example of exploiting Semantic Web service technology on the large scale is the semantic Grid<sup>17</sup>[96] as an extension of the Grid [132] in which information and services are given well-defined meaning. Conversely, the Grid and its scientific users provide application pull which will benefit the Semantic Web and, in particular, the development of Semantic Web service applications. Besides, for the Semantic Web and service community to develop large scale and robust, distributed solutions, it might be helpful to also look to the Grids portfolio of software specifications, middleware components and practical deployed e-science services applications. For example, the myGrid/Taverna project has built a component (Feta)<sup>18</sup> for adding and retrieving semantic service and workflow descriptions in bioinformatics [244]. There are a few semantic grid projects with working prototypes that are relevant including the OntoGrid project ([www.ontogrid.net](http://www.ontogrid.net)).

Current competitions in the Semantic Web service domain include the Semantic Web Services Challenge<sup>19</sup> and the Semantic Service Selection (S3) Contest<sup>20</sup>. The first of which attempting to qualitatively measure the minimal amount of programming required to adapt the semantics of given systems to new services, acknowledging that the complete automation of composing previously

---

<sup>17</sup> [www.semanticgrid.org](http://www.semanticgrid.org)

<sup>18</sup> Source is available through [www.mygrid.org.uk/taverna](http://www.mygrid.org.uk/taverna)

<sup>19</sup> <http://sws-challenge.org>

<sup>20</sup> <http://www.dfki.de/-klusch/s3>

unknown services is impossible, rather being a kind of Holy Grail of modern semantic technologies. This activity is complemented by the S3 Contest through the comparative evaluation of the performance of semantic service matchmaking tools based on respective service retrieval test collections like the OWLS-TC<sup>21</sup>.

### 3.8 Critique

Main critiques of Semantic Web services range from limitations of proposed frameworks via the lack of appropriate means of service coordination and software support to the legitimation of the research field as a whole. As one consequence, SWS technology still appears too immature for getting adopted by both common Web users and developers in practice, and industry for its commercial use on a large scale.

*Do we really need formal service semantics?*

Some recent critics of SWS technology argue against the significance of its claimed benefits for practical Web service applications in general. Key justification of this argument, is related to the general critics on Semantic Web technologies. In fact, the need of having formal logic-based semantics specified for Web services in practical human-centred applications is often questioned: It is completely unclear whether the complete lack of formal service semantics turns out to be rather negligible, or crucial for what kinds of service applications for the common Web user in practice, and on which scale.

Just recently, van Harmelen and Fensel [113] argued for a more tolerant and scalable Semantic Web reasoning based on approximated rather than strict logic-based reasoning. This is in perfect line with experimental results available for hybrid SWS matchmakers that combine both logic and approximated reasoning like the OWLS-MX (Klusch et al., 2005)[201], the WSMO-MX (Kaufner & Klusch, 2006)[182] and the syntactic OWLS-iMatcher (Bernstein & Kiefer, 2006) [32].

*Where are all the Semantic Web services?*

Another interesting question concerns the current reality of Semantic Web service technology in use. According to a recent survey of publicly available Semantic Web service descriptions in the surface Web [213], revealed that not more than around 1500 indexed semantic services in OWL-S, WSMO, WSDL-S or SAWSDL are accessible in the Web of which only about one hundred are deployed outside special test collections like the OWLS-TC<sup>22</sup>. Though we expect the majority of Semantic Web services being maintained in

---

<sup>21</sup> Available at [projects.semwebcentral.org/projects/owls-tc/](http://projects.semwebcentral.org/projects/owls-tc/)

<sup>22</sup> [projects.semwebcentral.org/projects/owls-tc/](http://projects.semwebcentral.org/projects/owls-tc/)

private project repositories and sites of the deep Web [152], it certainly does not reflect the strong research efforts carried out in the Semantic Web service domain world wide.

Of course, one might argue that this comes at no surprise in two ways. First, Semantic Web service technology is immature (with a standard announced just recently, that is SAWSDL) which still provides insufficient common ground supporting its exploitation by end users. Though this is certainly true, the other related side of this argument is that massive research and development of the field around the globe should have produced a considerable amount of even publicly visible Semantic Web service descriptions within the past half dozen of years.

Second, one might argue that it is not clear whether the surface Web and academic publications are the right place to look for Semantic Web service descriptions, as many of them would be intended for internal or inter-enterprise use but not visible for the public. Though this is one possible reason of the low numbers reported above, it indicates some lack of visibility to the common Web user to date.

*Where are the easy to use Semantic Web service tools for the public?*

As with Semantic Web application building in general, apart from the project prototypes and systems there is hardly any easy to use software support off the shelves available to the common user for developing, reusing and sharing her own Semantic Web services - which might hamper the current confluence of the field with the Web 2.0 into the so-called service Web 3.0 in practice.

*How to efficiently coordinate Semantic Web services?*

Despite tremendous progress made in the field in European and national funded research projects like DIP, Super, CASCUM, Scallops and SmartWeb, there still is plenty of room for further investigating the characteristics, potential, and limits of Semantic Web service coordination in both theory and practice. The Semantic Web Services Challenge<sup>23</sup> attempts to qualitatively measure the minimal amount of programming required to adapt the semantics of given systems to new services. This acknowledges that the complete automation of composing previously unknown services is impossible, rather being a kind of Holy Grail of modern semantic technologies. Besides, the comparative evaluation of developed Semantic Web service discovery tools is currently hard, if not impossible, to perform since the required large scale service retrieval test collections are still missing even for the standard SAWSDL. Related to this, there are no large scale experimental results on the scalability of proposed service coordination means in practice available.

Apart from the problem of scalable and efficient Semantic Web service discovery and composition, another open problem of Semantic Web service coordination as a whole is privacy preservation. Though there are quite a few

---

<sup>23</sup> <http://semantic-web-service-challenge.org>

approaches to user data privacy preservation for each of the individual coordination processes (discovery, composition, and negotiation), there is no integrated approach that allows to coherently secure Semantic Web service coordination activities.

### *Summary*

The interdisciplinary, vivid research and development of the Semantic Web did accomplish an impressive record in both theory and applications within just a few years since its advent in 2000. Though we identified several major gaps to bridge before Semantic Web service technology reaches maturity, its current convergence with Web 2.0 towards a service Web 3.0 in an envisioned Internet of Things holds promise to effectively revolutionize computing applications for our everyday life. In the following parts, we briefly survey research and development of Semantic Web service discovery, composition planning, and negotiation, and present innovative contributions to each of these areas.

## 3.9 Further Readings

For more comprehensive information on Semantic Web services, we refer to the accessible readings [348, 57, 115] on the subject. Examples of major funded research projects on Semantic Web services are

- the European funded integrated projects DIP<sup>24</sup> and ASG (Adaptive semantic services grid technologies)<sup>25</sup>
- SmartWeb - Mobile multi-modal provision of Semantic Web services,
- SCALLOPS<sup>26</sup> - Secure Semantic Web service coordination,
- CASCOM<sup>27</sup>, ARTEMIS<sup>28</sup> - Semantic Web services for e-health applications (mobile, P2P)

For more information about Semantic Web service description frameworks, we refer to the respective documents submitted to the W3C:

- OWL-S ([www.w3.org/Submission/OWL-S/](http://www.w3.org/Submission/OWL-S/))
- WSMO ([www.w3.org/Submission/WSMO/](http://www.w3.org/Submission/WSMO/))
- SAWSDL ([www.w3.org/2002/ws/sawSDL/](http://www.w3.org/2002/ws/sawSDL/))
- Semantic Web Services Framework SWSF ([www.daml.org/services/swsf/](http://www.daml.org/services/swsf/)) with SWSL-Rule ([www.w3.org/Submission/SWSF-SWSL/](http://www.w3.org/Submission/SWSF-SWSL/)) for monolithic FOL-based service representation by means of different variants of rule languages (DLP, HiLog, etc).

---

<sup>24</sup> [dip.semanticweb.org/](http://dip.semanticweb.org/)

<sup>25</sup> [asg-platform.org](http://asg-platform.org)

<sup>26</sup> [www-ag.s.dfki.uni-sb.de/~klusch/scallops/](http://www-ag.s.dfki.uni-sb.de/~klusch/scallops/)

<sup>27</sup> [www.ist-cascom.org](http://www.ist-cascom.org)

<sup>28</sup> [www.srdc.metu.edu.tr/webpage/projects/artemis/](http://www.srdc.metu.edu.tr/webpage/projects/artemis/)

## Semantic Service Discovery



---

## Introduction

Service discovery is the process of locating existing Web services based on the description of their functional and non-functional semantics. Discovery scenarios typically occur when one is trying to reuse an existing piece of functionality (represented as a Web service) in building new or enhanced business processes. A Semantic Web service, or in short semantic service, is a Web service which functionality is described by use of logic-based semantic annotation over a well-defined ontology (cf. chapter 3). In the following, we focus on the discovery of semantic services. Both service-oriented computing and the Semantic Web envision intelligent agents to proactively pursue this task on behalf of their clients.<sup>29</sup>

Semantic service discovery can be performed in different ways depending on the considered service description language, means of service selection and coordination through assisted mediation or performed in a peer-to-peer fashion. In general, any semantic service discovery framework needs to have the following components ([139]).

- **Semantic service description:** A semantic service description language (more precisely top-level ontology, also called service description format) is used to represent the functional and non-functional semantics of Web services. Examples of structured and logic-based semantic service description language are OWL-S and WSML. The standard Semantic Web service description language SAWSDL allows for a structured representation of service semantics in XML(S) with references to any kind of non-logic-based or logic-based ontology for semantic annotation.<sup>30</sup> Alternatively, in so-called monolithic logic-based service descriptions the functionality of a service is represented by means of a single logical expression of an appropriate logic, usually a description logic like OWL-DL or WSML-DL (cf. chapter 3).

---

<sup>29</sup> This introduction is part of the book chapter (Klusch, 2008a)[196].

<sup>30</sup> In this sense, SAWSDL services can be seen as a weaker form of semantic services, and WSDL services are no semantic services.

- **Semantic service selection:** Service selection encompasses semantic matching and ranking of services to select a single most relevant service to be invoked, starting from a given set of available services. This set can be collected and maintained, for example, by front-end search engine, or given by providers advertising their services at registries or middle-agents like matchmakers and brokers. Semantic service matching, or in short: service matching, is the pairwise comparison of an advertised service with a desired service (query) to determine the degree of their semantic correspondence (semantic match). This process can be non-logic-based, logic-based or hybrid depending on the nature of reasoning means used. Non-logic-based matching can be performed by means of, for example, graph matching, data mining, linguistics, or content-based information retrieval to exploit semantics that are either commonly shared (in XML namespaces), or implicit in patterns or relative frequencies of terms in service descriptions. Logic-based semantic matching of services like those written in the prominent service description languages OWL-S (Ontology Web Language for Services), WSML (Web Service Modeling Language) and the standard SAWSDL (Semantically Annotated WSDL) exploit standard logic inferences. Hybrid matching refers to the combined use of both types of matching.
- **Discovery architecture:** The conceptual service discovery architecture concerns the environment in which the discovery is assumed to be performed. This includes assumptions about the (centralized or decentralized P2P) physical or semantic overlay of the network, the kind of service information storage (e.g., service distribution, registries, and ontologies) and location mechanisms, as well as the agent society in the network (e.g., service consumers, providers, middle-agents).

In the following, we survey existing approaches to semantic service matching and discovery architectures. Examples of semantic service description languages were presented in the previous chapter.

### Classification of Semantic Web Service Matchmakers

Semantic service matching determines whether the semantics of a desired service (or goal) conform to that of an advertised service. This is at the very core of any semantic service discovery framework. Current approaches to semantic service matching can be classified according to

- what kinds and parts of service semantics are considered for matching, and
- how matching is actually performed in terms of non-logic-based or logic-based reasoning on given service semantics, or a hybrid combination of both, within or partly outside the respective service description framework (cf. chapter 3).



Figure 3.13 shows representative examples of implemented Semantic Web service matchmakers for each of these categories.

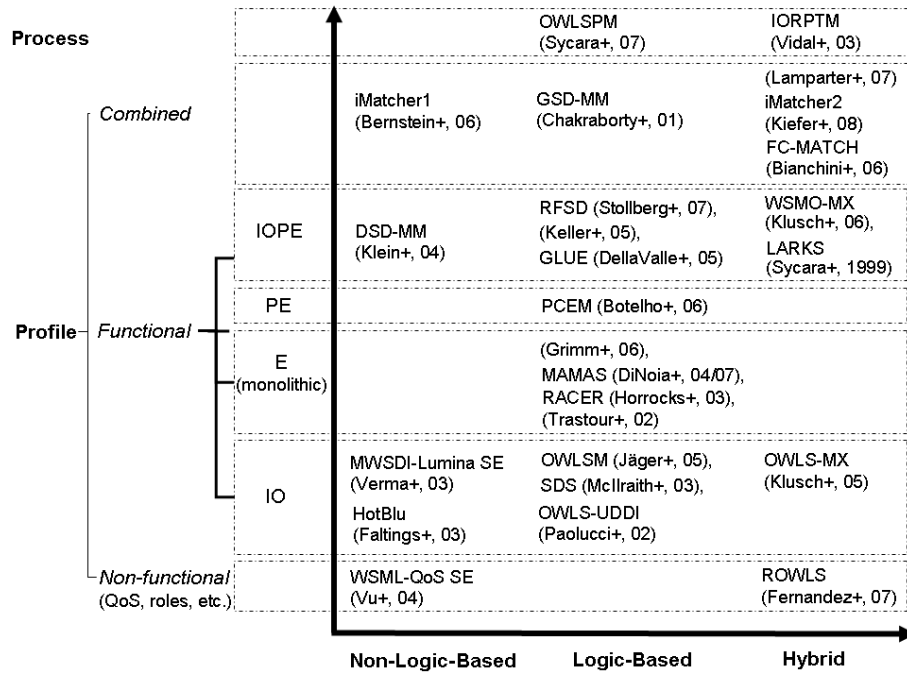


Fig. 3.13. Categories of Semantic Web service matchmakers.

*Non-logic-based, logic-based, and hybrid semantic service matching*

The majority of Semantic Web service matchmakers performs deductive, that is logic-based semantic service matching. In this sense, they are keeping with the original idea of the Semantic Web to determine semantic relations (thus resolve semantic heterogeneities) between resources including services based on logical inferencing on their semantic annotations that are formally grounded in description logics (DL) and/or rules (cf. chapter 2). As shown in figure 3.13, pure logic-based semantic matchmakers for services in OWL-S and WSML are currently prevalent. Non-logic-based semantic service matchmakers do not perform any logic-based reasoning to determine the degree of a semantic match between a given pair of service descriptions. Examples of non-logic-based semantic matching techniques are text similarity measurement, structured graph matching, and path-length-based similarity of concepts<sup>31</sup>.

<sup>31</sup> Please note that any kind of semantic service matching that identifies concepts or rules (which are logically defined in a given ontology) by their names only

Most Semantic Web service matchmakers perform service profile rather than service process model matching. Service profile matching (so-called "black-box" service matching) determines the semantic correspondence between services based on the description of their profiles. The profile of a service describes what it actually does in terms of its signature, that is its input and output (IO), as well as preconditions (P) and effects or postconditions (E), and non-functional aspects such as the relevant business category, name, quality, privacy and pricing rules of the service. We classify additional context information for service matching such as the organisational (social or domain) roles, or geographic location of service requesters and providers in their interaction as non-functional.

Service process-oriented matching (so-called "glass-box" service matching) determines the extent to which the desired operational behavior of a given service in terms of its process control and data flow matches with that of another service. Like with service profile matching, we can distinguish between non-logic based, logic based and hybrid semantic process matching approaches depending on whether automated reasoning on operational semantics specified in some certain logic or process algebraic language (e.g. CCS,  $\pi$ -calculus) is performed, or not. An overview of relevant approaches to process mining for process discovery is given in (van der Aalst & Weijters, 2004)[367].

*Supported Semantic Web service description formats*

Each of the implemented Semantic Web service matchmakers shown in figure 3.13 supports only one of the many existing Semantic Web service description formats (cf. chapter 3) as follows. This list is representative but not exhaustive.

- **OWL-S** matchmakers: Logic-based semantic matchmakers for OWL-S services are the OWLSM (Jäger et al., 2005)[177] and OWLS-UDDI (Paolucci et al., 2002)[286] focussing on service IO-matching, and the PCEM (Botelho et al., 2006)[45] that converts given OWL-S services to PDDL actions for PROLOG-based service PE-matching. Further OWL-S matchmakers are the hybrid service IO-matchmaker OWLS-MX (Klusch et al., 2006)[201], the hybrid non-functional profile matchmaker ROWLS (Fernandez et al., 2006)[116], the hybrid (combined) profile matchmaker FC-MATCH (Bianchini et al., 2006)[35], the non-logic-based (full) service matchmaker iMatcher1 (Bernstein & Kiefer, 2006)[32] and its hybrid successor iMatcher2 (Bernstein & Kiefer, 2008)[186]. An approach to logic-based OWL-S process model verification is in (Vaculin & Sycara, 2007)[366] while (Bae et al., 2006)[21] present an approach to the matching

---

does not classify as logic-based matching in the strict sense. Without any formal verification of the semantic relation between given (semantic service annotation) concepts based on their logical definitions, the matchmaker performs non-logic-based semantic service matching.

of OWL-S process dependency graphs based on syntactic similarity measurements, and Bansal and Vidal (2003)[23] propose a hybrid matchmaker that recursively compares the DAML-S process model dependency graphs.

- **WSML matchmakers:** Implemented approaches to WSML service discovery include the hybrid semantic matchmaker WSMO-MX (Klusch and Kaufer, 2006; cf. chapter 6), the logic-based matchmaker GLUE (Della Valle et al., 2005)[94], and the syntactic search engine (part of the WSMO studio) for QoS-enabled WSML service discovery in P2P networks (Vu et al., 2006)[377]. Approaches for logic-based semantic matching of so-called rich functional service descriptions (WSML-oriented) in abstract state spaces based on transaction logic are proposed in (Keller et al., 2005; Stollberg et al., 2006)[185, 349], though it is unclear to what extent they have been implemented.
- **WSDL-S/SAWSDL matchmakers:** The METEOR-S WSDI discovery infrastructure (Verma et al., 2004)[372] and the UDDI-based search component Lumina<sup>32</sup> are the only tool support of searching for SAWSDL services so far. While searching with Lumina is keyword based, the MWSDI discovery of SAWSDL services relies on non-logic-based matching means.
- **Monolithic DL-based matchmakers:** Only very few matchmakers are agnostic to the above mentioned structured Semantic Web service description formats without conversion by accepting monolithic descriptions of services in terms of a single service concept written in a given DL. In this case, semantic matching directly corresponds to DL inferencing, that is, semantic service matching is done exclusively within the logic theory. Examples of monolithic DL-based service matchmakers are RACER (Li & Horrocks, 2003)[232], MaMaS<sup>33</sup> (Di Noia et al., 2004; 2007)[99, 100], and the semantic service matching approaches proposed in (Grimm et al., 2006)[141] and (Trastour et al., 2002). Recently, (Lamparter & Ankolekar, 2007)[228] present an implemented approach to matching of monolithic service descriptions in OWL-DL extended with (non-functional) pricing policies (modeled as DL-safe SWRL rules) according to given preferences by means of SPARQL queries to a service repository.
- **Others:** Non-logic-based semantic service IOPE profile matchmakers for other structured service description formats are the DSD-MM matchmaker (Klein & König-Ries, 2004)[189] for DIANE services, the HotBlu matchmaker (Constantinescu & Faltings, 2003)[72] that performs numeric service IO-message data type matching, and the hybrid semantic service IOPE

---

<sup>32</sup> [ldis.cs.uga.edu/projects/meteor-s/downloads/Lumina/](http://ldis.cs.uga.edu/projects/meteor-s/downloads/Lumina/)

<sup>33</sup> [sisinflab.poliba.it/MAMAS-tng/](http://sisinflab.poliba.it/MAMAS-tng/)

matchmaker LARKS for services in an equally named service description format (Sycara et al., 1999; Sycara et al., 2002)[355].

In the following, we discuss each category of Semantic Web service matching together with selected representative examples of the above mentioned Semantic Web service matchmakers in more detail. This is complemented by a classification of existing service discovery architectures for which these matchmakers have been designed for, or can be used in principle. As stand-alone implementations, each matchmaker classifies as centralized service discovery system, though a few of them have been also tested for, or were originally developed for decentralized P2P service retrieval systems like the OWLS-MX and the OWLS-UDDI matchmaker, respectively, the WSMO-QoS search engine and the DReggie/GSD matchmaker.<sup>34</sup>

### Logic-Based Semantic Service Profile Matching

As mentioned above, logic-based semantic service matchmakers perform deductive reasoning on service semantics. The majority of such matchmakers pairwise compare logic-based descriptions of service profile semantics. In order to define these semantics, logical concepts and rules are taken from respective ontologies as first-order or rule-based background theories with a shared minimal vocabulary. Different ontologies of service providers and service requester are matched or aligned either at design time, or at runtime as part of the logic-based service matching process.

#### *Matching degrees*

The degree of logic-based matching of a given pair of semantic service profiles can be determined either (a) exclusively within the considered logic theory by means of logic reasoning, or (b) by a combination of logical inferences within the theory and algorithmic processing outside the theory. Prominent logic-based matching degrees are exact, plugin, subsumes, and disjoint which are defined differently depending on the parts of service semantics and the logic theory that is used to compute these degrees.

One prominent example for a software specification matching degree is the so-called plug-in match. A specification  $S$  plugs into (plug-in matches with) another specification  $R$ , if the effect of  $S$  is more specific than that of  $R$ , and vice versa for the preconditions of  $S$  and  $R$  (Zaremski & Wing, 1996)[391]. If this definition is restricted to effects only, the matching degree is called a post plug-in match. Unfortunately, the original notion of plug-in match

---

<sup>34</sup> For reasons of readability, the implemented (stand-alone) Semantic Web service matchmakers shown in figure 3.13 each representing a central discovery system by itself are not again listed in figure 3.14, and vice versa, that is, those matchmaking approaches being inherent part of the functionality of each node of decentralized discovery systems (but not available as stand-alone matchmaker) are not listed in figure 3.13.

has been adopted quite differently by most logic-based Semantic Web service matchmakers for both monolithic and structured service descriptions.

#### *Monolithic logic-based service matching*

Matching of monolithic logic-based semantic service descriptions (cf. chapter 3) is performed exclusively by means of logic inferences within the considered logic theory. That is, the functionality of a Web service is represented by a single (monolithic) expression in an appropriate logic, usually a description logic like OWL-DL or WSML-DL. As a consequence, monolithic logic-based semantic service matching reduces to standard first-order (description) logic reasoning such as checking the satisfiability of service and query concept conjunction, or the entailment of concept subsumption over a given knowledge base. Furthermore, it is agnostic to any form of structured stateless (I/O) or stateful (IOPE) representation of service semantics like in OWL-S and WSML. The prominent degrees of semantic matching used by the majority of monolithic logic-based semantic service matchmakers are logic equivalence, post-plug-in match, subsumes, and fail.

For example, the logical so-called post-plug-in match of an advertised service  $S$  with a service request  $R$  bases on the entailment of concept subsumption of  $S$  by  $R$  over a given knowledge base  $kb$  extended by the axioms of  $S$  and  $R$ :  $kb \cup S \cup R \models S \sqsubseteq R$ . That is, the matchmaker checks if in each first-order interpretation (possible world)  $I$  of  $kb$ , the set  $S^I$  of concrete provider services (service instances) is contained in the set  $R^I$  of service instances acceptable to the requester:  $S^I \subseteq R^I$ . This assures the requester that each provided service instance offers at least the requested functionality, maybe even more. In other words, service  $S$  is more specific than the request  $R$ , hence considered semantically relevant. In contrast, the so-called logical subsumes match assures the requester that her acceptable service instances are also acceptable to the provider:  $kb \cup S \cup R \models R \sqsubseteq S$ .

Some monolithic DL-based service matchmakers also check for a so-called intersection or potential match (Grimm, 2007)[139]. This matching degree indicates the principled compatibility of service  $S$  with request  $R$  with respect to the considered knowledge base  $kb$  by means of either concept intersection or non-disjointness. In the first case, the advertised service concept  $S$  potentially matches with the (desired service or) query concept  $R$  if their concept conjunction  $S \cap R$  is satisfiable with respect to  $kb$  in some possible world  $I$  such that  $S^I \cap R^I \neq \emptyset$  holds. In the second case, the monolithic logic-based semantic service matchmaker makes a stronger check by determining whether the intersection of both concept extensions is non-empty in each possible world.

In general, the complexity of matching monolithic DL-based service descriptions is equal to the combined DL complexity. For example, post-plug-in matching of service concepts in OWL-Full, that is SHOIQ<sup>+</sup> (including transitive non-primitive roles) has been shown to be undecidable (Baader et al., 2005)[20] but decidable for OWL-DL, WSML-DL and DL-safe SWRL.

One problem of monolithic DL-based service matching is the risk to return false positives due to incomplete knowledge specified in service descriptions  $S$ ,  $R$  or the domain ontology  $kb$  [141]. In other words, semantic matching of  $S$  with  $R$  with respect to  $kb$  based on monotonic DL reasoning under open-world assumption (OWA) can wrongly succeed due to the existence of possible but unwanted interpretations of concepts or roles used in  $S$  or  $R$  over  $kb$ . Such unwanted possible worlds of  $kb$  are intuitively ruled out by humans by default - which accounts for their usually non-monotonic reasoning under closed-world assumption<sup>35</sup>.

One solution to this problem is to explicitly capture such default (common-sense) knowledge by adding, for example, appropriate concept disjointness axioms or object assertions to the knowledge base  $kb$ . This excludes possible worlds which are "obviously" wrong (but allowed due to open-world semantics) but is considered impracticable as it requires the modeler to somewhat "overspecify" the  $kb$  with "obvious" information.

An alternative solution is to perform semantic matching of services with local closed-world reasoning [141]. Key idea is to exclude the unwanted possible worlds of knowledge base  $kb$  by means of an additional autoepistemic logic operator  $\mathbf{K}$ <sup>36</sup> that allows to restrict the interpretation of certain concepts  $C$  and roles  $r$  used in advertised and desired service descriptions  $S$  and  $R$  to named individuals (nominals) in the ABox of  $kb$  which are definitely known or not known to belong to them  $(C, r)$ <sup>37</sup>.

<sup>35</sup> The OWA states that the inability to deduce some fact from a knowledge base does not imply its contrary by default, that is, the fact may hold (not in all but) in some possible world (interpretations of  $kb$ ). For example, the intersection match of  $R = Flight \sqcap \forall from. UKCity$  with  $S = Flight \sqcap \forall from. USCity$  with respect to the knowledge base  $kb = \{UKCity \sqsubseteq EUCity, Flight \sqsubseteq \exists from. \top\}$  wrongly succeeds. The reason is that  $kb$  is underspecified in the sense that (due to the OWA) there can be possible worlds in which cities can be both in the UK and the US, which causes a false positive for the intersection match.

<sup>36</sup> The epistemic logic operator  $\mathbf{K}$  allows to refer to definitely known facts by intersecting all possible worlds:  $(\mathbf{K}C)^{I,E} = \bigcap_{I \in E} C^{I,E}$ . The epistemic concept  $\mathbf{K}C$  is interpreted as the intersection of extensions of concept  $C$  over all first-order interpretations of  $kb$ , that is the set of all individuals that are known to belong to  $C$  (in the epistemic model  $E(kb)$  of  $kb$ , that is the maximal non-empty set of all first-order interpretations of  $kb$ ).

<sup>37</sup> In the above example, the intersection match of the request  $R = Flight \sqcap \forall from. \mathbf{K}UKCity$  with service  $S = Flight \sqcap \forall from. \mathbf{K}USCity$  with respect to the matchmaker knowledge base  $kb = \{UKCity \sqsubseteq EUCity, Flight \sqsubseteq \exists from. \top, UKCity(London)\}$  correctly fails, hence avoids to return a false positive. The satisfiability of the epistemic concept  $S \sqcap R$  requires the existence of a named individual  $x$  in  $kb$  known to be both  $UKCity$  and  $USCity$  (that is  $kb$  entails  $UKCity(x) \sqcap USCity(x)$ , i.e.  $kb \models UKCity(x)$  and  $kb \models USCity(x)$ , for every possible world  $I$  in the epistemic model  $E(kb)$ ). While the named individual *London* in the ABox of  $kb$  is definitely known to belong to the concept  $UKCity$ , and also known to belong to  $EUCity$  due to the inclusion axiom in the TBox of

However, this local closure of concepts and roles in  $S, R$  for their interpretation in  $kb$  (i.e., locally closing off possible worlds of  $kb$  in  $S$  and  $R$  without any occurrence of  $\mathbf{K}$  in  $kb$ ) under the local closed world-assumption (LCWA)<sup>38</sup> by use of the  $\mathbf{K}$ -operator makes semantic matching dependent on the state of the world: It requires the existence of named individuals in the ABox of  $kb$  as representative (static) information on the locally closed concepts and roles<sup>39</sup>. Besides, using an autoepistemic extension of description logics like OWL-DL or WSM-L-DL for semantic service matching is still uncommon in practice, though (non-monotonic) reasoners such as for epistemic query answering in ALCK<sup>40</sup> can be easily integrated in a matchmaker.

Another application of non-monotonic reasoning to monolithic DL-based service matching is presented in (Colucci et al., 2005; DiNoia et al., 2007)[69, 100]. The respective matchmaker MaMaS provides non-standard explanation services, that are non-monotonic logical abduction and contraction, for partial (also called approximated, intersection, or potential) matches. For example, concept contraction computes an explanation concept  $G$  to explain why a request concept  $R$  is not compatible with service concept  $S$ , that is, why  $S \sqcap R$  is not satisfiable ( $S \sqcap R \sqsubseteq \perp$ ). For this purpose, it keeps the least specific concept expression  $K$  of concept  $R$  such that  $K$  is still compatible with  $S$ , i.e.  $\neg(K \sqcap S) \sqsubseteq \perp$ . The remaining set  $G$  of constraints of  $R$  represents the desired explanation of mismatch. Such kind of non-monotonic logical service matching is NP-hard already for the simple description logic ALN. However, research in this direction has just begun and is, in part, related to research on non-monotonic reasoning with Semantic Web rule languages (cf. chapter 2). Examples of implemented monolithic DL-based matchmakers for service concepts written in OWL(-DL) and DAML+OIL are MaMaS (DiNoia et al., 2004; 2007)[99, 100], respectively, RACER (Li & Horrocks, 2003). Remarkably, both

---

$kb$ , it is not definitely known to also belong to  $USCity$  ( $kb \not\vdash USCity(London)$ ).

There is also no other named individual in  $kb$  which is both known to be in  $UKCity$  and  $USCity$  such that  $S \sqcap R$  is not satisfied. An intersection match of  $R$  with different service  $S' = Flight \sqcap \forall from.KEUCity$  correctly succeeds.

<sup>38</sup> The LCWA assumes that all individuals of some concept, or all pairs of individuals of some role are explicitly known in the local knowledge base (selected local concept or role closure).

<sup>39</sup> In the above example, the intersection match  $S' \sqcap R$  would (wrongly) fail, hence causes a false negative, if the named individual  $London$  would not have been explicitly stated in  $kb$  to belong to  $UKCity$  as its representative by default: There would be no named individual definitely known to belong to both  $UKCity$  and  $EUCity$  in all possible worlds. Though  $UKCity(London)$  has to be added to the  $kb$  of the matchmaker to avoid false positives and negatives of intersection matches by non-monotonic epistemic query answering  $kb$  with  $\mathbf{K}$ , no (dynamic) information about concrete flights, i.e. individuals of concept  $Flight$ , has to be additionally specified in  $kb$ .

<sup>40</sup> <http://www.fzi.de/downloads/wim/KToy.zip>

matchmakers determine the degree of post-plug-in match inverse to its original definition in [391].

#### *Service specification or PE-matching*

The logic-based semantic matching of service specifications (so-called PE-matching) concerns the comparison of their preconditions (P) and effects (E) and originates from the software engineering domain. As mentioned above, the plug-in matching of two software components  $S, R$  requires that the logic-based definition of the effect, or postcondition of  $S$  logically implies that of  $R$ , while the precondition of  $S$  shall be more general than that of  $R$  (Zaremski & Wing, 1996)[391]. In other words, a logic-based semantic plug-in match of service specifications  $S, R$  requires (in every model of given knowledge base  $kb$ ) the effect of advertised service  $S$  to be more specific than requested, and its precondition to be more general than requested in  $R$ . Depending on the Semantic Web service description framework (cf. chapter 3), the logic language for defining service preconditions and effects ranges from, for example, decidable def-Horn (DLP), WSML-DL and OWL-DL to undecidable SWRL, KIF and F-Logic(LP).

For example, the logic-based service-PE matchmaker PCEM (Botelho et al., 2006)[45] exploits the Java-based light-weight Prolog engine tuProlog<sup>41</sup> for logic-based exact matching of service preconditions and effects written in Prolog. In particular, the PCEM matchmaker checks whether there is a possibly empty variable substitution such that, when applied to one or both of the logical propositions (PE), this results into two equal expressions, and applies domain specific inference rules (for computing subPartOf relations).

The hybrid semantic WSML service matchmaker WSMO-MX (Kaufer & Klusch, 2006)[182] is checking an approximated query containment over a given finite service instance base for service effects (postconditions, constraints) written in undecidable F-Logic(LP) using OntoBroker. The approach to semantic service IOPE matchmaking described in (Stollberg et al., 2007)[349] uses the VAMPIRE theorem prover for matching pairs of preconditions and effects written in FOL, while the hybrid service IOPE matchmaker LARKS (Sycara et al., 2002)[355] performs polynomial theta-subsumption checking of preconditions and postconditions in def-Horn. There are no non-logic-based or hybrid semantic service PE matchmaker available yet.

#### *Service signature and IOPE-matching*

Logic-based semantic matching of service signatures (input/output, IO), so called service profile IO-matching, is the stateless matching of declarative data semantics of service input and output parameters by logical reasoning within the theory and algorithmic processing outside the theory. For example, the logic-based plug-in matching of state-based service specifications (PE) can be

---

<sup>41</sup> <http://alice.unibo.it/xwiki/bin/view/Tuprolog/>



adopted to the plug-in matching of stateless service signatures (IO): Service  $S$  is expected to return more specific output data whose logically defined semantics is equivalent or subsumed by those of the desired output in request  $R$ , and requires more generic input data than requested in  $R$ .

More concrete, the signature of  $S$  plugs into the signature of request  $R$  iff  $\forall IN_S \exists IN_R: IN_S < IN_R \wedge \forall OUT_R \exists OUT_S: OUT_S \in LSC(OUT_R)$ , with  $LSC(C)$  the set of least specific concepts (direct children)  $C'$  of  $C$ , i.e.  $C'$  is a immediate sub-concept of  $C$  in the shared (matchmaker) ontology. The quantified constraint that  $S$  may require less input than specified in  $R$  guarantees at a minimum that  $S$  is, in principle, executable with the input provided by the user in  $R$ . This holds if and only if the logical service input concepts are appropriately mapped to the corresponding WSDL service input message data types in XMLS.

Examples of Semantic Web service matchmakers that perform logic-based semantic matching of service signatures only are the OWLSM (Jäger et al., 2005)[177] and the OWLS-UDDI (Paolucci et al., 2002)[286]. Though the latter determines a signature plug-in matching degree which is defined inverse to the original definition and restricted to the output. (Keller et al., 2005) and (Stollberg et al., 2007) propose approaches to logic-based semantic IOPE matching of Web services. In general, logic-based matching of stateless service descriptions with I/O concepts and conjunctive constraints on their relationship specified in SHOIN has been proven decidable though intractable (Hull et al., 2006)[170]. This indicates the respective decidability of IOPE matching for OWL-S (with OWL-DL) and WSML (with WSML-DL).

### Non-Logic-Based Semantic Service Profile Matching

As mentioned above, non-logic-based Semantic Web service matchmaker do not perform any logical inferencing on service semantics. Instead, they compute the degree of semantic matching of given pairs of service descriptions based on, for example, syntactic similarity measurement, structured graph matching, or numeric concept distance computations over given ontologies. There is a wide range of means of text similarity metrics from information retrieval, approximated pattern discovery, and data clustering from data mining, or ranked keyword, and structured XML search with XQuery, XIRQL or TeXQuery [147, 7]. In this sense, non-logic-based semantic service matching means exploit semantics that are implicit in, for example, patterns, subgraphs, or relative frequencies of terms used in the service descriptions, rather than declarative IOPE semantics explicitly specified in the considered logic.

One example is the matchmaker iMatcher1 (Bernstein & Kiefer, 2006)[32] which imprecisely queries a set of OWL-S service profiles that are stored as serialized RDF graphs in a RDF database with an extension of RDQL, called iRDQL, based on four (token and edit based) syntactic similarity metrics from information retrieval. The imprecise querying of RDF resources with similarity joins bases on TFIDF and the Levenshtein metric. The results are ranked

according to the numerical scores of these syntactic similarity measurements, and a user-defined threshold.

The DSD (DIANE service description format) service matchmaker (Klein & König-Ries, 2004)[189, 224] performs, in essence, graph matching over pairs of state-based service descriptions in the object-oriented service description language DSD (with variables and declarative object sets) without any logic-based semantics. The matching process determines what assignment of IOPE variables is necessary such that the state-based service offer is included in the set of service instances defined by the request, and returns a numeric (fuzzy) degree of DSD service matching.

### Hybrid Semantic Service Profile Matching

Syntactic matching techniques are first class candidates for the development of hybrid semantic service profile matching solutions that combine means of both crisp logic-based and non-logic-based semantic matching where each alone would fail. Indeed, first experimental evaluation of the performance of hybrid semantic service matchmakers OWLS-MX (Klusch et al., 2005) and iMatcher2 (Kiefer & Bernstein, 2008) show that logic-based semantic service selection can be significantly outperformed by the former under certain conditions.

LARKS (Sycara et al., 1999; 2002)[356, 355] has been the first hybrid semantic service IOPE matchmaker for services written in a frame-based language called LARKS. The matchmaker OWLS-MX (Klusch et al., 2005; 2006)[202, 201] bases in part on LARKS, and is the first hybrid semantic service signature (IO) matchmaker for OWL-S services. OWLS-MX complements deductive (DL) reasoning with approximated IR-based matching. For this purpose, each of its four hybrid variants OWLS-M1 to OWLS-M4 applies a selected token-based string similarity metric (cosine/TFIDF, extended Jaccard, Jensen-Shannon, LOI) to the given pair of service signature strings in order to determine their degree of text similarity-based matching. If the text similarity value exceeds a given threshold the failure of logic-based matching is tolerated, that means the service is eventually classified as semantically relevant to the given query. The ranking aggregates both types of matching degrees with respect to the total order of logic-based matching degrees. Experimental evaluation results over the test collection OWLS-TC together with a FP/FN-analysis of OWLS-MX showed that the performance of logic-based semantic matching can be improved by its combination with non-logic-based text similarity measurement (Klusch & Fries, 2007; Klusch et al., 2008)[199, 200].

Similarly, the hybrid semantic service profile matchmaker iMatcher2 (Kiefer & Bernstein, 2008)[186] uses multiple edit- or token-based text similarity metrics (Bi-Gram, Levenshtein, Monge-Elkan and Jaro similarity measures) to determine the degree of semantic matching between a given pair of OWL-S service profiles. Like OWLS-MX, the iMatcher transforms each structured service profile description into a weighted keyword vector that includes not

only the names but terms derived by means of logic-based unfolding of its service input and output concepts. In this sense, iMatcher2 classifies as a hybrid matchmaker. The experimental evaluation of iMatcher2 over the test collection OWLS-TC2.1 confirmed, in principle, the previously reported results of the evaluation of OWLS-MX.

In its adaptive mode, iMatcher2 can also be trained over a given retrieval training collection to predict the degree of semantic matching of unknown services to queries by means of selected regression models (support vector regression with a RBF kernel, linear and logistic regression). This regression-based induction is performed over the set of (a) the binary value of subjective semantic relevance as defined in the relevance sets, and (b) different text similarity values computed by means of the selected similarity metrics for each pair of query and service of the training collection. After training, the iMatcher2 first computes the text similarity values (using the selected similarity metrics) of a given query to all services of a given test collection, then uses the learned regression model to predict the combined similarity (or likelihood) of a match, and finally returns the answers in decreasing order of similarity. Experimental evaluation of the adaptive iMatcher2 showed that the combined logical deduction and regression-based learning of text similarities produces superior performance over logical inference only.

The hybrid semantic service matchmaker FC-MATCH (Bianchini et al., 2006)[35] does a combined logic-based and text similarity-based matching of single service and query concepts written in OWL-DL (SHOIN(D)). A service concept  $S$  is defined as logical conjunction of existential qualified role expressions where each role corresponds to a selected profile parameter:  $S = \exists \text{hasCategory}(C_1) \sqcap \exists \text{hasOperation}(C_2) \sqcap \exists \text{hasInput}(C_3) \sqcap \exists \text{hasOutput}(C_4)$ ). Hybrid matching degrees are computed by means of (a) combined checking of logic-based subsumption of profile concepts ( $C_i$ ) and (b) computing the so-called Dice (name affinity) similarity coefficient between terms occurring in these concepts according to the given terminological relationships of the thesaurus WordNet. FC-MATCH (FC stands for functional comparison) performs structured hybrid semantic matching of functional (I/O) and non-functional profile parameters (hasCategory, hasOperation). That is a combined matching of functional and non-functional parameters of OWL-S service profiles rewritten in special OWL-DL expressions. To the best of our knowledge, FC-MATCH has not been experimentally evaluated yet.

WSMO-MX (Kaufer & Klusch, 2006)[182] is the first hybrid semantic matchmaker for services written in a WSML-Rule variant, called WSML-MX. The hybrid service matching scheme of WSMO-MX is a combination of ideas of hybrid semantic matching as performed by OWLS-MX, the object-oriented graph matching of the matchmaker DSD-MM, and the concept of intentional matching of services proposed in (Keller et al., 2005). WSMO-MX applies different logic-based and text similarity matching filters to retrieve and rank services that are relevant to a query. The hybrid semantic matching degrees

are recursively computed by aggregated valuations of (a) ontology-based type matching (logical concept subsumption), (b) logical (instance-based) constraint matching in F-logic(LP) through approximative query containment, (c) relation name matching, and (d) syntactic similarity measurement as well. The experimental evaluation of WSMO-MX over an initial WSML service retrieval test collection is ongoing work.

However, it is not yet known what kind of hybrid service matching will scale best to the size of the Web in practice. Research in this direction is in perfect line with the just recent call in (van Harmelen and Fensel, 2007)[113] for a general shift in Semantic Web research towards scalable, approximative rather than strict logic-based reasoning.

### Logic-Based Semantic Service Process Matching

Automated semantic matching of service process models is uncommon, and was not intended by the designers of OWL-S and WSML. Besides, the semantics of process models in OWL-S or WSML have not been formally defined yet, while neither SAWSDL nor monolithic service descriptions offer any process model. This problem can be partly solved by intuitively rewriting the process model descriptions in an appropriate logic with automated proof system and respective analysis tool support.

For example, in (Vaculin & Sycara, 2007)[366], OWL-S service process models are mapped into (intuitively) equivalent logical Promela statements which are then efficiently evaluated by the SPIN model checker.<sup>42</sup> This allows to verify the correctness of a given service process model in terms of consistency and liveness properties of an advertised service like the Delivery process always executes after the Buy process. The result of such service process model checking could be used for process-oriented OWL-S service selection (by identifying properties of service process models to be verified with queries to match); this is a topic of ongoing research.

Alternatively, the matching of process models of OWL-S services that are grounded in WSDL (cf. chapter 3) can be, in principle, reduced to the matching of corresponding WSDL service orchestrations in BPEL. As mentioned before, the OWL-S process model captures a common subset of workflow features that can be intuitively mapped to BPEL (used to define WSDL service compositions) which offers an all-inclusive superset of such features (e.g. structured process activities in BPEL like Assignment, Fault Handler, Terminate are not available in OWL-S) [16]. Though BPEL has been given no formal semantics either yet, there are a few approaches to fill this gap based on Petri nets (Lohmann, 2007)[243] and abstract state machines (Fahland & Reisig, 2005)[111] that allow to at least verify liveness properties of WSDL service

---

<sup>42</sup> A model checker verifies if a given system (service process) model satisfies a desirable property. If the property does not hold, it returns a counter-example of an execution where the property fails.

orchestrations in BPEL [250]. However, there are no approaches to exploit any of the proposed formal BPEL semantics for semantic matching of OWL-S process models that correspond to BPEL orchestrations of WSDL services.

### **Non-Logic-Based and Hybrid Semantic Service Process Matching**

There are a few approaches to non-logic-based Semantic Web service process model matching. For example, (Bae et al., 2006)[21] present an approach to the matching of (business) process dependency graphs based on syntactic similarity measurements. Bansal and Vidal (2003)[23] propose a hybrid matchmaker (IO-RPTM) that recursively compares the DAML-S process model dependency graphs based on given workflow operations and logical match between IO parameter concepts of connected (sub-)service nodes of the process graphs. On the other hand, means of functional service process matching can be exploited to search for a set of relevant subservices of a single composite service.

### **Semantic Service Discovery Architectures**

Existing Semantic Web service discovery architectures and systems in the literature can be broadly categorized as centralized and decentralized by the way they handle service information storage and location in the considered service network (Aktas et al., 2006; Grimm, 2007)[5, 139]. A classification of implemented Semantic Web service discovery systems is given in figure 3.14). Centralized service discovery systems rely on one single, possibly replicated, global directory service (repository, registry) maintained by a distinguished so-called super-peer or middle agent like matchmaker, broker or mediator agent (Klusch & Sycara, 2001)[212]. Contrary, decentralized service discovery systems rely on distributing service storage information over several peers in a structured, unstructured or hybrid P2P network.

Semantic service discovery systems can be further classified with respect to the kind of semantic service matching means used by the intelligent agents in the network. For example, the exact keyword-based service location mechanisms of all contemporary P2P systems like JINI, SLP, Gnutella flooding, and DHT (distributed hash table) can be complemented or replaced by sophisticated logic-based semantic matching means to improve the quality of the search result.

As mentioned above, due to its generic functionality, any service matchmaker (cf. figure 3.13) can be used in arbitrary discovery architectures and systems. In the extremes, a matchmaker can either serve as a central service directory (index) or look-up service, or can be integrated into each peer of an unstructured P2P service network to support a semantic service search like in RS2D (Basters & Klusch, 2006)[39].

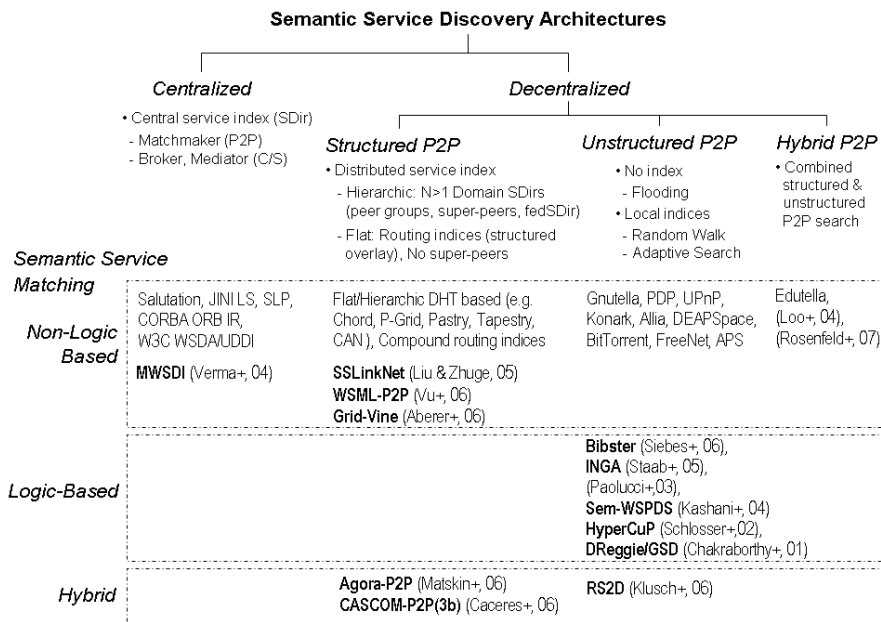


Fig. 3.14. Categories of Semantic Web service discovery architectures and systems.

### Centralized Semantic P2P Service Discovery

In centralized semantic P2P service networks, a dedicated central service directory (or matchmaker) returns a list of providers of semantically relevant services to the requester. Contrary to centralized client-server middleware or brokering, the requester then directly interacts with selected providers for service provision (Klusch & Sycara, 2001) [212]. The advantage of such centralized discovery architectures is a fast resource or service lookup time, though the central look-up server or registry like in JINI or the CORBA ORB interface registry is a single point of failure that can be only partially mitigated by replication and caching strategies.

An application of centralized P2P service discovery is the Napster music file sharing system, and the SETI@home system that is exploiting a vast set of distributed computational resources world wide to search for extraterrestrial signals. From the Semantic Web service discovery perspective, each of the above mentioned stand-alone Semantic Web service matchmakers, in principle, realizes a centralized logic-based semantic service discovery system by itself. For example, the SCALLOPS e-health service coordination system uses the hybrid semantic matchmaker OWLS-MX as a central matchmaker for the selection of relevant e-health services in a medical emergency assistance application. The same matchmaker is distributed to each peer of an unstructured

P2P network for decentralized OWL-S service discovery (Basters & Klusch, 2006)[39].

MWSDI (Verma et al., 2004)[372] is a centralized semantic P2P service system with non-logic-based semantic service signature matching. Each peer in the system maintains one domain specific WSDL-S (SAWSDL) service registry and respective ontologies; multiple peers can form a domain-oriented group. However, a distinguished central gateway or super-peer provides a global registries ontology (GRO) that maintains the complete taxonomy of all domain registries, the mappings between WSDL-S service I/O message types and concepts from shared domain ontologies in the system, associates registries to them, and serves as central look-up service for all peers. This central super-peer is replicated in form of so-called auxiliary peers for reasons of scalability. For service location, any client peer (user) selects the relevant domain registries via the central GRO at the super-peer which then performs non-logic-based semantic matching (structural XMLS graph matching, N-Gram-based syntactic similarity, synonyms/hyponyms/hypernyms in the GRO) of service input and output concepts with those of the desired service. However, it would be hard to build the GRO, and difficult for the user to query the GRO without knowing its details in advance.

### **Decentralized Semantic P2P Service Discovery**

Decentralized semantic service discovery relies on service information storage and location mechanisms that are distributed over all peers of structured, unstructured or hybrid P2P networks.

#### *Structured semantic P2P service systems*

Structured P2P networks have no central directory server but a significant amount of structure of the network topology (P2P overlay) which is tightly controlled. Resources are placed neither at random peers nor in one central directory but at specified locations for efficient querying. In other words, the service index of the system is distributed to all peers according to a given structured P2P overlay enforcing a deterministic content distribution which can be used for routing point queries.

Prominent examples of structured P2P systems are those with flat DHT-based resource distribution and location mechanism like Chord rings (Stoica+, 2001), Pastry (Rowstron+, 2001), Tapestry (Zhao+, 2001), CAN (Ratnasamy+, 2001), P-Grid (Aberer et al., 2006), and structured hierarchic P2P systems with super-peers. Flat DHT-based systems allow to route queries with certain keys to particular peers containing the desired data. But to provide this functionality all new content in the network has to be published at the peer responsible for the respective key, if new data on a peer arrives, or a new peer joins the network.

In structured hierarchical (also called N-super-peer) P2P systems, the peers are organized in ( $N > 1$ ) domain-oriented groups with possibly heterogeneous

service location mechanisms (e.g. hierarchic DHT, that is, one group with Chord ring overlay, another one with P-Grid overlay, etc.). Each group is represented by one super-peer hosting the group/domain service index. The set of super-peers, in turn, can be hierarchically structured with federated service directories in a super-peer (top-level) overlay of the network. Peers within a group query its super-peer which interacts with other super-peers to route the query to relevant peer groups for response. The functionality of a super-peer of one peer group is not necessarily fixed, but, in case of node failure, transferable to a new peer of that group. Typically JXTA, a collection of P2P protocols, is used to realize super-peer based P2P systems, though it does not enforce such architectures.

Examples of decentralized Semantic Web service discovery in structured P2P networks are WSPDS (Kashani et al., 2004)[181], SSLinkNet (Liu & Zhuge, 2005)[236], CASCOM-P2P<sub>3b</sub> (Caceres et al., 2006)[53], Grid-Vine (Aberer et al., 2006)[1], WSML-P2P (Vu et al., 2006)[377] and Agora-P2P (Küngas et al., 2006)[223, 237]. SSLinkNet, Agora-P2P and WSML-P2P exploit keyword-based discovery in a Chord ring, respectively, P-Grid system with non-logic-based semantic profile matching of services in WSDL, respectively, WSML. The Grid-Vine system performs non-logic-based semantic P2P content retrieval by means of so-called semantic gossiping with the underlying P-Grid system. The CASCOM and Agora-P2P systems have been demonstrated for logic-based semantic OWL-S (DAML-S) service discovery in hierarchic structured P2P networks.

In the SSLinkNet (Liu & Zhuge, 2005)[236], a Chord ring-based search is complemented by forwarding the same Web service request by the identified peers to relevant neighbors based on a given so-called semantic service link network. The semantic links between services are determined by non-logic-based semantic service matching, and are used to derive semantic relationships between service provider peers based on heuristic rules.

Similarly, the AGORA-P2P system (Küngas et al., 2006)[223, 237] uses a Chord ring as the underlying infrastructure for a distributed storage of information about OWL-S services over peers. Service input and output concept names are hashed as mere literals to unique integer keys such that peers holding the same key are offering services with equal literals in a circular key space. A service request is characterized as a syntactic multi-key query against this Chord ring. Both systems, SSLinkNet and AGORA-P2P, do not cope with the known problem of efficiently preserving the stability of Chord rings in dynamic environments.

The generic CASCOM semantic service coordination architecture has been instantiated in terms of a hierarchic structured P2P network with N interacting super-peers each hosting a domain service registry that make up a federated Web service directory. Each peer within a group can complement a keyword-based pre-selection of OWL-S services in their super-peer domain registries with a more complex semantic matching by a selected hybrid or logic-based semantic OWL-S matchmaker (ROWL-S, PCEM or OWLS-MX) on demand.



The respective semantic service search components are integrated into each peer (cf. chapter 16).

The Grid-Vine system (Aberer et al., 2006) [1] performs a hybrid semantic search of semantically annotated resources by means of so-called semantic gossiping between peers about their actual semantic knowledge (also called logical layer or semantic overlay of the P2P system). The semantic overlay is defined by (a) the set of peer ontologies in RDFS or XMLS that are used to encode document annotations in RDF (each RDF triple or concept in the peer schema represents a set of documents as its instances), and (b) a set of user-specified peer schema (concept) mappings that are used by the peers to translate received queries. The numeric "semantic quality" value of these directed concept mappings, hence the non-logic-based degree of semantic similarity between query and resource annotation concept of two peers is locally assessed by the requester through a quantitative analysis of (transitive) propagation cycles of the mappings (and their previous semantic quality value) which might be wrong but not by means of logic-based reasoning about concepts. The translation links, that is the mapping and its numeric "semantic quality" are continuously exchanged and updated by the peers: Semantic gossiping among peers is the propagation of queries to peers for which no direct but transitive translation links exist. The efficient location of resources for a given and translated query by the underlying P-Grid system bases on keyword-based matching of their identifiers, that are DHT keys.

Service discovery in structured P2P networks can provide search guarantees, in the sense of total service recall in the network, while simultaneously minimizing messaging overhead. However, this challenge has not been fully explored for unstructured P2P networks yet.

Service discovery in structured P2P networks can provide search guarantees, in the sense of total service recall in the network, while simultaneously minimizing messaging overhead. Typically, structured networks such as DHT-based P2P networks of  $n$  peers offer efficient  $O(\log(n))$  search complexity for locating even rare items, but they incur significantly higher organizational overheads (maintaining DHT, publishing)<sup>43</sup> than unstructured P2P networks. Alternatively, flooding-based or random-walks discovery in unstructured P2P networks are effective for locating highly replicated, means popular, but not rare items. Hybrid designs of P2P networks aim to combine the best of both worlds such as using random-walks (with state-keeping to prevent walkers from revisiting same peers) for locating popular items, and structured (DHT) search techniques for locating rare items [241].

---

<sup>43</sup> For example, peer  $p_n$  publishes each of its hashed items ( $term_i$ ) over the DHT network, that is the item gets stored in an inverted list ( $term_i, [..., p_n, ...]$ ) of some peer that is found in  $O(\log(n))$  hops.

In unstructured P2P systems, peers initially have no index nor any precise control over the network topology (overlay) or file placement based on any knowledge of the topology. That is, they do not rely on any structured network overlay for query routing as they have no inherent restrictions on the type of service discovery they can perform.

For example, resources in unstructured P2P systems like Gnutella or Morpheus are located by means of network flooding: Each peer broadcasts a given query in BFS manner to all neighbour peers within a certain radius (TTL) until a service is found, or the given query TTL is zero. Such network flooding is extremely resilient to network dynamics (peers entering and leaving the system), but generates high network traffic.

This problem can be mitigated by a Random Walk search where each peer builds a local index about available services of its direct neighbour peers over time and randomly forwards a query to one of them in DFS manner until the service is found<sup>44</sup> as well as replication and caching strategies based on, for example, access frequencies and popularity of services (Lu et al., 2002)[245]. Approaches to informed probabilistic adaptive P2P search like in APS (Tsoumakos & Roussopoulos, 2005)[364] improve on such random walks based on estimations over dynamically observed service location information stored in the local indices of peers. In contrast to the structured P2P search, this only provides probabilistic search guarantees, that is incomplete recall.

In any case, the majority of unstructured P2P service systems only performs keyword-based service matching and does not exploit any qualitative results from logic-based or hybrid semantic service matching to improve the quality of an informed search. In fact, only a few system are available for logic-based or hybrid semantic Web service retrieval such as DReggie/GSD (Chakraborty et al., 2002; Chen et al., 2001)[62, 64], HyperCuP (Schlosser et al., 2003)[328], Sem-WSPDS (Kashabi et al., 2004)[181], (Paolucci et al., 2003)[287], Bibster (Haase et al., 2006)[148], INGA (Löser et al., 2005)[242], and RS2D (Basters & Klusch, 2006)[39]. These systems differ in the way of how peers perform flooding or adaptive query routing based on evolving local knowledge about the semantic overlay, that is knowledge about the semantic relationships between distributed services and ontologies in unstructured P2P networks. Besides, all existing system implementations, except INGA and Bibster, perform semantic service IO profile matching for OWL-S (DAML-S), while HyperCuP peers dynamically build a semantic overlay based on monolithic service concepts. For example, Paolucci et al. (2003)[287] propose the discovery of relevant DAML-S services in unstructured P2P networks based on both the Gnutella P2P discovery process and a complementary logic-based service matching process (OWLS-UDDI matchmaker) over the returned answer set. However,

---

<sup>44</sup> This is valid in case the length of the random walk is equal to the number of peers flooded with bounded TTL or hops).

the broadcast or flooding-based search in unstructured P2P networks like Gnutella is known to suffer from traffic and load balancing problems. Though Bibster and INGA have not been explicitly designed for Semantic Web service discovery, they could be used for this purpose. In INGA (Löser et al., 2005)[242], peers dynamically adapt the network topology, driven by the dynamically observed history of successful or semantically similar queries, and a dynamic shortcut selection strategy, which forwards queries to a community of peers that are likely to best answer given queries. The observed results are used by each peer for maintaining a bounded local (recommender) index storing semantically labelled topic specific routing shortcuts (that connect peers sharing similar interests).

Similarly, in Bibster (Haase et al., 2006)[148] peers have prior knowledge about a fixed semantic overlay network that is initially built by means of a special first round advertisement and local caching policy. Each peer only stores those advertisements that are semantically close to at least one of their own services, and then selects for given queries only those two neighbours with top ranked expertise according to the semantic overlay it knows in prior. Further, prior knowledge about other peers ontologies as well as their mapping to local ontologies is assumed. This is similar to the ontology-based service query routing in HyperCuP (Schlosser et al., 2003)[328].

In RS2D (Basters & Klusch, 2006)[39], contrary to Bibster and DReggie/GSD, the peers perform an adaptive probabilistic risk-driven search for relevant OWL-S services without any fixed or prior knowledge about the semantic overlay. We describe this system in more detail in chapter 7.

#### *Semantic service discovery systems for hybrid P2P networks*

Hybrid P2P search infrastructures combine both structured and unstructured location mechanisms. For example, Edutella combines a super-peer network with routing indices and an efficient broadcast. In (Loo et al., 2004)[241] a flat DHT approach is used to locate rare items, and flooding techniques are used for searching highly replicated items. A similar approach of hybrid P2P query routing that adaptively switches between different kinds of structured and unstructured search together with preliminary experimental results are reported in (Rosenfeld et al., 2007)[318]. However, there are no hybrid P2P systems for semantic service discovery available yet.

Despite recent advances in the converging technologies of Semantic Web and P2P computing (Staab & Stuckenschmidt, 2006)[345], the scalability of semantic service discovery in structured, unstructured or hybrid P2P networks such as those for real-time mobile ad-hoc network applications is one major open problem. Research in this direction has just started. Preliminary solutions vary in the expressivity of semantic service description, and the complexity of semantic matching means ranging from computationally heavy Semantic Web service matchmakers like OWLS-MX in SCALLOPS and CASCOM, to those with a streamlined DL reasoner such as Krhype (Kleemann

& Sinner, 2007)[188] suitable for thin clients on mobile devices in IASON (Furbach et al., 2007)[124]. An example analysis of semantic service discovery architectures for realizing a mobile e-health application is given in [59].

## The Contributions

In the following four chapters, we present innovative means of hybrid semantic service matchmaking, and adaptive semantic service retrieval in unstructured P2P networks. These contributions are joint work with colleagues at the Carnegie Mellon University in Pittsburgh (USA), and my Master students Benedikt Fries, Ulrich Basters, and Frank Kaufer at DFKI.

*Chapter 4: Hybrid service matching with LARKS.* Early work on agent-based semantic service selection started in the late 1990s and include the first hybrid semantic matchmaker LARKS. Services (also called agent capabilities) are described in a frame-based language LARKS while the matching process uses five different filters to perform both logic-based semantic profile IOPE matching, and syntactic matching of the service description as a whole. Different degrees of partial matching are determined by different combinations of these filters. The LARKS matchmaker considerably influenced the initial work on the Semantic Web mark-up language DAML-S, the predecessor of OWL-S, and the subsequent development of a variety of Semantic Web service matchmakers. LARKS has been implemented in Java and successfully demonstrated for selected use cases in the military domain.

*Chapter 5: Hybrid semantic matching of OWL-S services.* Based on LARKS, we developed the first hybrid Semantic Web service signature matchmaker, called OWLS-MX. It complements logic-based semantic matching of OWL-S services with token-based syntactic similarity measurements. The results of the experimental evaluation of OWLS-MX are evidencing that logic-based semantic matching of OWL-S services can be in part significantly outperformed by both content-based and hybrid semantic service matching. Our experimental results were confirmed by Bernstein and Kiefer (2006, 2008)[32, 186]. Further experimental evaluation of the performance of OWLS-MX over an extended test collection OWLS-TC 2.2 together with a general analysis of its logic-based, hybrid and syntactic only false positives and false negatives is provided in (Klusch, Kapahnke & Fries, 2008)[200]. The matchmaker OWLS-MX was successfully used for semantic service selection in selected scenarios of emergency medical assistance in the e-health domain (cf. part 5).

*Chapter 6: Hybrid semantic matching of WSML services.* Further, only a few semantic matchmaker for WSML services are available. In this chapter, we present the first hybrid matchmaker, called WSMO-MX, for services in a LP extension of WSML-Rule, called WSML-MX. The hybrid service matching scheme of WSMO-MX, in essence, is a combination of the ideas of hybrid se-

semantic matching as realized by OWLS-MX, the object-oriented graph matching proposed by Klein and König-Ries (2004), and the concept of intentional matching of services presented by Keller et al. (2005)[185]. The matchmaker WSMO-MX performs hybrid semantic service IOP matching by combining logic-based reasoning with text similarity measurement. Different matching degrees are recursively computed by the aggregated valuation of logic-based type matching and constraint matching in F-Logic(LP), keyword-based relation matching, and text similarity measurement.

The results of our preliminary experimental evaluation of WSMO-MX over an initial test collection WSML-TC1 are reported in (Kaufer & Klusch, 2008) [207, 183]. Again, the experiments evidenced the potential of hybrid over logic-based only semantic matching in terms of increased service retrieval performance.

*Chapter 7: Semantic service discovery in pure P2P networks.* How to perform a semantic search for services in unstructured P2P networks in an efficient and robust way? In this chapter, we move beyond the state-of-the-art solutions for informed adaptive probabilistic and object-specific search in unstructured P2P networks. Key idea of our approach, called RS2D (Risk-Driven Semantic Service Discovery), is to let each peer agent (a) dynamically build and maintain its local view of the semantic overlay of the network, (b) use the OWLS-MX matchmaker for hybrid semantic service matching, and (c) learn the average query-answering behaviour of its direct neighbours in the network. The decision to whom to forward a semantic service request is then driven by its estimated probabilistic (Bayes' conditional) risk of routing failure in terms of both the implied semantic loss and communication costs. Notably, this kind of informed adaptive search not only renders the RS2D system independent from any fixed global ontology like in the DReggie/GSD system, or special advertisement rounds in advance like in the Bibster system. The RS2D performed comparatively well with respect to both retrieval and robustness.

An alternative version, called RS2D 2.0 (Basters & Klusch, 2006)[26], that uses a different semantic loss function showed a lower precision of service retrieval but with less communication effort than the original version presented in this chapter.

## Open Problems

Some major open problems of semantic service discovery are the following.

- *Approximated matching.* How to deal with uncertain, vague or incomplete descriptions of service semantics and user preferences for service selection? Fuzzy, probability, and possibility theory are first class candidates for the design of approximated (hybrid) semantic service matching (and ranking) algorithms to solve this problem. In particular, efficient reasoners for respective extensions of Semantic Web (rule) languages like probabilistic

pOWL, fuzzyOWL, or pDatalog can be applied to reason upon semantic service annotations under uncertainty and with preferences.

However, there are no such semantic service matchmakers available yet. Apart from the first hybrid matchmakers for OWL-S and WSML services, OWLS-MX and WSMO-MX, the same holds for the integrated use of means of statistical analysis from data mining or information retrieval for approximative matching of semantic service descriptions.

- *Scalability.* How to reasonably trade off the leveraging of expensive logic-based service selection means with practical requirements of resource-bounded, just-in-time and light-weight service discovery in mobile ad-hoc, unstructured or hybrid P2P service networks? What kind of approximated and/or adaptive semantic service discovery techniques scale best for what environment (network, user context, services distribution, etc) and application at hand? Required very large scale, comparative service retrieval performance experiments under real world conditions have not been conducted yet.
- *Adaptive discovery.* How to leverage semantic service discovery by means of machine learning and human-agent interaction? Though a variety of adaptive personal recommender and user interface agents have been developed in the field, none of the currently implemented semantic Web service matchmakers is capable of flexibly adapting to its changing user, network, and application environment.
- *Privacy.* How to protect the privacy of individual user profile data that are explicit or implicit in service requests submitted to a central matchmaker, or relevant service providers? Approaches to privacy preserving Semantic Web service discovery are still very rare, and research in this direction appears somewhat stagnant. Amongst the most powerful solutions proposed are the Rei language for annotating OWL-S services with privacy and authorization policies [95, 179], and the information flow analysis-based checking of the privacy preservation of sequential OWL-S service plans [174, 175]. However, nothing is known about the scalability of these solutions in practice yet.
- *Lack of tool support and test collections.* Support of Semantic Web service discovery by means of easy to use software tools is still lagging behind the theoretical advancements, though there are differences to what extent this is valid for what semantic service description framework (cf. figure 3.13). In particular, there is no official test collection for evaluating the retrieval performance of service discovery approaches (matchmakers, search engines) for the standard SAWSDL and WSML, while there are two publicly available for OWL-S (OWLS-TC2, SWS-TC). There are no solutions for the integrated matching of different services that are specified in dif-

ferent languages like SAWSDL, OWL-S and WSML. Relevant work on refactoring OWL-S and WSML to the standard SAWSDL is ongoing.





---

## Hybrid Service Matching with LARKS

K. Sycara, M. Klusch, S. Widoff, J. Lu: LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace. *Autonomous Agents and Multi-Agent Systems*, 5(2), pages 173 - 204, Kluwer Academic, 2002.



# LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace\*

KATIA SYCARA AND SETH WIDOFF

katia@cs.cmu.edu

*The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA*

MATTHIAS KLUSCH

klusch@dfki.de

*Deduction and Multiagent Systems Lab, DFKI GmbH, Saarbrücken, Germany*

JIANGUO LU

jglu@cs.toronto.edu

*Computer Science Department, University of Toronto, Canada*

**Abstract.** Service matchmaking among heterogeneous software agents in the Internet is usually done dynamically and must be efficient. There is an obvious trade-off between the quality and efficiency of matchmaking on the Internet. We define a language called LARKS for agent advertisements and requests, and present a flexible and efficient matchmaking process that uses LARKS. The LARKS matchmaking process performs both syntactic and semantic matching, and in addition allows the specification of concepts (local ontologies) via ITL, a concept language. The matching process uses five different filters: context matching, profile comparison, similarity matching, signature matching and constraint matching. Different degrees of partial matching can result from utilizing different combinations of these filters. We briefly report on our implementation of LARKS and the matchmaking process in Java. Fielded applications of matchmaking using LARKS in several application domains for systems of information agents are ongoing efforts.

**Keywords:** interoperability, multi-agent systems, matchmaking, capability description

## 1. Introduction

The amount of services and deployed software agents in the most famous offspring of the Internet, the World Wide Web, is exponentially increasing. In addition, the Internet is an open environment, where information sources, communication links and agents themselves may appear and disappear unpredictably. Thus, an effective, automated search and selection of relevant services or agents is essential for human users and agents as well.

We distinguish three general agent categories in the Cyberspace, *service providers*, *service requester*, and *middle agents*. Service providers provide some type of service, such as finding information, or performing some particular domain specific problem solving. Requester agents need provider agents to perform some service for them. Agents that help locate others are called middle agents [6]. *Matchmaking* is the

\*This research has been sponsored in part by Office of Naval Research grant N-00014-96-16-1-1222, and by DARPA grant F-30602-98-2-0138.

process of finding an appropriate provider for a requester through a middle agent, and has the following general form: (1) Provider agents advertise their capabilities to middle agents, (2) middle agents store these advertisements, (3) a requester asks some middle agent whether it knows of providers with desired capabilities, and (4) the middle agent matches the request against the stored advertisements and returns the result, a subset of the stored advertisements.

While this process at first glance seems very simple, it is complicated by the fact that not only local information sources but even providers and requesters in the Cyberspace are usually heterogeneous and incapable of understanding each other. This gives rise to the need for a common language for describing the capabilities and requests of software agents in a convenient way. Besides, one has to devise an efficient mechanism to determine a structural and semantic match of descriptions in that language. This means in particular using methods for reconciling potentially semantic heterogeneous informations [23]. There is an obvious trade-off between the quality and efficiency of matchmaking on the Internet.

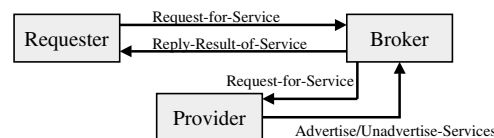
In the following, we briefly present the agent capability description language, LARKS, and then discuss the matchmaking process using LARKS. The paper concludes with a brief comparison with related works. We have implemented LARKS and the associated powerful matchmaking process, and are currently incorporating it within our RETSINA multi-agent infrastructure framework [44].

## 2. Matchmaking among heterogeneous agents

In the process of matchmaking (see Figure 1) three different kinds of collaborating agents involved are:

1. *Provider* agents provide their capabilities, e.g., information search services, retail electronic commerce for special products, etc., to their users and other agents.

### • Information Brokering



### • Information Matchmaking

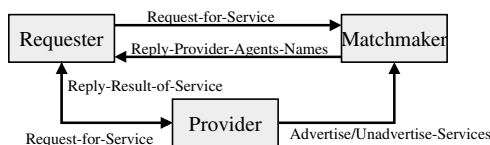


Figure 1. Service brokering vs. matchmaking.

2. *Requester* agents consume informations and services offered by provider agents in the system. Requests for any provider agent capabilities have to be sent to a matchmaker agent.
3. *Matchmaker* agents mediate among both, requesters and providers, for some mutually beneficial cooperation. Each provider must first register himself with a matchmaker. Provider agents advertise their capabilities (advertisements) by sending some appropriate messages describing the kind of service they offer. Every request a matchmaker receives will be matched with his actual set of advertisements. If the match is successful the matchmaker returns a ranked set of appropriate provider agents and the relevant advertisements to the requester.

In contrast to a broker agent, a matchmaker does not deal with the task of contacting the relevant providers, transmitting the service request to the service provider and communicating the results to the requester. This avoids data transmission bottlenecks, but it might increase the amount of interactions among agents.

### 2.1. *Agent capability description language requirements*

There is an obvious need to describe agent capabilities in a common language before any advertisement, request or even matchmaking among the agents can take place. In fact, the formal description of capabilities is one of the major problems in the area of software engineering and AI. Some of the main desired features of such a agent capability description language are the following.

- *Expressiveness*: The language is expressive enough to represent not only data and knowledge, but also to describe the meaning of program code. Agent capabilities are described at an abstract rather than implementation level.
- *Inferences*: Inferences on descriptions written in this language are supported. A user can read any statement in the language, and software agents are able to process, especially to compare any pair of statements automatically.
- *Ease of Use*: Every description should not only be easy to read and understand, but also easy to write by the user. The language should support the use of domain or common ontologies for specifying agents capabilities.
- *Application in the Web*: One of the main application domains for the language is the specification of advertisements and requests of agents in the Web. The language allows for automated exchange and processing of information among these agents.

In addition, the matchmaking process on a given set of capability descriptions and a request, both written in the chosen ACDL, should be efficient, accurate—not only relying on keyword extraction and comparison—and fully automated.

## 3. **The agent capability description language LARKS**

Representing capabilities is a difficult problem that has been one of the major concerns in the areas of software engineering, AI, and more recently, in the area of

Internet computing. There are many program description languages, like the Vienna Development Method (VDM), VDM++ [29] or Z [35], to describe the program functionality. These languages concern too detail-rich to be feasibly searched. Also, reading and writing specifications in these languages require sophisticated training. On the other hand, the interface definition languages, like WIDL [47], go to the other extreme by omitting the functional descriptions of the services entirely. Only the input and output signature information are provided.

In AI, knowledge description languages, like KL-ONE [3], or knowledge interchange formats such as KIF [22] are meant to describe the knowledge instead of the actions of a service. The action representation formalisms like STRIPS are too restrictive to represent complicated service. Some agent communication languages like KQML [10] and FIPA ACL [11, 12] concentrate on specifying communication performatives (message types) between agents but leave the content part of the language unspecified.

In Internet computing, various description formats are being proposed, notably the Web Interface Definition Language (WIDL) [47] and the Resource Description Framework (RDF) [36]. Although the RDF also aims at the interoperability between Web applications, it is intended rather to be a basis for describing metadata. RDF allows different vendors to describe the properties and relations between resources on the Web. That enables other programs, like Searchbots, to automatically extract relevant information, and to build a graph structure of the resources available on the Web, without the need to give any specific information. However, the description does not describe the functionalities of the *services* available in the Web.

Since no existing language satisfies our requirements, we propose an ACDL, called LARKS (Language for Advertisement and Request for Knowledge Sharing) that enables advertising, requesting and matching agent capabilities.

### 3.1. Specification in LARKS

A specification in LARKS is a frame with the following slot structure.

Context	Context of specification
Types	Declaration of used variable types
Input	Declaration of input variables
Output	Declaration of output variables
InConstraints	Constraints on input variables
OutConstraints	Constraints on output variables
ConcDescriptions	Ontological descriptions of used words
TextDescription	Textual description of specification

The frame slot types have the following meaning.

- **Context:** The context of the specification in the local domain of the agent.
- **Types:** Optional definition of the data types used in the specification.
- **Input and Output:** Input/output variable declarations for the specification. In addition to the usual type declarations, there may also be concept attachments

to disambiguate types of the same name. The concepts themselves are defined in the concept description slot `ConcDescriptions`.

- `InConstraints` and `OutConstraints`: Logical constraints on input/output variables that appear in the input/output declaration part. The constraints are described as Horn clauses.<sup>1</sup>
- `ConcDescriptions`: Optional description of the meaning of words used in the specification. The description relies on concepts defined in a given local domain ontology. Attachment of a concept `C` to a word `w` in any of the slots above is done in the form: `w*C`. That means that the concept `C` is the ontological description of the word `w`. The concept `C` is included in the slot `ConcDescriptions`.
- `TextDescription`: Optional text description of the meaning of the specification as a request for or advertisement of agent capabilities. In addition, the meaning of input and output declaration, type and context part of the specification may be described by attaching textual comments.

In our current implementation we assume each local domain ontology to be written in the concept language ITL (Information Terminological Language) [43]. Following section gives examples for how to attach concepts defined in this language in a LARKS specification, and also shows an example domain ontology in ITL. A generic interface for using ontologies in LARKS expressed in languages other than ITL will be implemented in near future.

Every specification in LARKS can be interpreted as an advertisement as well as a request; this depends on the purpose for which an agent sends a specification to some matchmaker agent(s). Every LARKS specification must be wrapped up in an appropriate KQML message by the sending agent indicating if the message content is to be treated as a request or an advertisement.

### 3.2. Examples of specifications in LARKS

The following two examples show how to describe in LARKS the capability to sort a given list of items, and return the sorted list. Example 3.1 is the specification of the capability to sort a list of at most 100 integer numbers, whereas in Example 3.2 a more generic kind of sorting real numbers or strings is specified in LARKS. Since the `ConcDescriptions` slot is empty, i.e., there is no concept attachment in the specification, the semantics of used words in it are assumed to be known to the matchmaker. Examples of how to use concept attachments in a specification are given in the next section.

#### Example 3.1 (Sorting integer numbers)

---

<code>IntegerSort</code>	
<hr/>	
<code>Context</code>	<code>Sort</code>
<code>Types</code>	
<code>Input</code>	<code>xs: ListOf Integer;</code>
<code>Output</code>	<code>ys: ListOf Integer;</code>

InConstraints	le(length(xs),100);
OutConstraints	before(x,y,ys) < - ge(x,y); in(x,ys) < - in(x,xs);
ConcDescriptions	
TextDescription	sort list of at most 100 integer numbers

---

### Example 3.2 (Generic sort of real numbers or strings)

<i>GenericSort</i>	
Context	<i>Sorting</i>
Types	
Input	xs: ListOf Real   String;
Output	ys: ListOf Real   String;
InConstraints	
OutConstraints	before(x,y,ys) < -ge(x,y); before(x,y,ys) < -preceeds(x,y); in(x,ys) < -in(x,xs);
ConcDescriptions	
TextDescription	<i>sorting list of real numbers or strings</i>

---

The next example is a specification of an agent's capability to buy stocks from particular companies, e.g., IBM, Apple or HP, at a stock market.

### Example 3.3 (Selling stocks by a portfolio agent)

sellStock	
Context	Stock, StockMarket;
Types	StockSymbols = {IBM, Apple, HP, SIEMENS, Daimler-Chrysler}, Money = Real;
Input	symbol: StockSymbols; yourMoney: Money; shares: Money;
Output	yourStock: StockSymbols; yourShares: Money; yourChange: Money;
InConstraints	yourMoney >= shares*currentPrice(symb);
OutConstraints	yourChange = yourMoney - shares*currentPrice(symb); yourShares = shares; yourStock = symb;
ConcDescriptions	
TextDescription	buying stocks from IBM, Apple, HP, SIEMENS, or Daimler-Chrysler at the stock market.

---

Given the name of the stock, the amount of money available for buying stocks and the shares for one stock, the agent is able to order stocks at the stock market. The constraints on the order are that the amount for buying stocks given by the user covers the shares times the current price for one stock. After performing the

order the agent will inform the user about the stock, the shares, and the gained benefit.

### 3.3. Using domain knowledge in LARKS

As mentioned before, LARKS offers the option to use application domain knowledge in any advertisement or request. This is done by using a local ontology for describing the meaning of a word in a LARKS specification. An example for such a domain ontology is given in the next section.

Local ontologies can be formally defined using, for example, concept languages such as ITL, BACK, LOOM, CLASSIC or KRIS, a full-fledged first order predicate logic, such as the knowledge interchange format (KIF) [22], or even the unified modeling language (UML) [13].

The main benefit of using domain knowledge in LARKS specifications is twofold:

1. the user can specify in more detail what she/he is requesting or advertising, and
2. the matchmaker agent is able to make automated inferences on such kind of additional, formally defined semantic descriptions while matching LARKS specifications, thereby improving the overall quality of matching.

As mentioned before, our current implementation of LARKS assumes the domain ontology to be written in the concept language ITL [43]. The research area on concept languages (or description logics) in AI has its origins in the theoretical deficiencies of semantic networks in the late 70's. KL-ONE [3] was the first concept language providing a well-founded semantics for a more natural language-based description of knowledge. Since then different concept languages have been intensively investigated; they are almost all decidable fragments of first-order predicate logic. The following is a simple example for a request and an advertisement written in LARKS in the air combat mission domain.

**Example 3.4 (A request and advertisement of agent capabilities).** We applied the matchmaking process using LARKS in the application domain of air combat missions. As an example for specification consider the following request and advertisement, 'ReqAirMissions' and 'AWAC-AirMissions,' respectively. The request is to find an agent which is capable to give information on deployed air combat missions launched in a given time interval. Some provider agent in this domain advertises his capability to provide information about a special kind of (AWAC) air combat missions.

---

#### *ReqAirMissions*

---

Context	Attack, Mission*AirMission
Types	Date = (mm: Int, dd: Int, yy: Int), DeployedMission = ListOf(mType: String, mID:String  Int)
Input	sd: Date, ed: Date



Output	missions: Mission
InConstraints	sd <= ed.
OutConstraints	deployed(mID), launchedAfter(mID,sd), launchedBefore(mID,ed).
ConcDescriptions	AirMission = (and Mission (atleast 1 has-airplane) (all has-airplane Airplane) (all has-MissionType aset(AWAC,CAP,DCA,HVAA)))
TextDescription	capable of providing information on deployed air combat missions launched in a given time interval
<hr/>	
<i>AWAC-AirMissions</i>	
<hr/>	
Context	Combat, Mission*AWAC-AirMission
Types	Date = (mm: Int, dd: Int, yy: Int) DeployedMission = ListOf(mt: String, mid:String  Int, mStart: Date, mEnd: Date)
Input	start: Date, end: Date
Output	missions: DeployedMission;
InConstraints	start <= end.
OutConstraints	deployed(mID), mt = AWAC, launchedAfter(mid,mStart), launchedBefore(mID,mEnd).
ConcDescriptions	AWAC-AirMission = (and AirMission (atleast 1 has-airplane) (atmost 1 has-airplane) (all has-airplane aset(E-2)))
TextDescription	capable of providing information on deployed AWAC air combat missions launched in some given time interval

Suppose that a provider agent such as, for example, HotBot, Excite, or even a meta-searchbot, like SavvySearch or MetaCrawler, advertises the capability to find informations about any type of computers. The administrator of the agent may specify that capability in LARKS as follows.

### Example 3.5 (Finding informations on computers)

---

#### FindComputerInfo

---

Context	Computer*Computer;
Types	InfoList = ListOf (model: Model*ComputerModel, brand: Brand*Brand, price: Price*Money, color: Color*Colors);

Input	brands: SetOf Brand*Brand; areas: SetOf State; processor: SetOf CPU*CPU; priceLow*LowPrice: Integer; priceHigh*HighPrice: Integer;
Output	Info: InfoList;
InConstraints	
OutConstraints	sorted(Info).
ConcDescriptions	Computer = (and Product (exists has-processor CPU) (all has-memory Memory) (all is-model ComputerModel)); LowPrice = (and Price (ge 1800) (exists in-currency aset(USD))); HighPrice = (and Price (le 50000) (exists in-currency aset(USD))); ComputerModel = aset(HP-Vectra,PowerPC-G3,Thinkpad770,Satellite315); CPU = aset(Pentium,K6,PentiumII,G3,Merced) [Product, Colors, Brand, Money]

---

Please note that provider and requester agents do not have to share the meaning of any words used in LARKS specifications. For example, suppose that the agents do not share the meaning of the word ‘Computer’ listed as a keyword in the Context slot of both, an advertisement and request, respectively. Without any concept attachment the matchmaker agent matches both specifications to be in the same context though they may refer to different domains of discourse.

Any knowledge on relations among concepts attached to a pair of words to be compared when matching two specifications helps the matchmaker agent to determine the semantic similarity between these words. All attached concepts in a given specification are formally defined in a local domain ontology<sup>2</sup> of provider or requester agent.

When multiple domain ontologies exist the matchmaker agent has to cope with the known ontological mismatch problem. If the agents share a common domain ontology equal or different names of concepts possess the same or different semantics, respectively. However, the more difficult case occurs when the agents do not share the same domain ontology; this may occur, for example, when agent capabilities were specified in the same application domain by different people. In this case, equality of concept names does not necessarily mean the equality of their semantics but has to be determined by the matchmaker agent using the concept definitions.<sup>3</sup> For this purpose the matchmaker agent dynamically builds and maintains a partially global terminology based on the received concept definitions. It is assumed that the vocabulary of basic words used in the definition of concepts of this terminology is dynamically shared by the providers and requesters. This provides a minimal common basis for a well-founded canonical interpretation of any concept in the ontology of the matchmaker.

**3.3.1. Example for a domain ontology in the concept language ITL.** Conceptual knowledge about a given application domain, or even common-sense, may be defined by a set of concepts and roles as terms in a given concept language. In the current implementation of LARKS we use the concept language ITL for this purpose. Each term as a definition of some concept *C* is a conjunction of logical constraints which

are necessary for any object to be an instance of  $C$ . The set of terminological definitions forms a particular style of an ontology, the *terminology*. Any definition of concepts in a terminology relies on

- a set of concepts and roles already defined in the terminology and/or
- a given basic vocabulary of words (primitive components) which are not defined in the terminology, that is, their semantics are assumed to be known and consistently used across boundaries.

The following terminology, is written in the concept language ITL and defines concepts in the computer application domain. It may be used in Example 3.5 in the former section.

```

Product      = (and (all is-manufactured-by Brand) (atleast 1 is-manufactured-by)
                (all has-price Price))
Computer     = (and Product (exists has-processor CPU) (all has-memory Memory)
                (all is-model ComputerModel))
Notebook     = (and Computer (all has-price
                (and (and (ge 1000) (le 2999)) (all in-currency aset(USD)))
                (all has-weight (and kg (le 5)) (all is-manufactured-by Company))
                (all is-model aset(Thinkpad380, Thinkpad770,Satellite315))))
Brand        = (and Company (all is-located-in State))
State        = (and (all part-of Country) aset(VA,PA,TX,OH,NY))
Company      = aset(IBM,Toshiba,HP,Apple,DEC,Dell,Gateway)
Colors       = aset(Blue,Green,Yellow,Red)
Money        = (and Real (all in-currency aset(USD,DM,FF,Y,P)))
Price        = Money
LowPrice     = (and Price (le 1800) (exists in-currency aset(USD)))
HighPrice    = (and Price (ge 5000) (exists in-currency aset(USD)))
ComputerModel = aset(HP-Vectra,PowerPC-G3,Thinkpad-380, Thinkpad-770,Satellite-315)
CPU          = aset(Pentium,K6,PentiumII,G3,Merced)

```

Obviously, at some point the providers and requesters must share a certain basic vocabulary to enable a meaningful comparison of used concepts. It is assumed that the basic set of primitive words of the partially global terminology of the match-maker is unique and shared with providers and requesters. The name of the used local terminology or domain ontology is denoted in the KQML message which wraps the LARKS specification.

**3.3.2. Subsumption relationships among concepts.** One of the main inferences on ontologies written in concept languages is the computation of the *subsumption* relation among two concepts: A concept  $C$  subsumes another concept  $C'$  if the extension of  $C'$  is a subset of that of  $C$ . This means, that the logical constraints defined in the term of the concept  $C'$  logically imply those of the more general concept  $C$ .

Any concept language is decidable if it is decidable for concept subsumption between two concepts defined in that language. The concept language ITL, which we use, is NP-complete decidable. We compromise expressiveness of the NP-complete

decidable ITL for (polynomial) tractability in our subsumption algorithm, which is correct but incomplete. For the mechanism of subsumption computation we refer the reader to, for example, [24, 32, 41, 42].

The computation of subsumption relationships between all concepts in a ontology yields a so-called concept hierarchy. Both the subsumption computation and the concept hierarchy are used in the matchmaking process (see Section 4.1.2).

We assume that the subsumption relation between two concepts may be identified with a real world semantic relation. Like in [39], we utilize an injective, domain-independent mapping between primitive components that occur in the concept definitions on the basis of given synonym relations.<sup>4</sup>

The matchmaker computes the subsumption relations between the concepts included in any advertisement he receives from registered provider agents. This yields a (set of) subsumption hierarchies of available concepts from a variety of local domain ontologies. An extension of the partial global ontology of the matchmaker with additional types of relations is presented in Section 4.1.4. Please note, that this ontology is not necessarily the union of all local domain ontologies of providers, and is *dynamically* built by the matchmaker while processing advertisements from registered provider agents. Any user or agent, requester or provider, may browse through the matchmaker's ontology and use the included concepts for describing the meaning of words in a specification of a request or advertisement in LARKS.<sup>5</sup>

#### 4. The matchmaking process using LARKS

As mentioned before, we differentiate between three different kinds of collaborating information agents: provider, requester and matchmaker agents. Figure 2 shows an overview of the matchmaking process using LARKS.

The matchmaker agent processes a received request in the following main steps:

- Compare the request with all advertisements in the advertisement database.
- Determine the provider agents whose capabilities match best with the request. Every pair of request and advertisement has to go through several different filters during the matchmaking process.
- Inform the requesting agent by sending them the contact addresses and related capability descriptions of the relevant provider agents.

For being able to perform a steady, just-in-time matchmaking process the information model of the matchmaker agent is comprised of the following components.

1. *Advertisement database (ADB)*. This database contains all advertisements written in LARKS the matchmaker receives from provider agents.
2. *Partial global ontology*. The ontology of the matchmaker consists of all ontological descriptions of words in advertisements stored in the ADB. Such a description is included in the slot `ConcDescriptions` and sent to the matchmaker with any advertisement.

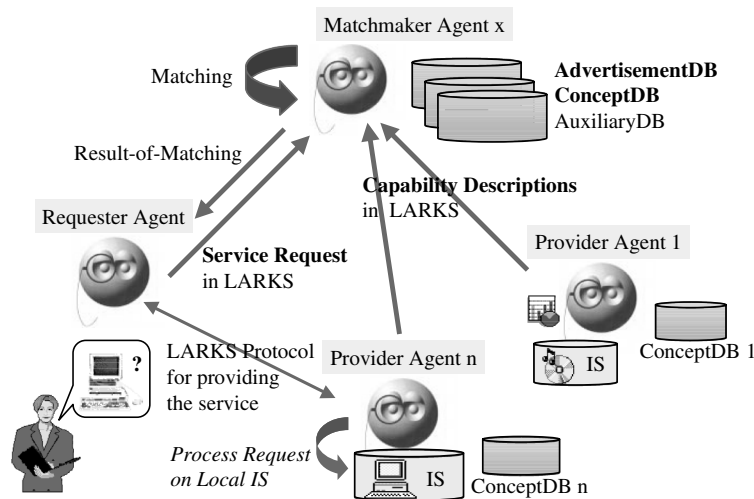


Figure 2. Matchmaking using LARKS: An overview.

3. *Auxiliary database.* The auxiliary data for the matchmaker comprises a database for word pairs and word distances, basic type hierarchy, and internal data.

As mentioned before, the ontology of a matchmaker agent is not necessarily equal to the union of local domain ontologies of all provider agents who are actually registered at the matchmaker. This also holds for the advertisement database. Thus, a matchmaker agent has only partial global knowledge on available information in the overall multi-agent system; this partial knowledge might also be not up-to-date concerning the actual time of processing incoming requests. This is due to the fact that for efficiency reasons changes in the local ontology of a provider agent will not be propagated immediately to all matchmaker agents he is registered at. In the following we will describe the matchmaking process using LARKS in more detail.

#### 4.1. *Filtering stages of the matchmaking process*

Agent capability matching is the process of determining whether an advertisement registered in the matchmaker matches a request. But when can we say two descriptions *match* against each other? Does it mean that they have the same text? Or the occurrence of words in one description sufficiently overlap with those of another description? When both descriptions are totally different in text, is it still possible for them to match? Even if they match in a given sense, what can we then say about the matched advertisements? Before we go into the details of the matchmaking process, we should clarify the various types of matches of two specifications.

#### 4.1.1. *Types of matching in LARKS*

*4.1.1.1. Exact match.* Of course, the most accurate match is when both descriptions are equivalent, either equal literally, or equal by renaming the variables, or equal logically obtained by logical inference. This type of matching is the most restrictive one.

*4.1.1.2. Plug-in match.* A less accurate but more useful match is the so-called *plug-in* match. Roughly speaking, plug-in matching means that the agent whose capability description matches a given request can be “plugged into the place” where that request was raised. Any pair of request and advertisement can differ in the signatures of their input/output declarations, the number of constraints, and the constraints themselves. As we can see, exact match is a special case of plug-in match, that is, wherever two descriptions are exact match, they are also plug-in match.

A simple example of a plug-in match is that of the match between a request to sort a list of integers and an advertisement of an agent that can sort both list of integers and list of strings. This example is elaborated in Section 5. Another example of plug-in match is between the request to find some computer information without any constraint on the output and the advertisement of an agent that can provide these informations and sorts the respective output.

*4.1.1.3. Relaxed match.* The least accurate but most useful match is the so-called *relaxed* match. A relaxed match has a much weaker semantic interpretation than a exact match and plug-in match. In fact, relaxed match will not tell whether two descriptions semantically match or not. Instead it determines how close the two descriptions are by returning just a numerical distance value. Two descriptions match if the distance value is smaller than a preset threshold value. Normally the plug-in match and the exact match will be a special case of the relaxed match if the threshold value is not too small.

An example of a relaxed match is that of the request to find the place (or address) where to buy a Compaq Pentium233 computer and the capability description of an agent that may provide the price and contact phone number for that computer dealer.

Different users in different situation may want to have different types of matches. Although people usually may prefer to have plug-in matches, such a kind of match does not exist in many cases. Thus, people may try to see the result of a relaxed match first. If there is a sufficient number of relaxed matches returned a refined search may be performed to locate plug-in matching advertisements. Even when people are interested in a plug-in match for their requests only, the computational costs for this type of matching might outweigh its benefits.

*4.1.2. Different filters of matching in LARKS.* For the matchmaking process we adopt several different methods from the area of information retrieval, AI and software engineering for computing syntactical and semantic similarity among agent capability descriptions. These methods are particularly efficient in terms of

performance as needed for dynamic matchmaking in the Internet. To summarize, the matching process is designed with respect to the following criteria:

- The matching should *not be based on keyword retrieval only*. Instead, unlike the usual free text search engines, the semantics of requests and advertisements should be taken into consideration.
- The matching process should be *automated*. A vast amount of agents appear and disappear in the Internet. It is nearly impossible for a user to manually search or browse all agents capabilities.
- The matching process should be *accurate*. For example, if the matches returned by the match engine are claimed to be exact match or plug-in match, those matches should satisfy the definitions of exact matching and plug-in matching.
- The matching process should be *efficient*, that is, it should be fast.
- The matching process should be *effective*, that is, the set of matches should not be too large. For the user, typing in a request and receiving hundreds of matches is not necessarily very useful. Instead, we prefer a small set of highly rated matches to a given request.

To fulfill the matching criteria listed above, the matching process is organized as a series of five increasingly stringent filters on candidate agents:

1. Context matching
2. Profile comparison
3. Similarity matching
4. Signature matching
5. Constraint matching.

All filters are independent from each other; each of them narrows the set of matching candidates with respect to a given filter criterion. The computational costs of these filters are in increasing order. Users may select any combination of these filters on demand. For example, when efficiency is the major concern, a user might select only the context and profile filters (similar to most conventional SearchBots in the Internet).

Context matching selects those advertisements in the ADB which can be compared with the request in the same or similar context. This filter roughly prunes off advertisements which are not relevant for a given request. The comparison of profiles, similarity and signature matching compare the request with any advertisement selected by the context matching. The request and advertisement profile comparison uses a weighted keyword representation for the specifications and a given term frequency based similarity measure [38]. The last filter, constraint matching, focus on the (input/output) constraints and declaration parts of the specifications. It checks if the input/output constraints of any pair of request and advertisement logically match (see Section 4.1.5).

Concerning the different types of matching there is the following relation to the different filters used in our matchmaker. The first three filters are meant for relaxed matching, and the signature and constraint matching filter are meant for plug-in matching.

**4.1.3. Different matching modes of the matchmaker.** Based on the given types and filters of matching we did implement four different modes of matching for the matchmaker:

1. *Complete Matching Mode.* In this mode all filters are considered for matching requests and advertisements in LARKS.
2. *Relaxed Matching Mode.* Only the context, profile and similarity filter are considered.
3. *Profile Matching Mode.* Only the context matching and comparison of profiles is done.
4. *Plug-In Matching Mode.* In this mode, the matchmaker performs only the signature and constraint matching.

If the considered advertisement and request contain conceptual attachments, i.e., ontological descriptions of used words, then in most of the filters, except for the comparison of profiles, we need a way to determine the semantic distance between the defined concepts. For that we use the computation of subsumption relationships and a weighted associative network.

**4.1.4. Computation of semantic distances among concepts.** We have presented the notion of concept subsumption in Section 3.3.2. But the concept subsumption gives only a generalization/specialization relation based on the definition of the concepts via roles and attribute sets. In particular for matchmaking the identification of additional relations among concepts is very useful because it leads to a deeper semantic understanding. Moreover, since the expressivity of the concept language ITL is restrictive so that performance can be enhanced, we need some way to express additional associations among concepts.

For this purpose we use a so-called weighted associative network, that is a semantic network with directed edges between concepts as nodes. Any edge denotes the kind of a binary relation among two concepts, and is labeled in addition with a numerical weight (interpreted as a fuzzy number). The weight indicates the strength of belief in that relation, since its real world semantics may vary.<sup>6</sup> We assume that the semantic network consists of three kinds of binary, weighted relationships: (1) generalization, (2) specialization (as inverse of generalization), and (3) positive association among concepts [7]. The *positive association* is the most general relationship among concepts in the network indicating them as synonyms in some context. Such a semantic network is called an *associative network* (AN).

In our implementation an AN is created by the matchmaker by using the computed concept subsumption hierarchy and additional associations extracted from the WordNet ontology [9]. We assume that the terminological subsumption relation among two concepts in the partial global ontology of the matchmaker may be identified with a real world semantical relation among them. That means, all subsumption relations are used for setting the generalization and specialization relations among concepts in the corresponding AN. Positive association, generalization and specialization relations are transitive.



Table 1. Kind of paths in an AN

	g	s	p
g	g	p	p
s	p	s	p
p	p	p	p

As mentioned above, every edge in the AN is labeled with a fuzzy weight. These weights are set by the user or automatically by default. The distance between two concepts in an AN is then computed as the strength of the shortest path among them. For performance reasons the matchmaker does not deal with dynamically resolving ambiguities due to potential genericity and polysemy in the AN (see, e.g., [8]). Combining the strength of each relation in a path is done by using the following triangular norms for fuzzy set intersections [27]:

$$\begin{aligned} \tau_1(\alpha, \beta) &= \max\{0, \alpha + \beta - 1\} & n = -1 \\ \tau_2(\alpha, \beta) &= \alpha \cdot \beta & n = 0 \\ \tau_3(\alpha, \beta) &= \min\{\alpha, \beta\} & n = \infty \end{aligned}$$

Since we have three different kinds of relationships among two concepts in an AN the kind and strength of a path among two arbitrary concepts in the network is determined as shown in Tables 1 and 2. For a formal discussion of that issue we refer to the work of [7, 8, 26].

For all  $0 \leq \alpha, \beta \leq 1$  holds that  $\tau_1(\alpha, \beta) \leq \tau_2(\alpha, \beta) \leq \tau_3(\alpha, \beta)$ . Each triangular norm is monotonic, commutative and associative, and can be used as axiomatic skeletons for fuzzy set intersection. We restrict ourselves to a pessimistic, neutral, and optimistic t-norm  $\tau_1$ ,  $\tau_2$  and  $\tau_3$ , respectively.

Since these triangular norms are not mutually associative the strength of a path in an associative network depends on the direction of strength composition. This asymmetry in turn might lead to unintuitive derived results: Consider, e.g., a path consisting of just three relations among four concepts  $C_1, C_2, C_3, C_4$  with  $C_1 \Rightarrow_{g, 0.6} C_2 \Rightarrow_{g, 0.8} C_3 \Rightarrow_{p, 0.9} C_4$ . It holds that  $\tau_2(\tau_3(0.6, 0.8), 0.9) = 0.54$ , but the strength of the same path in opposite direction is  $\tau_2(\tau_2(0.9, 0.8), 0.6) = 0.43$ . According to Fankhauser and Neuhold [8] we can avoid this asymmetry by imposing a precedence relation ( $3 > 2 > 1$ ) for strength combination (see Table 3).

The computation of semantic distances among concepts is used in most of the filters of the matching process. We will now describe each of the filters in detail.

Table 2. Strength of paths in an AN

	g	s	p
g	$\tau_3$	$\tau_1$	$\tau_2$
s	$\tau_1$	$\tau_3$	$\tau_2$
p	$\tau_2$	$\tau_2$	$\tau_2$

Table 3. Computational precedence for the strength of a path

	g	s	p
g	2	3	1
s	1	2	1
p	1	1	3

#### 4.1.5. The filters of the matchmaking process

*4.1.5.1. Context matching.* Any matching of two specifications has to be in an appropriate context. In LARKS to deal with restricting the advertisement matching space to those in the same domain as the request, each specification supplies a list of keywords meant to describe the semantic domain of the service. When comparing two specifications it is assumed that their context or domains are the same (or at least sufficiently similar) as long as (1) the real-valued distances between the roots of considered words do not exceed a given threshold, and (2) the distance between the attached concepts of the pairs of most similar words does not exceed a threshold.

Word distance is computed using the trigger-pair model [37]. If two words are significantly co-related, then they are considered trigger-pairs, and the value of the co-relation is domain specific. In the current implementation we use the Wall Street Journal corpus of one million word pairs to compute the word distance.

For example, both specifications ‘ReqAirMissions’ and ‘AWACS-AirMissions’ (see Example 3.4) pass the context filter as to be in a sufficiently similar context. The most similar word pairs are (Attack, Combat), (Mission, Mission), and the concept AirMission subsumes the concept AWACS-AirMission.

To summarize, the context matching consists of two consecutive steps:

1. For every pair of words  $u, v$  given in the `Context` slots compute the real-valued word distances  $d_w(u, v) \in [0, 1]$ . Determine the most similar matches for any word  $u$  by selecting words  $v$  with the minimum distance value  $d_w(u, v)$ . These distances must not exceed a given threshold.
2. For every pair of most similar matching words, check that the semantic distance among the attached concepts does not exceed a given threshold.

*4.1.5.2. Comparison of profiles.* The comparison of two profiles relies on a standard technique from the Information Retrieval area, called term frequency-inverse document frequency weighting (TF-IDF) (see [38]). According to that, any specification in LARKS is treated as a document.

Each word  $w$  in a document  $Req$  is weighted for that document in the following way. The number of times  $w$  occurs throughout all documents is called the document frequency  $df(w)$  of  $w$ . The used collection of documents is not unlimited, such as the advertisement database of the matchmaker.

Thus, for a given document  $d$ , the relevance of  $d$  based on a word  $w$  is proportional to the number  $wf(w, d)$  of times the word  $w$  occurs in  $d$  and inverse proportional to  $df(w)$ . A weight  $h(w, d)$  for a word in a document  $d$  out of a set  $D$

of documents denotes the significance of the classification of  $w$  for  $d$ , and is defined as follows:

$$h(w, d) = wf(w, d) \cdot \log\left(\frac{|D|}{df(w)}\right).$$

The weighted keyword representation  $wkv(d, V)$  of a document  $d$  contains for every word  $w$  in a given dictionary  $V$  the weight  $h(w, d)$  as an element. Since most dictionaries provide a huge vocabulary we cut down the dimension of the vector by using a fixed set of appropriate keywords determined by heuristics and the set of keywords in LARKS itself.

The similarity  $dps(Req, Ad)$  of a request  $Req$  and an advertisement  $Ad$  under consideration is then calculated by:

$$dps(Req, Ad) = \frac{Req \cdot Ad}{|Req| \cdot |Ad|}$$

where  $Req \cdot Ad$  denotes the inner product of the weighted keyword vectors. If the value  $dps(Req, Ad)$  does exceed a given threshold  $\beta \in \mathbf{R}$  both documents pass the profile filter. For example, the profiles of both specifications in Example 3.4 are similar with degree 0.65.

The matchmaker then checks if the declarations and constraints of both specifications for a request and advertisement are sufficiently similar. This is done by a pairwise comparison of declarations and constraints in two steps:

1. *Similarity matching* and
2. *Signature matching*

*4.1.5.3. Similarity matching.* The profile filter has two limitations. It does not consider the structure of the description. That means the filter, for example, is not able to differentiate among input and output declarations of a specification. Besides, profile comparison does not rely on the semantics of words themselves. Thus the filter is not able to recognize that the word pair (Computer, Notebook), for example, should have a closer distance than the pair (Computer, Book).

Computation of similarity relies on a combination of distance values as calculated for pairs of input and output declarations, and input and output constraints. Each of these distance values is computed in terms of the distance between concepts and words that occur in their respective specification section. The values are computed at the time of advertisement submittal and stored in the matchmaker database.

Let  $E_i, E_j$  be variable declarations or constraints, and  $S(E)$  the set of words in  $E$ . The similarity among two expressions  $E_i$  and  $E_j$  is determined by pairwise computation of word distances as follows:

$$Sim(E_i, E_j) = 1 - \left( \left( \sum_{(u,v) \in S(E_i) \times S(E_j)} d_w(u, v) \right) / |S(E_i) \times S(E_j)| \right)$$

The similarity value  $Sim(S_a, S_b)$  among two specifications  $S_a$  and  $S_b$  in LARKS is computed as the average of the sum of similarity computations among all pairs of declarations and constraints:

$$Sim(S_a, S_b) = \frac{\sum_{(E_i, E_j) \in (D(S_a) \times D(S_b)) \cup (C(S_a) \times C(S_b))} Sim(E_i, E_j)}{|(D(S_a) \times D(S_b)) \cup (C(S_a) \times C(S_b))|}$$

with  $D(S)$  and  $C(S)$  denoting the input/output declaration and input/output constraint part of a specification  $S$  in LARKS, respectively. Both specifications in Example 3.4 pass the similarity filter with a similarity value of 0.83.

**4.1.5.4. Signature matching.** The similarity filter takes into consideration the semantics of individual words in the description. However, it does not take the meaning of the logical constraints in a LARKS specification into account. This is done in our matchmaking process by the signature and constraint filters. The two filters are designed to work together to look for a so-called semantic plug-in match known in the software engineering area [16, 20, 50].

The signature filter first considers the declaration parts of the request and the advertisement, and determines pairwise if their signatures of the (input or output) variable types match following the type inference rules given below.

**Definition 4.1 (Subtyping Inference Rules).** Consider two types  $t_1$  and  $t_2$  as part of an input or output variable declaration part (in the form `Input  $v : t_1$` ; or `Output  $v : t_2$` ;) in a LARKS specification.

1. Type  $t_1$  is a subtype of type  $t_2$  (denoted as  $t_1 \preceq_{st} t_2$ ) if this can be deduced by the following subtype inference rules.
2. Two types  $t_1, t_2$  are equal ( $t_1 =_{st} t_2$ ) if  $t_1 \preceq_{st} t_2$  and  $t_2 \preceq_{st} t_1$  with
  - (a)  $t_1 =_{st} t_2$  if they are identical  $t_1 = t_2$
  - (b)  $t_1 \mid t_2 =_{st} t_2 \mid t_1$  (commutative)
  - (c)  $(t_1 \mid t_2) \mid t_3 = t_1 \mid (t_2 \mid t_3)$  (associative)

*Subtype Inference Rules:*

- (1)  $t_1 \preceq_{st} t_2$  if  $t_2$  is a type variable
- (2)  $\frac{t_1 =_{st} t_2}{t_1 \preceq_{st} t_2}$
- (3)  $t_1, t_2$  are sets,  $\frac{t_1 \subseteq t_2}{t_1 \preceq_{st} t_2}$
- (4)  $t_1 \preceq_{st} t_1 \mid t_2$
- (5)  $t_2 \preceq_{st} t_1 \mid t_2$
- (6)  $\frac{t_1 \preceq_{st} t_2, s_1 \preceq_{st} s_2}{(t_1, s_1) \preceq_{st} (t_2, s_2)}$

$$\begin{aligned}
(7) \quad & \frac{t_1 \preceq_{st} t_2, s_1 \preceq_{st} s_2}{t_1 \mid s_1 \preceq_{st} t_2 \mid s_2} \\
(8) \quad & \frac{t_1 \preceq_{st} t_2}{\text{SetOf}(t_1) \preceq_{st} \text{SetOf}(t_2)} \\
(9) \quad & \frac{t_1 \preceq_{st} t_2}{\text{ListOf}(t_1) \preceq_{st} \text{ListOf}(t_2)}
\end{aligned}$$

Matching of two signatures  $sig$  and  $sig'$  is defined by a binary string-valued function  $fsm$  on signatures with

$$fsm(sig, sig') = \begin{cases} sub & sig' \preceq_{st} sig \\ Sub & sig \preceq_{st} sig' \\ eq & sig =_{st} sig' \\ disj & else \end{cases}$$

Having described both filters of the syntactical matching we now define the meaning of syntactical matching of two specifications written in LARKS.

**Definition 4.2 (Syntactical matching of specifications in LARKS).** Consider two specifications  $S_a$  and  $S_b$  in LARKS with  $n_k$  input declarations,  $m_k$  output declarations, and  $v_k$  constraints  $n_k, m_k \in \mathbf{N}, k \in \{a, b\}$ , two declarations  $D_i, D_j$ , and constraints  $C_i, C_j$  in these specifications, and  $V$  a given dictionary for the computation of weighted keyword vectors. Let  $\beta, \gamma, \theta$  be real threshold values for profile comparison and similarity matching.

- *Declarations  $D_i$  and  $D_j$  syntactically match* if they are sufficiently similar:

$$Sim(D_i, D_j) \geq \gamma \wedge fsm(D_i, D_j) \neq disj.$$

- *Constraints  $C_i$  and  $C_j$  syntactically match* if they are sufficiently similar:

$$Sim(C_i, C_j) \geq \gamma.$$

If both words in every pair  $(u, v) \in S(E_i) \times S(E_j)$  of most similar words are associated with a concept  $C$  and  $C'$ , respectively, then the distance among  $C$  and  $C'$  in the so-called associative network of the matchmaker must not exceed a given threshold value  $\theta$ .

The syntactical match of two declarations or constraints is denoted by a boolean predicate *Synt*.

- The *specifications  $S_a$  and  $S_b$  syntactically match* if
  1. their profiles match, that is,  $dps(S_a, S_b) \geq \beta$ , and
  2. for each declaration or constraint  $E_i, i \in \{1, \dots, n_a\}$  in the declaration or constraint part of  $S_a$  there exists a most similar matching declaration or constraint

$E_j, j \in \{1, \dots, n_b\}$  in the declaration or constraint part of  $S_b$  such that

$$\text{Synt}(E_i, E_j) \wedge \text{Sim}(E_i, E_j) = \max\{\text{Sim}(E_i, E_y), y \in \{1, \dots, n_b\}\}$$

(Analogous for each declaration or constraint in  $S_b$ .)

3. for each pair of declarations determined in (2.) the matching of their signatures is of the same type, that is, for each  $(D_i, D_j)$  in (2.) it holds that the value  $\text{fsm}(D_i, D_j)$  is the same, and
4. the similarity value  $\text{Sim}(S_a, S_b)$  exceeds a given threshold.

**4.1.5.5. Constraint matching.** By using the syntactical filter many matches might be found in a large agent society. Hence, it is important to use some kind of semantic information (other than optionally attached concepts and the associative network) to narrow the search, and to pin down more precise matches. This is done by the constraint filter.

The most common and natural interpretation for a specification (even for a software program) is using sets of pre- and post-conditions, denoted as  $\text{Pre}_S$  and  $\text{Post}_S$ , respectively. In a simplified notation, any specification  $S$  can be represented by the pair  $(\text{Pre}_S, \text{Post}_S)$ .

A software component description  $D_2$  ‘semantically plug-in matches’ another component description  $D_1$  if (1) their signatures match, (2) the set of input constraints of  $D_1$  logically implies that of  $D_2$ , and (3) set of output constraints of  $D_2$  logically implies that of  $D_1$ . In our implementation the logical implication among constraints is computed using polynomial  $\theta$ -subsumption checking for Horn clauses [31].

**Definition 4.3 (Constraint-based semantic matching of two specifications).** Consider two specifications  $S(\text{Pre}_S, \text{Post}_S)$  and  $T(\text{Pre}_T, \text{Post}_T)$ .

The specification  $T$  *semantically matches* the specification  $S$  if

$$(\text{Pre}_S \Rightarrow \text{Pre}_T) \wedge (\text{Post}_T \Rightarrow \text{Post}_S)$$

That means, the set of pre-conditions of  $S$  logically implies that of  $T$ , and the set of post-conditions of  $S$  is logically implied by that of  $T$ .

Plug-in matching of LARKS specifications is valuable for selecting advertisements which are not as constrained in the input parameters as the considered request, but will return equal or greater number of more specific output parameters. For example, the advertisement ‘AWAC-AirMission’ plugs into the request ‘ReqAirMissions’ in Example 3.4.

The problem in performing the semantical matching is that the logical implication is not decidable for first order predicate logic, and even not for an arbitrary set of Horn clauses. To make the matching process tractable and feasible, we have to decide on the expressiveness of the language used to represent the pre- and post-conditions, and to choose a relation that is weaker than logical implication. The  $\theta$ -subsumption relation [31] among two constraints  $C, C'$  (denoted as  $C \leq_{\theta} C'$ ) appears to be a suitable choice for semantical matching, because it is computationally tractable and semantically sound.

### Plug-in Semantical Matching in LARKS

It is proven in the software engineering area that if the condition of semantical matching in Definition 4.3 holds, and the signatures of both specifications match, then  $T$  can be directly used in the place of  $S$ , that is,  $T$  plugs in  $S$ .

**Definition 4.4 (Plug-in semantical matching of two specifications).** Given two specifications  $Spec1$  and  $Spec2$  in LARKS then  $Spec2$  **plug-in matches**  $Spec1$  if

- The signatures of their variable declaration parts matches (Section 4.1.4.3).
- For every clause  $C1$  in the set of input constraints of  $Spec1$  there is a clause  $C2$  in the set of input constraint of  $Spec2$  such that  $C1 \preceq_{\theta} C2$ .
- For every clause  $C2$  in the set of output constraints of  $Spec2$  there is a clause  $C1$  in the set of output constraints of  $Spec1$  such that  $C2 \preceq_{\theta} C1$ .

where  $\preceq_{\theta}$  denotes the  $\theta$ -subsumption relation between constraints.

**$\theta$ -Subsumption between constraints.** One suitable selection of the language and the relation is the (definite program) clause and the so-called  $\theta$ -subsumption relation between clauses, respectively [31].<sup>7</sup> In the following we will only consider Horn clauses. A general form of Horn clause is  $a_0 \vee (\neg a_1) \vee \dots \vee (\neg a_n)$ , where each  $a_i, i \in \{1, \dots, n\}$  is an atom. This is equivalent to  $a_0 \vee \neg(a_1 \wedge \dots \wedge a_n)$ , which in turn is equivalent to  $(a_1 \wedge \dots \wedge a_n) \Rightarrow a_0$ .<sup>8</sup> We adopt the standard notation for that clause as  $a_0 \leftarrow a_1, \dots, a_n$ ; in PROLOG the same clause is written as  $a_0: a_1, \dots, a_n$ .

Examples of definite program clauses are

- $Date.year > 1995, sorted(computerInfo)$ ,
- $before(x, y, ys) \leftarrow ge(x, y)$ , and
- $scheduleMeeting(group1, group2, interval, meetingDuration, meetTime) \leftarrow belongs \times (p1, group1), belongs(p2, group2), subset(meetTime, interval), length(meetTime) = meetingDuration, available(p1, meetTime), available(p2, meetTime)$ .

We say that a clause  $C$   **$\theta$ -subsumes** another clause  $D$  (denoted as  $C \succeq_{\theta} D$ ) if there is a substitution  $\theta$  such that  $C\theta \subseteq D$ .  $C$  and  $D$  are  $\theta$ -equivalent if  $C \preceq_{\theta} D$  and  $D \preceq_{\theta} C$ .

Examples of  $\theta$ -subsumption between clauses are

- $P(a) \leftarrow Q(a) \preceq_{\theta} P(X) \leftarrow Q(X)$
- $P(X) \leftarrow Q(X), R(X) \preceq_{\theta} P(X) \leftarrow Q(X)$ .

Since a single clause is not expressive enough, we need to use a set of clauses to express the pre and post conditions (that is, the input and output constraints) of a specification in LARKS. A set of clauses is treated as a conjunction of those clauses.

Subsumption between two set of clauses is defined in terms of the subsumption between single clauses. More specifically, let  $S$  and  $T$  be such sets of clauses. Then, we define that  $S$   $\theta$ -subsumes  $T$  if every clause in  $T$  is  $\theta$ -subsumed by a clause in  $S$ .

There is a complete algorithm to test the  $\theta$ -subsumption relation, which is in general NP-complete but polynomial in certain cases. On the other hand,  $\theta$ -subsumption

is a weaker relation than logical implication, that is, from  $C \preceq_{\theta} D$  we can only infer that  $C$  logically implies  $D$  but not vice versa.<sup>9</sup>

## 5. Examples of matchmaking using LARKS

Consider the specifications ‘IntegerSort’ and ‘GenericSort’ (see Examples 3.1 and 3.2) as a request of sorting integer numbers and an advertisement for some agent’s capability of sorting real numbers and strings, respectively.

IntegerSort	
Context	Sort
Types	
Input	xs: ListOf Integer;
Output	ys: ListOf Integer;
InConstraints	le(length(xs),100);
OutConstraints	before(x,y,ys) < - ge(x,y); in(x,ys) < - in(x,xs);
ConcDescriptions	
TextDescription	sort of list of at most 100 integer numbers
GenericSort	
Context	Sorting
Types	
Input	xs: ListOf Real   String;
Output	ys: ListOf Real   String;
InConstraints	
OutConstraints	before(x,y,ys) < - ge(x,y); before(x,y,ys) < - precedes(x,y); in(x,ys) < - in(x,xs);
ConcDescriptions	
TextDescription	sorting a list of real numbers or strings

Assume that the requester and provider agent sends the request IntegerSort and advertisement GenericSort to the matchmaker, respectively. Figure 3 describes the overall matchmaking process for that request.

1. *Context Matching.* Both words in the Context declaration parts are sufficiently similar. We have no referenced concepts to check for terminologically equity. Thus, the matching process proceeds with the following two filtering stages.
2. *Syntactical Matching.*
  - (a) *Comparison of Profiles.* According to the result of TF-IDF method both specifications are sufficiently similar:
  - (b) *Signature Matching.* Consider the signatures  $t_1 = (\text{ListOf Integer})$  and  $t_2 = (\text{ListOf Real|String})$ . Following the subtype inference rules 9., 4., and 1. it holds that  $t_1 \preceq_{st} t_2$ , but not vice versa, thus  $fsm(D_{11}, D_{21}) = \text{sub}$ . Analogous for  $fsm(D_{12}, D_{22}) = \text{sub}$ .



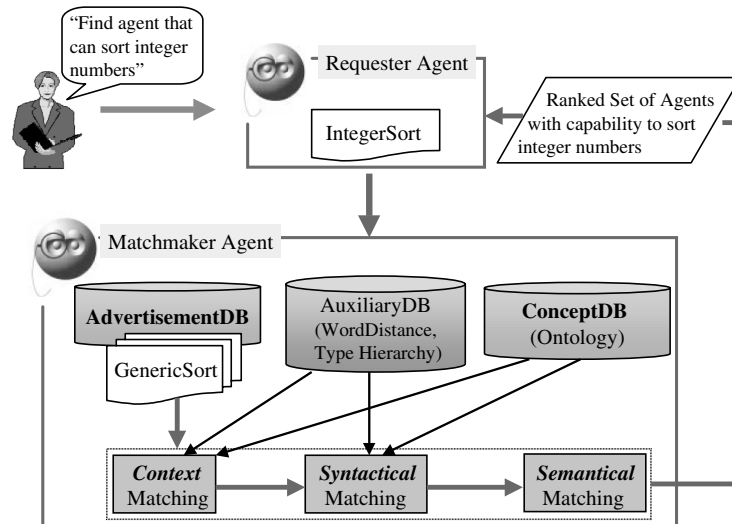


Figure 3. An example of matchmaking using LARKS.

- (c) *Similarity Matching*. Using the current auxiliary database for word distance values similarity matching of constraints yields:

$le(\text{length}(xs), 100)$	$null$	$= 1.0$
$before(x, y, ys) < -ge(x, y)$	$in(x, ys) < -in(x, xs)$	$= 0.5729$
$in(x, ys) < -in(x, xs)$	$before(x, y, ys) < -preceeds(x, y)$	$= 0.4375$
$before(x, y, ys) < -ge(x, y)$	$before(x, y, ys) < -preceeds(x, y)$	$= 0.28125$

The similarity of both specifications is computed as:

$$Sim(\text{IntegerSort}, \text{GenericSort}) = 0.64.$$

3. *Constraint Matching*. The advertisement *GenericSort* also plug-in matches with the request *IntegerSort*, because the set of input constraints of *IntegerSort* is  $\theta$ -subsumed by that of *GenericSort*, and the output constraints of *GenericSort* are  $\theta$ -subsumed by that of *IntegerSort*. Thus *GenericSort* plugs into *IntegerSort*. Please note that this does not hold vice versa.

## 6. Implementation

We did implement the language LARKS and the matchmaking process using LARKS in Java. Figure 4 shows the user interface of the matchmaker agent.

To help visualize the matchmaking process, we devised a user interface that traces the path of the advertisement result set for a request through the matchmaker's filters. The filters can be configured by selecting the checkboxes beneath the desired

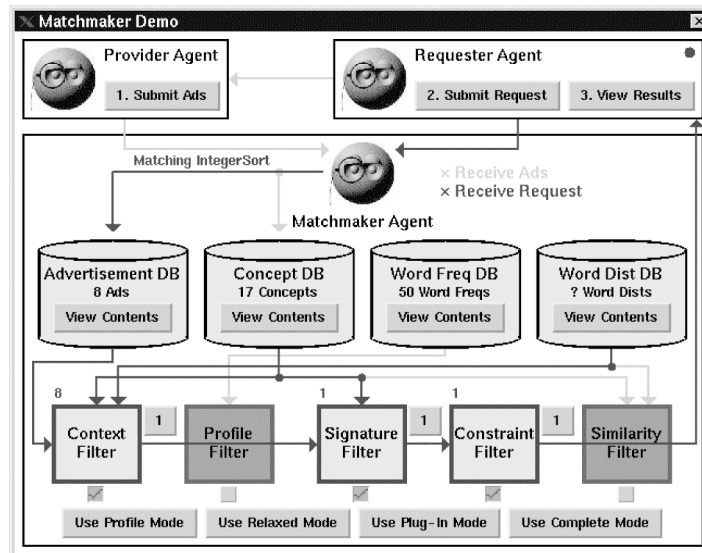


Figure 4. The user interface of the matchmaker agent.

filters—disabled filters are darkened and bypassed. As the result set passes from one filter to the next, the filter’s outline highlights, the number above the filter increments as it considers an advertisement, and the number above its output arrow increments as advertisements successfully pass through the filter. Pushing the buttons above each inter-filter arrow reveals the result advertisement set for the preceding filter.

## 7. Related work

For dealing with semantic heterogeneity among distributed, autonomous information sources there exist solutions in the multidatabase and information systems area for years. Many of them are based on a database-style modeling of data, global schema, and use of meta-information such as provided by a common ontology or different domain ontologies for a content-based source selection [2, 14, 15, 39]. Others focus on information retrieval (IR) techniques for best-match queries, and relevance assessment. Alternative solutions towards an adaptive process for revealing semantic interdependencies among heterogeneous data objects is proposed, for example, by SCOPES [34].

However, the main problem of dynamic matchmaking in the Internet is to deal with the trade-off between performance and quality of matching. Complex reasoning has to be restricted to allow meaningful semantic matches of requests and advertisements in a reasonable time. Unlike other approaches to matchmaking or brokering in multi-agent systems [2, 28, 33], the presented matchmaking process using LARKS offers a flexible approach to satisfy both requirements. It does not deal with a global integration of heterogeneous source descriptions in terms of database schemas, but

with comparing descriptions of functional capabilities such as constrained actions to provide services. For this purpose it combines techniques from IR, software engineering and description logics area in an appropriate way to perform such filtering efficiently. The matchmaker agent does not need to perform any complex query activities such as, for example, by broker agents in InfoSleuth [33] or the mediator agent in SIMS [2]. In addition, we have developed protocols for efficient, distributed matchmaking among multiple matchmaker agents [19]. We now discuss some of the related works in a more detail.

### 7.1. *Work related to matchmaking and mediation*

The earliest matchmaker we are aware of is the ABSI facilitator, which is based on the KQML specification and uses the KIF as the content language. The KIF expression is basically treated like the Horn clauses. The matching between the advertisement and request expressed in KIF is the simple unification with the equality predicate. Matchmaking using LARKS performs better than ABSI in both, the language and the matching process. The plug-in matching in LARKS uses the  $\theta$ -subsumption test, which select more matches that are also semantically matches.

The SHADE and COINS [28] are matchmakers based on KQML. The content language of COINS allows for the free text and its matching algorithm utilizes the tf-idf. The content language of SHADE matchmaker consists of two parts, one is a subset of KIF, another is a structured logic representation called MAX. MAX uses logic frames to declaratively store the knowledge. SHADE uses a frame like representation and the matcher uses the prolog like unifier.

A more recent service broker-based information system is InfoSleuth [18, 33]. The content language supported by InfoSleuth is KIF and the deductive database language LDL++, which has a semantics similar to Prolog. The constraints for both the user request and the resource data are specified in terms of some given central ontology. It is the use of this common vocabulary that enables the dynamic matching of requests to the available resources. The advertisements specify agents' capabilities in terms of one or more ontologies. The constraint matching is an intersection function between the user query and the data resource constraints. If the conjunction of all the user constraints with all the resource constraints is satisfiable, then the resource contains data which are relevant to the user request.

Another related research area is that on mediators among heterogeneous information systems [1, 45]. Each local information system is wrapped by a so-called wrapper agent and their capabilities are described in two levels. One is what they can provide, usually described in the local data model and local database schema. Another is what kind of queries they can answer; usually it is a subset of the SQL language. The set of queries a service can accept is described using a grammar-like notation. The matching between the query and the service is simple: it just decides whether the query can be generated by this grammar. This area emphasizes the planning of database queries according to heterogeneous information systems not providing complete SQL services. Those systems are not supposed to be searched for among a vast number of resources on the Internet. The description of capabil-

ities and matching are not only studied in the agent community, but also in other related areas.

### 7.2. *Work related to capability description*

The problem of capability and service descriptions can be tackled at least from the following different approaches:

1. *Software specification techniques.* Agents are computer programs that have some specific characteristics. There are numerous work for software specifications in formal methods, like model-oriented VDM and Z [35], or algebraic-oriented Larch. Although these languages are good at describing computer programs in a precise way, the specification usually contains too much details to be of interests to other agents. Besides, those existing languages are so complex that the semantic comparison between the specifications is impossible. The reading and writing of these specifications also require substantial training.
2. *Action representation formalisms.* Agent capability can be seen as the actions that the agents perform. There are a number of action representation formalisms in AI planning like the classical one the STRIPS. The action representation formalism are inadequate in our task in that they are propositional and not involving data types.
3. *Concept languages for knowledge representation.* There are various terminological knowledge representation languages. However, ontology itself does not describe capabilities. On the other hand, it provides auxiliary concepts to assist the specification of the capabilities of agents.
4. *Database query capability description.* The database query capability description technique is developed as an attempt to describe the information sources on the Internet, such that an automated integration of information is possible. In this approach the information source is modeled as a database with restricted querying capabilities.

### 7.3. *Work related to service retrieval*

There are three broad approaches to service retrieval. One is the information retrieval techniques to search for relevant information based on text, another is the software component retrieval techniques [16, 20, 50] to search for software components based on software specifications. The third one is to search for Web resources that are typically described as database models [30, 45].

In the software component search techniques, Zaremski and Wing [50] defined several notions of matches, including the exact match and the plug-in match, and formally proved the relationship between those matches. Goguen et al. [16] proposed to use a sequence of filters to search for software components, in order to increase the efficiency of the search process. Jeng and Cheng [20] computed the distance between similar specifications. All of these works are based on the algebraic

specification of computer programs. No concept description and concept hierarchy are considered in their work.

In Web resource search techniques, Li and Danzig [30] proposed a method to look for better search engines that may provide more relevant data for the user concerns, and rank those search engines according to their relevance to user's query. They propose the directory of services to record descriptions of each information server, called a server description. A user sends his query to the directory of services, which determines and ranks the servers relevant to the user's request. Both the query and the server are described using boolean expression. The search method is based on the similarity measure between the two boolean expressions.

## 8. Conclusion

The Internet is an open system where heterogeneous agents can appear and disappear dynamically. As the number of agents on the Internet increases, there is a need to define middle agents to help agents locate others that provide requested services. In prior research, we have identified a variety of middle agent types, their protocols and their performance characteristics. Matchmaking is the process that brings requester and service provider agents together. A provider agent advertises its know-how, or capability to a middle agent that stores the advertisements. An agent that desires a particular service sends a middle agent a service request that is subsequently matched with the middle agent's stored advertisements. The middle agent communicates the results to the requester (the way this happens depends on the type of middle agent involved). We have also defined protocols that allow more than one middle agent to maintain consistency of their advertisement databases. Since matchmaking is usually done dynamically and over large networks, it must be efficient. There is an obvious trade-off between the quality and efficiency of service matching in the Internet.

We have defined and implemented a language, called LARKS, for agent advertisement and request and a matchmaking process using LARKS. LARKS judiciously balances language expressivity and efficiency in matching. LARKS performs both syntactic and semantic matching, and in addition allows the specification of concepts (local ontologies) via ITL, a concept language.

The matching process uses five filters, namely context matching, comparison of profiles, similarity matching, signature matching and semantic matching. Different degrees of partial matching can result from utilizing different combinations of these filters. Selection of filters to apply is under the control of the user (or the requester agent).

## Acknowledgments

We would like to thank Davide Brugali for helpful discussions, and Zhendong Niu and Seth Widoff for help with the implementation of the matchmaker agent using LARKS.

## Notes

1. In the future, we plan to allow for using ISO/IEC 13211-1 standard compliant Prolog programs to describe constraints and functional capabilities.
2. For syntax and set-theoretical semantics of used concept language ITL we refer to [43].
3. For methods of determining subsumption or equality of concepts defined in an ontology using a concept language such as ITL we refer to Section 3.3.2.
4. For a further discussion on possible loss of semantics due to mapping among multiple different ontologies we refer to for example [40].
5. This is similar to the common use of domain namespaces in XML [49] for semantically tagging Web page contents.
6. The relationships are fuzzy, and one cannot possibly associate all concepts with each other.
7. A *clause* is a finite set of *literals*, which is treated as the universally quantified disjunction of those *literals*. A *literal* may be positive or negative. A positive *literal* is an *atom*, a negative literal is the negation of an *atom*. A *definite program clause* is a clause with one positive literal and zero or more negative literals. A *definite goal* is a clause without positive literals. A *Horn clause* is either a definite program clause or a definite goal.
8. The literal  $a_0$  is called the head of the clause, and  $(a_1 \wedge \dots \wedge a_n)$  is called the body of the clause.
9. Please also note that the  $\theta$ -subsumption relation is similar to the query containment in database. When advertisements are database queries, specification matching is reduced to the problem of query containment testing.

## References

1. J. L. Ambite and C. A. Knoblock, "Planning by rewriting: Efficiently generating high-quality plans," in *Proc. Fourteenth National Conf. Artif. Intell.*, Providence, RI, 1997.
2. Y. Arens, C. A. Knoblock, and C. Hsu, "Query processing in the SIMS information mediator," in A. Tate (ed.), *Advanced Planning Technology*, AAAI Press: CA, 1996.
3. R. J. Brachman and J. G. Schmolze, "An overview of the KL-ONE knowledge representation system," *Cognitive Science*, vol. 9, no. 2, pp. 171–216, 1985.
4. J. E. Caplan and M. T. Harandi, "A logical framework for software proof reuse," in *Proc. ACM SIGSOFT Symp. Software Reusability, April 1995*. ACM Software Engineering Note, 1995.
5. S. Cranefield, A. Diaz, and M. Purvis, "Planning and matchmaking for the interoperation of information processing agents," The Information Science Discussion Paper Series No. 97/01, University of Otago, 1997.
6. K. Decker, K. Sycara, and M. Williamson, "Middle-agents for the internet," in *Proc. 15th IJCAI*, Nagoya, Japan, August 1997, pp. 578–583.
7. P. Fankhauser, M. Kracker, and E. J. Neuhold, "Semantic vs. structural resemblance of classes. Special issue: Semantic issues in multidatabase systems," *ACM SIGMOD RECORD*, vol. 20, no. 4, pp. 59–63, 1991.
8. P. Fankhauser and E. J. Neuhold, "Knowledge based integration of heterogeneous databases," in *Proc. IFIP Conf. DS-5 Semantics of Interoperable Database Systems*, Lorne, Victoria, Australia, 1992.
9. C. Fellbaum (ed.), *WordNet: An Electronic Lexical Database*, MIT Press, 1998. <http://www.cogsci.princeton.edu/wn/>
10. T. Finin, R. Fritzon, D. McKay, and R. McEntire, "KQML as an Agent Communication Language," in *Proc. 3rd Int. Conf. Information and Knowledge Management CIKM-94*, ACM Press, 1994.
11. FIPA: Foundation for Intelligent Physical Agents. <http://drogo.cse.it/fipa/>, see also, L. Chiariglione, "FIPA—Agent technologies achieve maturity," *AgentLink Newsletter*, vol. 1, November 1999, <http://www.agentlink.org>
12. FIPA Agent Communication Language. <http://www.fipa.org/spec/fipa99spec.htm>, 1999.
13. M. Fowler, *UML Distilled: Applying the Standard Object Modeling Language*, Addison-Wesley: Reading, MA, 1997.

14. H. Garcia-Molina, et al., "The TSIMMIS approach to mediation: Data models and languages," in *Proc. Workshop NGITS*, 1995. <ftp://db.stanford.edu/pub/garcia/1995/tsimmis-models-languages.ps>
15. M. R. Genesereth, A. M. Keller, and O. Duschka, "Infomaster: An information integration system," in *Proc. ACM SIGMOD Conference*, May 1997.
16. J. Goguen, D. Nguyen, J. Meseguer, Luqi, D. Zhang, and V. Berzins, "Software component search," *J. Systems Integration*, vol. 6, pp. 93–134, 1996.
17. G. Huck, P. Fankhauser, K. Aberer, and E. J. Neuhold, "Jedi: Extracting and synthesizing information from the web," in *Proc. Int. Conf. Cooperative Information Systems CoopIS'98*, IEEE Computer Society Press, 1998.
18. N. Jacobs and R. Shea, "Carnot and InfoSleuth—Database technology and the WWW," in *ACM SIGMOD Int. Conf. Management of Data*, May 1995.
19. S. Jha, P. Chalasani, O. Shehory, and K. Sycara, "A formal treatment of distributed matchmaking," in *Proc. Second Int. Conf. Autonomous Agents (Agents 98)*, Minneapolis, MN, May 1998.
20. J.-J. Jeng and B. H. C. Cheng, "Specification matching for software reuse: A foundation," in *Proc. ACM SIGSOFT Symposium Software Reusability*, ACM Software Engineering Note, Aug. 1995.
21. V. Kashyap and A. Sheth, "Semantic heterogeneity in global information systems: The role of meta-data context and ontology," in M. P. Papazoglou and G. Schlageter (eds.), *Cooperative Information Systems: Trends and Directions*, Academic Press, 1998.
22. KIF. Knowledge Interchange Format: <http://logic.stanford.edu/kif/>
23. W. Kim, et al., "On resolving schematic heterogeneity in multidatabase systems," *Intl. J. on Distributed and Parallel Databases*, vol. 1, pp. 251–279, 1993.
24. M. Klusch, "Cooperative information agents on the Internet," Ph.D. Thesis, University of Kiel, December 1996 (in German) Kovac Verlag, Hamburg, 1998, ISBN 3-86064-746-6.
25. M. Klusch (ed.), *Intelligent Information Agents*, Springer, ISBN 3-540-65112-8, 1999.
26. M. Kracker, "A fuzzy concept network," in *Proc. IEEE Int. Conf. Fuzzy Systems*, 1992.
27. R. Kruse, E. Schwecke, and J. Heinsohn, *Uncertainty and Vagueness in Knowledge Based Systems*, Springer, 1991.
28. D. Kuokka and L. Harrada, "On using KQML for matchmaking," in *Proc. 3rd Int. Conf. on Information and Knowledge Management CIKM-95*, AAAI/MIT Press, 1995, pp. 239–45.
29. K. C. Lano, *Formal Object-oriented Development*. Formal Approaches to Computing and Information Technology Series, Springer-Verlag, 1995.
30. S.-H. Li and P. B. Danzig, "Boolean similarity measures for resource discovery," *IEEE Trans. on Knowledge and Data Engineering*, vol. 9, no. 6, November/December, 1997.
31. S. Muggleton and L. De Raedt, "Inductive logic programming: Theory and methods," *J. of Logic Programming*, vol. 19, no. 20, pp. 629–679, 1994.
32. B. Nebel, *Reasoning and Revision in Hybrid Representation Systems*, Lecture Notes in Artificial Intelligence LNAI Series, vol. 422, Springer, 1990.
33. M. Nodine and J. Fowler, "An overview of active information gathering in Infosleuth," *Proc. Int. Conf. on Autonomous Agents*, USA, 1999, submitted.
34. A. Ouksel, "A framework for a scalable agent architecture of cooperating heterogeneous knowledge sources," in M. Klusch (ed.), *Intelligent Information Agents*, chap. 5, Springer, 1999.
35. B. Potter, J. Sinclair, and D. Till, *Introduction to Formal Specification and Z*, Prentice-Hall International Series in Computer Science, 1996.
36. Resource Description Framework (RDF) Schema Specification. <http://www.w3.org/TR/WD-rdf-schema/>.
37. R. Rosenfield, "Adaptive statistic language model," Ph.D. thesis, Carnegie Mellon University, 1994.
38. G. Salton, *Automatic Text Processing: The Transformation, Analysis and Retrieval of Information by Computer*, Addison-Wesley, Reading, MA, 1989.
39. A. Sheth, E. Mena, A. Illaramendi, and V. Kashyap, "OBSERVER: An approach for query processing in global information systems based on interoperation across pre-existing ontologies," in *Proc. Int. Conf. on Cooperative Information Systems CoopIS-96*, IEEE Computer Soc. Press, 1996.
40. A. Sheth, A. Illaramendi, V. Kashyap, and E. Mensa, "Managing multiple information sources through ontologies: Relationship between vocabulary heterogeneity and loss of information," in *Proc. ECAI-96*, Budapest, 1996.

41. G. Smolka and B. Nebel, "Representation and reasoning with attributive descriptions," IWBS Report 81, IBM Deutschland Wissenschaftl, Zentrum, 1989.
42. G. Smolka and M. Schmidt-Schauss, "Attributive concept description with complements," *AI* vol. 48, 1991.
43. K. Sycara, J. Lu, and M. Klusch, "Interoperability among heterogeneous software agents on the Internet," Carnegie Mellon University, PA, Technical Report CMU-RI-TR-98-22.
44. K. Sycara, K. Decker, A. Pannu, M. Williamson, and D. Zeng, "Distributed intelligent agents," *IEEE Expert*, pp. 36–46, December 1996.
45. V. Vassalos and Y. Papakonstantinou, "Expressive Capabilities Description Languages and Query Rewriting Algorithms," available at <http://www-cse.ucsd.edu/yannis/papers/vpcap2.ps>
46. G. Wickler, "Using Expressive and Flexible Action Representations to Reason about Capabilities for Intelligent Agent Cooperation," <http://www.dai.ed.ac.uk/students/gw/phd/story.html>
47. WIDL, "The W3C Web Interface Definition Language," <http://www.w3.org/TR/NOTE-widl>
48. WordNet—A Lexical Database for English. <http://www.cogsci.princeton.edu/wn/>
49. XML, "Extensible markup language," World Wide Web Consortium (W3C) Working Draft, 17 November 1997. <http://www.w3.org/TR/WD-xml-link>.
50. A. M. Zaremski and J. M. Wing, "Specification matching of software components," Technical Report CMU-CS-95-127, 1995.



## Hybrid Semantic Matching of OWL-S Services

M. Klusch, B. Fries, K. Sycara: Automated Semantic Web Service Discovery with OWLS-MX. Proceedings of the 5th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), Hakodate, Japan, ACM Press, 2006.

# Automated Semantic Web Service Discovery with OWLS-MX \*

Matthias Klusch,

German Research Center for  
Artificial Intelligence  
Multiagent Systems Group  
Saarbruecken, Germany  
klusch@dfki.de

Benedikt Fries

University of the Saarland  
Computer Science  
Department  
Saarbruecken, Germany  
Develin@gmx.de

Katia Sycara

Carnegie Mellon University  
Robotics Institute  
Pittsburgh PA, USA  
katia+@cs.cmu.edu

## ABSTRACT

We present an approach to hybrid semantic Web service matching that complements logic based reasoning with approximate matching based on syntactic IR based similarity computations. The hybrid matchmaker, called OWLS-MX, applies this approach to services and requests specified in OWL-S. Experimental results of measuring performance and scalability of different variants of OWLS-MX show that under certain constraints logic based only approaches to OWL-S service I/O matching can be significantly outperformed by hybrid ones.

## Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Retrieval models; H.4 [Information Systems Applications]: Miscellaneous

## Keywords

OWL-S, matchmaking, information retrieval

## 1. INTRODUCTION

Key to the success of effectively retrieving relevant services in the future semantic Web is how well intelligent service agents may perform semantic matching in a way that goes far beyond of what standard service discovery protocols such as UPnP, Jini, or Salutation-Lite can deliver. Central to the majority of contemporary approaches to semantic Web service matching is that the formal semantics of services specified, for example, in OWL-S or WSMO are explicitly

\*This work has been supported by the German Ministry of Education and Research (BMBF 01-IW-D02-SCALLOPS), the European Commission (FP6 IST-511632-CASCOM), and the DARPA DAML program under contract F30601-00-2-0592.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS 2006 May 8-12, 2006, Hakodate, Hokkaido, Japan  
Copyright 2006 ACM 1-59593-303-4/06/0005 ...\$5.00.

defined in some decidable description logic based ontology language such as OWL-DL [8] or F-Logic, respectively. This way, standard means of description logic reasoning can be exploited to automatically determine services that semantically match with a given service request based on the kind of terminological concept subsumption relations computed in the corresponding ontology. Prominent examples of such logic-based only approaches to semantic service discovery are provided by the OWLS-UDDI matchmaker [16], RACER [11], MAMA [5], and the WSMO service discovery approach [20].

These approaches do not exploit semantics that are implicit, for example, in patterns or relative frequencies of terms in service descriptions as computed by techniques from data mining, linguistics, or content-based information retrieval (IR). The objective of hybrid semantic Web service matching is to improve semantic service retrieval performance by appropriately exploiting means of both crisp logic based and approximate semantic matching where each of them alone would fail.

Consider, for example, a pair of real world concepts that are semantically synonymous or very closely related, but differing in their terminological definitions which are part of the underlying ontology. In particular, the crisp conjunctive logical concept expressions are differing with respect to a few pairs of unmatched logical constraints only. In this case, both concepts would be logically classified as disjoint siblings in a concept subsumption hierarchy such that any description logic reasoner would fail to detect the original real world semantic relationship. As a consequence, if the semantic comparison of both concepts is essential to discover services that are relevant to a given request, any logic based only matching approach would necessarily fail. The underpinning general problem is that standard logical specification of real world concept semantics is known to be inadequate. One operational way to cope with this problem would be to tolerate logical matching failures up to a specified extent by complementary approximate matching based on syntactic concept similarity computations. Of course, we acknowledge that the adaptation to the latter eventually is on the user's end.

In this paper, we present the first hybrid OWL-S service matchmaker called OWLS-MX, that exploits means of both crisp logic based and IR based approximate matching. Our experimental evaluation shows that under certain constraints this way of matching can indeed outperform logic

based only approaches.

The remainder of the paper is structured as follows. After brief background information on OWL-S in section 2, we present the hybrid matching filters, the generic algorithm of OWLS-MX together with its variants, and a simple example in section 3. Some details on the implementation of OWLS-MX version 1.1 are given in sections 4. The experimental results of measuring performance and scalability of OWLS-MX are presented in section 5, before we briefly comment on related work in section 6, and conclude in section 7.

## 2. OWL-S SERVICES

In the following, we briefly introduce the essentials of the semantic Web service description language OWL-S that are needed to understand the concepts of hybrid service matching. For more details, we refer the reader to, for example, [13].

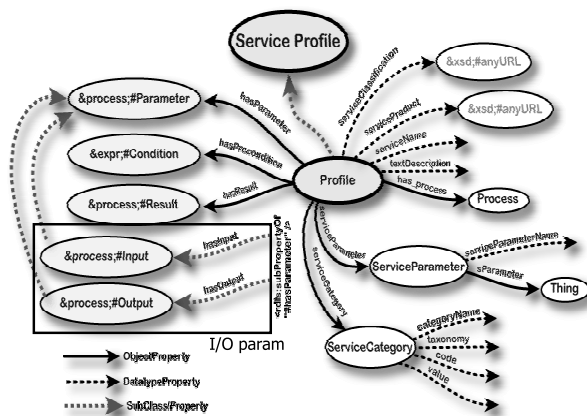


Figure 1: Parametric structure of OWL-S service profiles

OWL-S is an OWL-based Web service ontology, which supplies a core set of markup language constructs for describing the properties and capabilities of Web services in unambiguous, computer-intepretable form. The overall ontology consists of three main components: the service profile for advertising and discovering services; the process model, which gives a detailed description of a service's operation; and the grounding, which provides details on how to interoperate with a service, via messages. Specifically, it specifies the signature, that is the inputs required by the service and the outputs generated; furthermore, since a service may require external conditions to be satisfied, and it has the effect of changing such conditions, the profile describes the preconditions required by the service and the expected effects that result from the execution of the service.

To the best of our knowledge, the majority of current OWL-S matchmakers performs service I/O based profile matching that exploits defined semantics of concepts as values of service parameters `hasInput` and `hasOutput` (cf. figure 1). Exceptions include service process based approaches like in [3]. There exists no implemented matchmaker that performs an integrated service IOPE matching by means of additional reasoning on logically defined preconditions and effects. Related work on logic based semantic web rule languages such as SWRL and RuleML is ongoing.

## 3. HYBRID SERVICE MATCHING

Hybrid semantic service matching performed by the matchmaker OWLS-MX exploits both logic-based reasoning and content-based information retrieval techniques for OWL-S service profile I/O matching. In the following, we define the hybrid semantic filters of OWLS-MX, the generic OWLS-MX algorithm, and its five variants according to the used IR similarity metrics.

### 3.1 Matching filters of OWLS-MX

OWLS-MX computes the degree of semantic matching for a given pair of service advertisement and request by successively applying five different filters EXACT, PLUG IN, SUBSUMES, SUBSUMED-BY and NEAREST-NEIGHBOR. The first three are logic based only whereas the last two are hybrid due to the required additional computation of syntactic similarity values.

Let  $T$  be the terminology of the OWLS-MX matchmaker ontology specified in OWL-Lite (SHIF(D)) or OWL-DL (SHOIN(D));  $CT_T$  the concept subsumption hierarchy of  $T$ ;  $LSC(C)$  the set of least specific concepts (direct children)  $C'$  of  $C$ , i.e.  $C'$  is immediate sub-concept of  $C$  in  $CT_T$ ;  $LGC(C)$  the set of least generic concepts (direct parents)  $C'$  of  $C$ , i.e.,  $C'$  is immediate super-concept of  $C$  in  $CT_T$ ;  $Sim_{IR}(A, B) \in [0, 1]$  the numeric degree of syntactic similarity between strings  $A$  and  $B$  according to chosen IR metric  $IR$  with used term weighting scheme and document collection, and  $\alpha \in [0, 1]$  given syntactic similarity threshold;  $\doteq$  and  $\succeq$  denote terminological concept equivalence and subsumption, respectively.

**Exact match.** Service  $S$  EXACTLY matches request  $R \Leftrightarrow \forall IN_S \exists IN_R: IN_S \doteq IN_R \wedge \forall OUT_R \exists OUT_S: OUT_R \doteq OUT_S$ . The service I/O signature perfectly matches with the request with respect to logic-based equivalence of their formal semantics.

**Plug-in match.** Service  $S$  PLUGS INTO request  $R \Leftrightarrow \forall IN_S \exists IN_R: IN_S \succeq IN_R \wedge \forall OUT_R \exists OUT_S: OUT_S \in LSC(OUT_R)$ . Relaxing the exact matching constraint, service  $S$  may require less input than it has been specified in the request  $R$ . This guarantees at a minimum that  $S$  will be executable with the provided input iff the involved OWL input concepts can be equivalently mapped to WSDL input messages and corresponding service signature data types. We assume this as a necessary constraint of each of the subsequent filters.

In addition,  $S$  is expected to return more specific output data whose logically defined semantics is exactly the same or very close to what has been requested by the user. This kind of match is borrowed from the software engineering domain, where software components are considered to plug-in match with each other as defined above but not restricting the output concepts to be direct children of those of the query.

**Subsumes match.** Request  $R$  SUBSUMES service  $S \Leftrightarrow \forall IN_S \exists IN_R: IN_S \succeq IN_R \wedge \forall OUT_R \exists OUT_S: OUT_R \succeq OUT_S$ . This filter is weaker than the plug-in filter with respect to the extent the returned output is more specific than requested by the user, since it relaxes the constraint of immediate output concept subsumption. As a consequence, the returned set of relevant services is extended in principle.

**Subsumed-by match.** Request  $R$  is SUBSUMED BY service  $S \Leftrightarrow \forall IN_S \exists IN_R: IN_S \supseteq IN_R \wedge \forall OUT_R \exists OUT_S: (OUT_S \doteq OUT_R \vee OUT_S \in LGC(OUT_R)) \wedge SIM_{IR}(S, R) \geq \alpha$ . This filter selects services whose output data is more general than requested, hence, in this sense, subsumes the request. We focus on direct parent output concepts to avoid selecting services returning data which we think may be too general. Of course, it depends on the individual perspective taken by the user, the application domain, and the granularity of the underlying ontology at hand, whether a relaxation of this constraint is appropriate, or not.

**Logic-based fail.** Service  $S$  fails to match with request  $R$  according to the above logic-based semantic filter criteria.

**Nearest-neighbor match.** Service  $S$  is NEAREST NEIGHBOR of request  $R \Leftrightarrow \forall IN_S \exists IN_R: IN_S \supseteq IN_R \wedge \forall OUT_R \exists OUT_S: OUT_R \supseteq OUT_S \vee SIM_{IR}(S, R) \geq \alpha$ .

**Fail.** Service  $S$  does not match with request  $R$  according to any of the above filters.

The OWLS-MX matching filters are sorted according to the size of results they would return, in other words according to how relaxed the semantic matching. In this respect, we assume that service output data that are more general than requested relaxes a semantic match with a given query. As a consequence, we obtain the following total order of matching filters

EXACT < PLUG-IN < SUBSUMES < SUBSUMED-BY <  
LOGIC-BASED FAIL < NEAREST-NEIGHBOR < FAIL.

### 3.2 Generic OWLS-MX matching algorithm

The OWLS-MX matchmaker takes any OWL-S service as a query, and returns an ordered set of relevant services that match the query each of which annotated with its individual degree of matching, and syntactic similarity value. The user can specify the desired degree, and syntactic similarity threshold. OWLS-MX then first classifies the service query I/O concepts into its local service I/O concept ontology. For this purpose, it is assumed that the type of computed terminological subsumption relation determines the degree of semantic relation between pairs of input and concepts.

Auxiliary information on whether an individual concept is used as an input or output concept by any registered service is attached to this concept in the ontology. The respective lists of service identifiers are used by the matchmaker to compute the set of relevant services that I/O match the given query according to its five filters.

In particular, OWLS-MX does not only pairwise determine the degree of logical match but syntactic similarity between the conjunctive I/O concept expressions in OWL-Lite. These expressions are built by recursively unfolding each query and service input (output) concept in the local matchmaker ontology. As a result, the unfolded concept expressions are including primitive components of a basic shared vocabulary only. Any failure of logical concept subsumption produced by the integrated description logic reasoner of OWLS-MX will be tolerated, if and only if the degree of syntactic similarity between the respective unfolded service and request concept expressions exceeds a given similarity threshold.

The pseudo-code of the generic OWLS-MX matching process is given below (cf. algorithms 1 - 3). Let  $INPUTS_S = \{ IN_{S,i} | 0 \leq i \leq s \}$ ,  $INPUTS_R = \{ IN_{R,j} | 0 \leq j \leq n \}$ ,  $OUTPUTS_S = \{ OUT_{S,k} | 0 \leq k \leq r \}$ ,  $OUTPUTS_R = \{ OUT_{R,t} | 0 \leq t \leq m \}$ , set of input and output concepts used in the profile I/O parameters HASINPUT and HASOUTPUT of registered service  $S$  in the set *Advertisements*, and the service request  $R$ , respectively. Attached to each concept in the matchmaker ontology are auxiliary data that informs about which registered service is using this concept as an input and/or output concept.

**Algorithm 1 Match:** Find advertised services  $S$  that best hybridly match with a given request  $R$ ; returns set of  $(S, degreeOfMatch, SIM_{IR}(R, S))$  with maximum degree of match (*dom*) unequal FAIL (uses algs. 2 and 3 to compute *dom*), and syntactic similarity value exceeding a given threshold  $\alpha$ .

---

```

1: function MATCH(Request  $R$ ,  $\alpha$ )
2:   local  $result, degreeOfMatch, hybridFilters = \{$ 
      SUBSUMED-BY, NEAREST NEIGHBOUR  $\}$ 
3:   for all  $(S, dom) \in CANDIDATES_{inputset}(INPUTS_R) \wedge$ 
       $(S, dom') \in CANDIDATES_{outputset}(OUTPUTS_R)$  do
4:      $degreeOfMatch \leftarrow \min(dom, dom')$ 
5:     if  $degreeOfMatch \geq minDegree \wedge ($ 
       $degreeOfMatch \notin hybridFilters \vee$ 
       $SIM_{IR}(R, S) \geq \alpha)$  then
6:        $result := result \cup \{ (S, degreeOfMatch,$ 
       $SIM_{IR}(R, S)) \}$ 
7:     end if
8:   end for
9:   return  $result$ 
10: end function

```

---

In the following section, we present five variants of this generic OWLS-MX matchmaking scheme.

### 3.3 OWLS-MX variants

We implemented different variants of the generic OWLS-MX algorithm, called OWLS-M1 to OWLS-M4, each of which uses the same logic-based semantic filters but different IR similarity metric  $SIM_{IR}(R, S)$  for content-based service I/O matching. The variant OWLS-MO performs logic based only semantic service I/O matching.

**OWLS-M0.** The logic-based semantic filters EXACT, PLUG-IN, and SUBSUMES are applied as defined in section 3.1, whereas the hybrid filter SUBSUMED-BY is utilized without checking the syntactic similarity constraint.

**OWLS-M1 to OWLS-M4.** The hybrid semantic matchmaker variants OWLS-M1, OWLS-M3, and OWLS-M4 compute the syntactic similarity value  $SIM_{IR}(OUT_S, OUT_R)$  by use of the loss-of-information measure, extended Jacquard similarity coefficient, the cosine similarity value, and the Jensen-Shannon information divergence based similarity value, respectively.

Based on the experimental results of measuring the performance of similarity metrics for text information retrieval provided by Cohen and his colleagues [4], we selected the top performing ones to build the OWLS-MX variants. These symmetric token-based string similarity measures are defined as follows.

---

**Algorithm 2** Find services which *input* matches with that of the request; returns set of  $(S, dom)$  with minimum degree of match *dom* unequal FAIL.

---

```

1: function CANDIDATESinputset(INPUTSR)
2:   local  $H, dom, r$ 
3:   ▷ If a service input matches with multiple request
      inputs the best degree is returned
4:    $H := \{ (S, IN_{S,i}, dom) \in \bigcup_{j=1..n}$ 
      CANDIDATESinput(INR,j) |  $dom = \mathit{argmax}_l \{$ 
       $(S, IN_{S,i}, dom_i) \mid 1 \leq l \leq n, 1 \leq i \leq s \}$ 
5:   ▷ If all inputs of service  $S$  are matched by those of
      the request,  $S$  can be executed, and the minimum
      degree of its potential match is returned
6:   for all  $S \in \mathit{Advertisement}$  do
7:     if  $\{ (S, IN_{S_1}, dom_1), \dots, (S, IN_{S_s}, dom_s) \} \subseteq H$ 
      then
8:        $r := r \cup \{ (S, \mathit{MIN}(dom_1, \dots, dom_s)) \}$ 
9:     end if
10:  end for
11:  ▷ Services with no input can always be exe-
      cuted and are preliminary EXACT-match can-
      didates:  $\mathit{SERVNoIN}() = \{ (S, \mathit{EXACT}) \mid S \in$ 
       $\mathit{Advertisements} \wedge \mathit{INPUTS}_S = \emptyset \}$ 
12:  ▷ Remaining, unmatched services are at least
      NEAREST NEIGHBOUR-match candidates:  $\mathit{REMSERV}() = \{ (S, \mathit{NEAREST NEIGHBOUR}) \mid S \in$ 
       $\mathit{Advertisements} \wedge \langle S, \mathit{degreeOfMatch}' \rangle \notin r \}$ 
13:  return  $r := r \cup \mathit{SERVNoIN}() \cup \mathit{REMSERV}()$ 
14: end function
15:
16: function CANDIDATESinput(INR,j) ▷
      Classify request input concept into ontology, and use
      the auxiliary concept data to collect services that at
      least plug-in match with respect to its input.
17:   local  $r$ 
18:    $r := r \cup \{ (S, IN_S, \mathit{EXACT}) \mid S \in \mathit{Advertisements},$ 
       $IN_S \in \mathit{input}_{SS}, IN_S \doteq IN_{R,j}, \}$ 
19:    $r := r \cup \{ (S, IN_S, \mathit{PLUG-IN}) \mid S \in \mathit{Advertisements},$ 
       $IN_S \in \mathit{input}_{SS}, IN_S \geq IN_{R,j}, \}$ 
20:   return  $r$ 
21: end function

```

---



---

**Algorithm 3** Find services which *output* matches with that of the request; returns set of  $(S, dom)$  with minimum degree of match unequal FAIL.

---

```

1: function CANDIDATESoutputset(OUTPUTSR)
2:   local  $r, dom$ 
3:   if OUTPUTSR =  $\emptyset$  then
4:     return  $\{ (S, \mathit{EXACT}) \mid S \in \mathit{Advertisements} \}$ 
5:   end if
6:   for all  $S \in \mathit{Advertisements}$  do
7:     if  $(S, dom_t) \in \mathit{CANDIDATES}_{output}(\mathit{OUT}_{R,t}) \wedge$ 
       $dom_t \geq \mathit{SUBSUMES}$  for  $t = 1..m$  then
8:        $r := r \cup \{ (S, \mathit{MIN}\{dom_1, \dots, dom_m\}) \}$ 
9:     else if  $(S, dom_t) \in \mathit{CANDIDATES}_{output}(\mathit{OUT}_{R,t})$ 
       $\wedge dom_t \in \{ \mathit{EXACT}, \mathit{SUBSUMES} \}$  for  $t = 1..m$ 
      then
10:       $r := r \cup \{ (S, \mathit{SUBSUMED-BY}) \}$ 
11:    end if
12:  end for
13:  ▷ Any remaining, unmatched service is a potential
      NEAREST NEIGHBOUR-match:  $\mathit{REMSERV}() = \{ (S,$ 
       $\mathit{NEAREST NEIGHBOUR}) \mid S \in \mathit{Advertisements} \wedge$ 
       $S \notin r \}$ 
14:  return  $r := r \cup \mathit{REMSERV}()$ 
15: end function
16:
17: function CANDIDATESoutput(OUTR,t) ▷ Classify request
      output concept into ontology, and use the auxiliary
      concept data to collect services with output concepts
      that match with OUTR,t.
18:   local  $r$ 
19:    $r := r \cup \{ (S, \mathit{EXACT}) \mid \mathit{OUT}_S \doteq \mathit{OUT}_{R,t} \}$ 
20:    $r := r \cup \{ (S, \mathit{PLUG-IN}) \mid \mathit{OUT}_S \in \mathit{LSC}(\mathit{OUT}_{R,t}) \wedge S$ 
       $\notin r \}$ 
21:    $r := r \cup \{ (S, \mathit{SUBSUMES}) \mid \mathit{OUT}_S \leq \mathit{OUT}_{R,t} \wedge S \notin r$ 
       $\}$ 
22:    $r := r \cup \{ (S, \mathit{SUBSUMED-BY}) \mid \mathit{OUT}_S \in \mathit{LGC}(\mathit{OUT}_{R,t})$ 
       $\}$ 
23:   return  $r$ 
24: end function

```

---

- The *cosine* similarity metric

$$Sim_{Cos}(S, R) = \frac{\vec{R} \cdot \vec{S}}{\|\vec{R}\|_2 \cdot \|\vec{S}\|_2} \quad (1)$$

with standard TFIDF term weighting scheme, and the unfolded concept expressions of request  $R$  and service  $S$  are represented as  $n$ -dimensional weighted index term vectors  $\vec{R}$  and  $\vec{S}$  respectively.  $\vec{R} \cdot \vec{S} = \sum_{i=1}^n w_{i,R} \times w_{i,S}$ ,  $\|X\|_2 = \sqrt{\sum_i w_{i,X}^2}$ , and  $w_{i,X}$  denotes the weight of the  $i$ -th index term in vector  $X$ .

- The *extended Jacquard* similarity metric  $Sim_{EJ}(S, R) =$

$$\frac{\vec{R} \cdot \vec{S}}{\|\vec{R}\|_2 + \|\vec{S}\|_2 - \vec{R} \cdot \vec{S}} \quad (2)$$

with standard TFIDF term weighting scheme.

- The *intensional loss of information* based similarity metric  $Sim_{LOI}(S, R) =$

$$1 - \frac{LOI_{IN}(R, S) + LOI_{OUT}(R, S)}{2} \quad (3)$$

$$LOI_x(R, S) = \frac{|PC_{R,x} \cup PC_{S,x}| - |PC_{R,x} \cap PC_{S,x}|}{|PC_{R,x}| + |PC_{S,x}|} \quad (4)$$

with  $x \in \{IN, OUT\}$ ,  $PC_{R,x}$  and  $PC_{S,x}$  set of primitive components in unfolded logical input/output concept expression of request  $R$  and service  $S$

- The *Jensen-Shannon information divergence* based similarity measure  $Sim_{JS}(S, R) = \log 2 - JS(S, R) =$

$$\frac{1}{2 \log 2} \sum_{i=1}^n h(p_{i,R}) + h(p_{i,S}) - h(p_{i,R} + p_{i,S}) \quad (5)$$

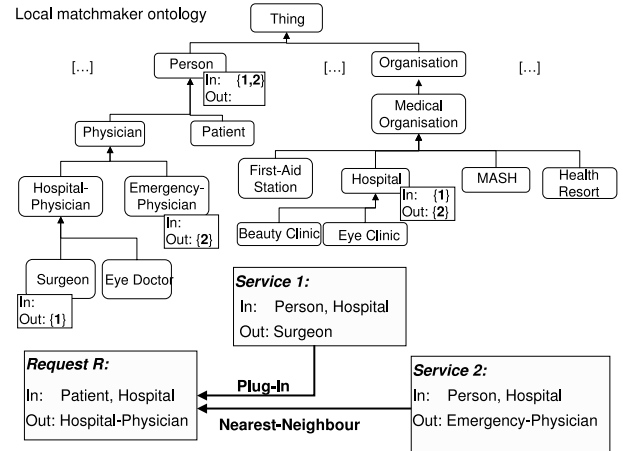
with probability term frequency weighing scheme, *e.g.*,  $p_{i,R}$  denotes the probability of  $i$ -th index term occurrence in request  $R$ , and  $h(x) = -x \log x$ ,

The extended Jacquard metric is a standard for measuring the degree of overlap as the ratio of the number of shared terms (primitive components) of unfolded concepts of both service and request, and the number of terms possessed by either of them. In contrast to the TFIDF/cosine similarity metric, it does not favor the document with common terms. The Jensen-Shannon measure is based on the information-theoretic, non-symmetrical Kullback-Leibler divergence measure. It measures the pairwise dissimilarity of conditional probability term distributions between service and request text rather than looking at the whole collection as it is the case for the TFIDF/cosine, or the extended Jacquard metric. The loss of (intensional) information in case some concept  $A$  is terminologically substituted by concept  $B$ , can be measured as the inverse ratio of the number of matching primitive components with those which remain unmatched in terminologically disjoint unfolded concept constraints. The symmetric LOI-based similarity value for a given pair of service and request is then computed analogously for all I/O concept definitions involved.

### 3.4 Example

Let us illustrate the hybrid service retrieval with OWLS-MX by means of a simple example. Suppose the concept

subsumption hierarchy or taxonomy of the OWLS-MX matchmaker ontology, the service request  $R$  for physicians of hospital  $h$  that provide treatment to patient  $p$ , and relevant service advertisements  $S_1$  and  $S_2$  are as shown in figure 2.



**Figure 2: Example of hybrid service matching with OWLS-MX**

Service  $S_1$  is considered semantically relevant to request  $R$ , since it returns for any given person  $p$  and hospital  $h$ , the individual surgeon of  $h$  that operated on  $p$ . Likewise, service  $S_2$  is relevant to  $R$ , since it returns those emergency physicians who provided emergency treatment to  $p$  before her transport to hospital  $h$ . Hence, both services  $S_1$  and  $S_2$  should be returned as matching results to the user.

However, the logic based only variant OWLS-M0 determines  $S_1$  as plug-in matching with  $R$  but fails to return  $S_2$ , since the formal semantics of the output concept siblings "emergency physician" and "hospital physician" in the ontology are terminologically disjoint. In this example, the set of terminological constraints of unfolded concepts  $c$  correspond to the set of primitive components ( $c^p$ ) of which the individual concepts are canonically defined in the matchmaker ontology  $T$ . Hence, the unfolded concept expressions are as follows.

- $unfolded(Patient, T) = (and Patient^p Person^p)$
- $unfolded(Hospital, T) = (and Hospital^p (and MedicalOrganisation^p Organisation^p))$
- $unfolded(HospitalPhysician, T) = (and HospitalPhysician^p (and Physician^p Person^p))$
- $unfolded(Surgeon, T) = (and Surgeon^p (and HospitalPhysician^p (and Physician^p Person^p)))$
- $unfolded(EmergencyPhysician, T) = (and EmergencyPhysician^p (and Physician^p Person^p))$

As a result, for example, OWLS-M1 would return  $S_1$  as plug-in matching service with syntactic similarity value of  $Sim_{LOI}(R, S_1) = 0.87$ . In contrast to OWLS-MO, it also returns  $S_2$ , since this service is nearest-neighbor matching with the request  $R$ : Their implicit semantics exploited by the IR similarity metric LOI (cf. (5), (6)) with  $Sim_{LOI}$

$(R, S_2) = \frac{(1 - \frac{5-4}{5+4}) + (1 - \frac{4-2}{3+3})}{2} = 0.78 \geq \alpha = 0.7$  is sufficiently similar. Our preliminary experimental results show that this kind of matching relaxation may be useful in practice.

## 4. IMPLEMENTATION

We implemented the OWLS-MX matchmaker variants version 1.1 in Java using the OWL-S API 1.1 beta with the tableaux OWL-DL reasoner Pellet developed at university of Maryland (cf. <http://www.mindswap.org>). As the OWL-S API is tightly coupled with the Jena Semantic Web Framework, developed by the HP Labs Semantic Web research group (cf. <http://jena.sourceforge.net/>), the latter is also used to modify the OWLS-MX matchmaker ontology.

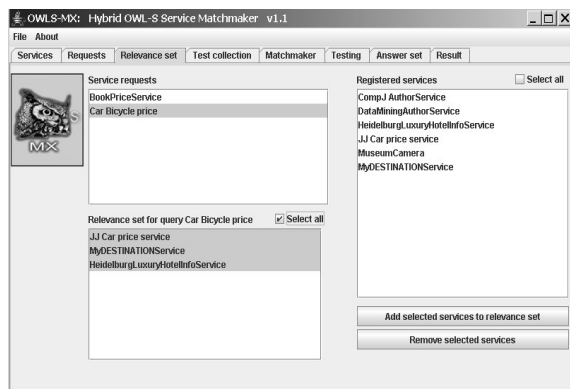


Figure 3: OWLS-MX v1.1 screenshot: Definition of service request and relevance set

Figures 3 to 5 show some screenshots of the OWLS-MX version 1.1 graphical user interface.

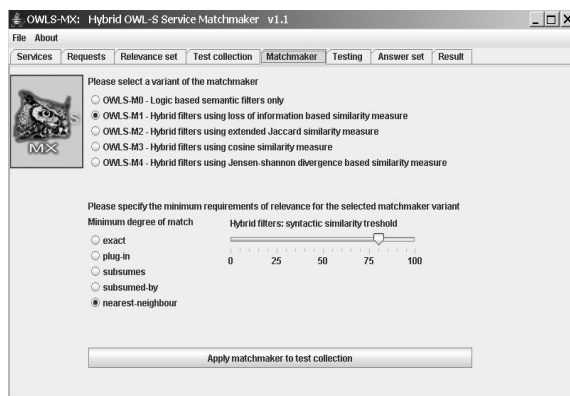


Figure 4: OWLS-MX v1.1 screenshot: Selection of OWLS-MX variant

After parsing service advertisements and requests, the respective input and output concepts are analysed and, if necessary, added to the local matchmaker ontology together with auxiliary data on their unfolding. As a consequence, the matchmaker ontology is dynamically built and growing with the number of services and underlying ontologies loaded. In addition, the matchmaker ontology is extended with auxiliary information for each concept whether it is used as an input or output concept of which service registered at the matchmaker. Service requests are treated

similarly, except that they are not stored in the extended matchmaker ontology.

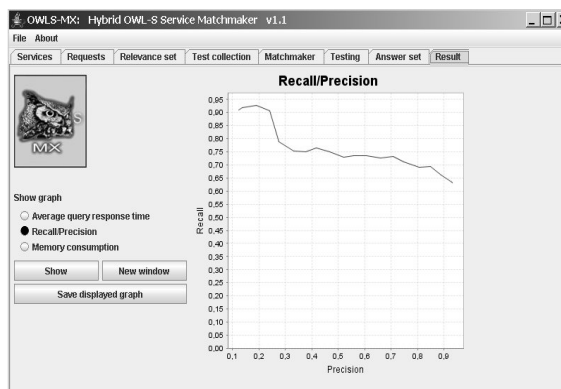


Figure 5: OWLS-MX v1.1 screenshot: Display of selected type of results (performance)

For each service request concept, the service identifiers attached to its immediate parent and child concepts of the enhanced matchmaker ontology are retrieved. The semantic degree of matching for each service is then determined by applying the semantic filters on this set of matching candidates. After this step, the syntactic similarity is computed by applying the selected IR similarity metric to the strings of unfolded concepts of the query and each registered service. Both the semantic degree of match and the syntactic similarity value determine the hybrid degree of matching of one service with the request. If this hybrid degree is better than or equal to the minimum degree specified by the user, then this service will be returned as potentially relevant.

In practice, OWLS-MX spend the largest amount of time with classifying the ontologies used by the registered services to check for new concepts not known to the matchmaker, and then to classify them into the matchmaker ontology.

## 5. EXPERIMENTAL EVALUATION

For measuring the service I/O retrieval performance of each OWLS-MX variant we used the OWL-S service retrieval test collection OWLS-TC v2. This collection consists of more than 570 services specified in OWL-S 1.1 covering seven application domains, that are education, medical care, food, travel, communication, economy, and weaponry. The majority of these services were retrieved from public IBM UDDI registries, and semi-automatically transformed from WSDL to OWL-S. OWLS-TC v2 provides a set of 28 test queries each of which is associated with a set of 10 to 20 services that two of the co-authors subjectively defined as relevant according to the standard TREC definition of binary relevance [17]<sup>1</sup>. The collection OWLS-TC v2 is available as open source at

<http://projects.semwebcentral.org/projects/owl-s-tc/>.

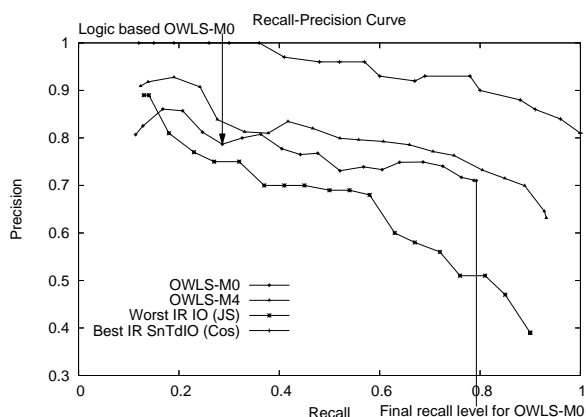
In terms of measuring the retrieval performance of each OWLS-MX variant, we adopted the evaluation strategy of

<sup>1</sup>Please note, that no standardized test collection for OWL-S service retrieval does exist yet. Therefore, like with any other reported results on retrieval performance of alternative OWL-S service matchmakers developed by different research groups world wide, we have to consider both our test collection and experimental results as preliminary.

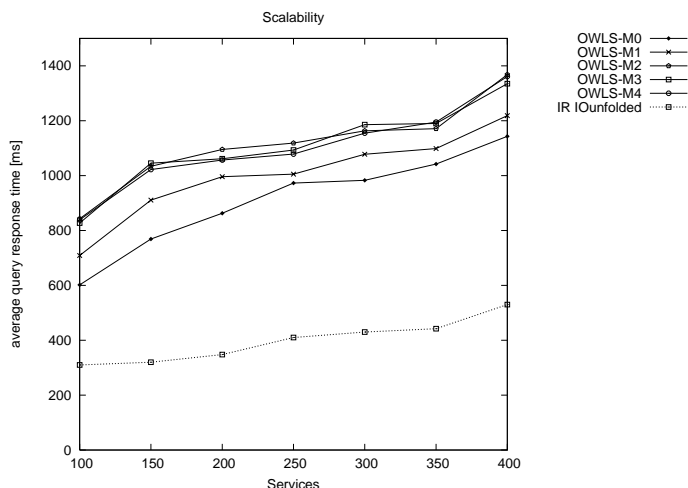
micro-averaging the individual precision-recall curves [18]. Let  $Q$  be the set of test queries (service requests) in OWLS-TC,  $A$  the sum of relevant documents of all requests in  $Q$ ,  $A_R$  the answer set of relevant services (service advertisements) for request  $R \in Q$ . For each request  $R$ , we consider  $\lambda = 20$  steps up to its maximum recall value, and measure the number  $|B_{\lambda R}|$  of relevant documents retrieved (recall) at each of these steps. Similarly, we measure related precision with the number  $B_\lambda$  of retrieved documents at each step  $\lambda$ . The micro-averaging of recall and precision (at step  $\lambda$ ) over all requests, as we used it for performance evaluation is then defined as

$$Rec_\lambda = \sum_{R \in Q} \frac{|A_R \cap B_{\lambda R}|}{|A|}, Prec_\lambda = \sum_{R \in Q} \frac{|A_R \cap B_{\lambda R}|}{|B_\lambda|} \quad (6)$$

The micro-averaged R-P curves of the top and worse performing IR similarity metric together with those for the OWLS-MX variants as well as the average query response time plots are displayed in figures 4 and 5, respectively.



**Figure 6: Recall-precision performance of logic based OWLS-M0 vs best hybrid OWLS-M4 vs IR based service I/O matching**



**Figure 7: Average query response time of OWLS-MX vs IR based service matching**

These experimental results provide, in particular, evidence in favor of following conclusions.

- The best IR similarity metric (Cosine/TFIDF) applied to the concatenated unfolded service profile I/O concept expressions performs close to the pure logic based OWLS-M0 (see figure 6: Best IR SnTdlO (Cos) vs. OWLS-M0). But OWLS-M0 is only superior to IR based matching (cf. figure 6: Worst IR IO (JS) denoting Jensen-Shannon divergence based similarity metric) at the very expense of its recall. However, for discovering semantic web services, precision may be more important to the user than recall, since the set of relevant services is supposed to be subject to continuous change in the semantic Web in practice.
- Pure logic based semantic matching by OWLS-M0 can be outperformed by hybrid semantic matching, in terms of both recall and precision. That is the case, for example, by use of the best performing hybrid matchmaker OWLS-M4 (cf. figure 6). The main reason for this is, that the additional IR based similarity check of the nearest-neighbor filter allows OWLS-M1 to M4 to find relevant services that OWLS-M0 would fail to retrieve.
- Hybrid semantic matching by OWLS-MX can be outperformed by each of the selected syntactic IR similarity metrics to the extent additional parameters with natural language text content are used. That is the case, for example, by applying the cosine similarity metric to the extended set of service profile parameters including not only hasInput and hasOutput but also serviceName and textDescription (cf. figure 6).
- Both pure logic based and all hybrid OWLS-MX matchmakers are significantly outrun by IR based service retrieval in terms of average query response time almost by size of magnitude (cf. figure 7). This is due to the additional computational efforts required by OWLS-MX to determine concept subsumption relationships in NEXPTIME description logic OWL-DL based on the imported large ontologies the OWL-S services refer to.

## 6. RELATED WORK

Quite a few semantic Web service matchmakers have been developed in the past couple of years such as the OWLS-UDDI matchmaker [16], RACER [11], SDS [12], MAMA [5], HotBlu [6], and [10]. Like OWLS-MX, the majority of them does perform profile based service signature (I/O) matching. Alternate approaches propose service process-model matching [3], recursive tree matching [2], P2P discovery [1], automated selection of WSMO services [20] and METEOR-S for WSDL-S services [19]. Except LARKS [15], none of them is hybrid, in the sense that it exploits both explicit and implicit semantics by complementary means of logic based and approximate matching. To the best of our knowledge, OWLS-MX is the only hybrid matchmaker for OWL-S services yet.

The OWLS-MX matchmaker bases on LARKS [15]. However, LARKS differs from OWLS-MX in that it uses a proprietary capability description language and description logic different from OWL-S and OWL-DL, respectively. Furthermore, LARKS does not perform any subsumes and subsumed-



by nor nearest-neighbour matching, and has not been experimentally evaluated yet.

The purely logic based variant OWLS-M0 of OWLS-MX is quite similar to the OWLS-UDDI matchmaker [16] but differs from it in several aspects. Firstly, the latter makes use of a different notion of plug-in matching, and does not perform additional subsumed-by matching. Secondly, OWLS-M0 classifies arbitrary query concepts into its dynamically evolving ontology with commonly shared minimal basic vocabulary of primitive components instead of limiting query I/O concepts to terminologically equivalent service I/O concepts in a shared static ontology as the OWLS-UDDI matchmaker does.

## 7. CONCLUSIONS

Our approach to hybrid semantic Web service matching, called OWLS-MX, utilizes both logic based reasoning and IR techniques for semantic Web services in OWL-S. Experimental evaluation results provide evidence in favor of the proposition that building semantic Web service matchmakers *purely* on description logic reasoners may be insufficient, hence should give a clear impetus for further studies, research and development of more powerful approaches to service matching in the semantic Web across disciplines.

## 8. REFERENCES

- [1] F. Banaei-Kashani, C.-C. Chen, and C. Shahabi. Wspds: Web services peer-to-peer discovery service. In *Proceedings of International Symposium on Web Services and Applications (ISWS)*, 2004.
- [2] S. Bansal and J. Vidal. Matchmaking of web services based on the daml-s service model. In *Proceedings of Second International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, Melbourne, Australia, 2003.
- [3] A. Bernstein and M. Klein. Towards high-precision service retrieval. In *IEEE Internet Computing*, 8(1):30-36, 2004.
- [4] W. Cohen, P. Ravikumar, and S. Fienberg. A comparison of string distance metrics for name-matching tasks. In *Proc. IJCAI-03 Workshop on Information Integration on the Web (IIWeb-03)*. DBLP at <http://dblp.uni-trier.de>, 2003.
- [5] S. Colucci, T. D. Noia, E. D. Sciascio, F. Donini, and M. Mongiello. Concept abduction and contraction for semantic-based discovery of matches and negotiation spaces in an e-marketplace. In *Proc. 6th Int Conference on Electronic Commerce (ICEC 2004)*. ACM Press, 2004.
- [6] I. Constantinescu and B. Faltings. Efficient matchmaking and directory services. In *Proceedings of IEEE/WIC International Conference on Web Intelligence*, 2003.
- [7] T. Grabs and H.-J. Schek. Flexible information retrieval on xml documents. In *Intelligent Search on XML Data, Applications, Languages, Models, Implementations, and Benchmarks*. Springer, 2003.
- [8] I. Horrocks, P. Patel-Schneider, and F. van Harmelen. From shiq and rdf to owl: The making of a web ontology language. *Web Semantics*, 1(1), Elsevier, 2004.
- [9] U. Keller, R. Lara, A. Polleres, and D. Fensel. Automatic location of services. In *Proceedings of European Semantic Web Conference (ESWC)*, Springer, LNAI 3532, 2005.
- [10] M. Klein and B. Koenig-Ries. Coupled signature and specification matching for automatic service binding. In *Proceedings of European Conference on Web Services*, Springer, LNAI, 183-197, 2004.
- [11] L. Li and I. Horrocks. A software framework for matchmaking based on semantic web technology. In *Proc. 12th Int World Wide Web Conference Workshop on E-Services and the Semantic Web (ESSW 2003)*, 2003.
- [12] D. Mandell and S. McIlraith. A bottom-up approach to automating web service discovery, customization, and semantic translation. In *Proc. 12th Int Conference on the World Wide Web (WWW 2003)*. ACM Press, 2003.
- [13] OWL-S. Semantic markup for web services; w3c member submission 22 november 2004. <http://www.w3.org/Submission/2004/SUBM-OWL-S-20041122/>.
- [14] A. Sheth, C. Ramakrishnan, and C. Thomas. Semantics for the semantic web: The implicit, the formal, and the powerful. *Semantic Web and Information Systems*, 1(1), Idea Group, 2005.
- [15] K. Sycara, M. Klusch, S. Widoff, and J. Lu. Larks: Dynamic matchmaking among heterogeneous software agents in cyberspace. *Autonomous Agents and Multi-Agent Systems*, 5(2), Kluwer, 2002.
- [16] K. Sycara, M. Paolucci, A. Anolekar, and N. Srinivasan. Automated discovery, interaction and composition of semantic web services. *Web Semantics*, 1(1), Elsevier, 2003.
- [17] TREC. Text retrieval conference. <http://trec.nist.gov/data/>.
- [18] C. van Rijsbergen. *Information Retrieval*. 1979.
- [19] K. Verma, K. Sivashanmugam, A. Sheth, A. Patil, S. Oundhakar, and J. Miller. Meteor-s wsdi: A scalable infrastructure of registries for semantic publication and discovery of web services. *Information Technology and Management*, 2004.
- [20] U. Keller, R. Lara, H. Lausen, A. Polleres, D. Fensel. Automatic Location of Services. In *Proceedings of the 2nd European Semantic Web Conference*, LNCS 3532, 2005.

---

## Hybrid Semantic Matching of WSML Services

F. Kaufer and M. Klusch: WSMO-MX: A Logic Programming Based Hybrid Service Matchmaker. Proceedings of the 4th IEEE European Conference on Web Services (ECOWS), Zurich, Switzerland, pages 162 - 170, IEEE CS Press, 2006.

# WSMO-MX: A Logic Programming Based Hybrid Service Matchmaker

Frank Kaufer

Matthias Klusch

German Research Center for Artificial Intelligence

Deduction and Multi-Agent Systems Lab

Stuhlsatzenhausweg 3, Saarbrücken

E-mail: {frank.kaufer, klusch}@dfki.de

## Abstract

*In this paper, we present an approach to hybrid semantic web service matching based on both logic programming, and syntactic similarity measurement. The implemented matchmaker, called WSMO-MX, applies different matching filters to retrieve WSMO-oriented service descriptions that are semantically relevant to a given query with respect to seven degrees of hybrid matching. These degrees are recursively computed by aggregated valuations of ontology based type matching, logical constraint and relation matching, and syntactic similarity as well.*

## 1 Introduction

The problem of efficiently retrieving relevant services in the envisioned semantic web has been solved so far by only a few approaches for services described in OWL-S [15, 10], and WSML [7, 17]. Though, existing proposals for rule based service mediation in WSMO do not provide a general purpose matchmaking scheme for services in WSML.

This, in particular, motivated us to develop a hybrid semantic matchmaker, called WSMO-MX, that applies different matching filters to retrieve WSMO services that are semantically relevant to a given query including the goal to be satisfied. Both services and goals are described in a Logic Programming (LP) variant of WSML, called WSML-MX, which is based on WSML-Rule. The hybrid matching scheme of WSMO-MX combines the ideas of hybrid semantic matching realized by OWLS-MX [10], the object-oriented structure based matching proposed by Klein & König-Ries [9], and the concept of intentional matching introduced by Keller et. al [6].

The remainder of this paper is structured as follows. In section 2, we introduce WSML-MX, align it with WSML-Rule, and describe the modelling of services in WSML-MX. Section 3 presents the hybrid semantic matching approach of our matchmaker WSMO-MX by means of its dif-

ferent filters of matching, which is then exemplified in section 4. We provide some details of the implementation of WSMO-MX in section 4, and briefly discuss related work and conclude in section 5 and 6, respectively.

## 2 Service modelling with WSML-MX

The web service modelling language WSML is the formal language for the web service modelling ontology (WSMO). However, WSML still is under development, and there is no full-fledged reasoner with WSML parser available yet. Therefore we developed a formally grounded variant of WSML called WSML-MX directly in F-Logic [8, 1]. WSML-MX is similar expressive as WSML-Rule, which has a different (more verbose) syntax as F-Logic but can be mapped into this.

Central to WSML-MX is the notion of *derivative* which is an extended version of the object set introduced by Klein and König-Ries [9]. A derivative  $D_T$  in WSML-MX encapsulates an ordinary concept  $T$  (in this context called type) defined in a given ontology by attaching meta-information merely about the way how  $T$  can be matched with any other type. Such information is defined in terms of different meta-relations of the derivative  $D_T$ . As type  $T$  is defined to be either atomic or a complex type with relations, the derivative  $D_T$  can also have a set of relations different from  $T$ , though this set is empty by default. The structure of a derivative is shown in figure 1. Per naming convention, the identifier of a derivative  $D_T$  of type  $T$  is denoted by  $T.D_n$  such as *Person.D42* for type *Person*.

WSML-MX uses the main and clearly motivated elements required for service matching from WSML, that are *goal*, *service*, *capabilities*, *preconditions*, and *postconditions* but not *effect* and *assumption*. Please note that the formal semantics of capabilities in WSML is still open. Any service in WSML-MX is modelled as a derivative with a relation called capability and a derivative of type capability as range. Pre- and postcondition are relations of the latter derivative both referring to a so called

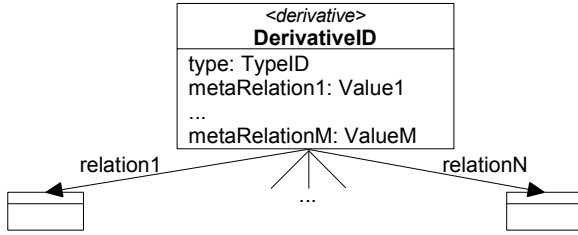


Figure 1. Derivative structure in WSML-MX

state. A state is a set of state parts, which are derivatives each defined as atomic, or as complex by means of relations with derivatives as range. Hence, any service derivative in WSML-MX can be represented as a directed object-oriented graph with derivatives considered as nodes and relations between them as edges, as shown in figure 2.

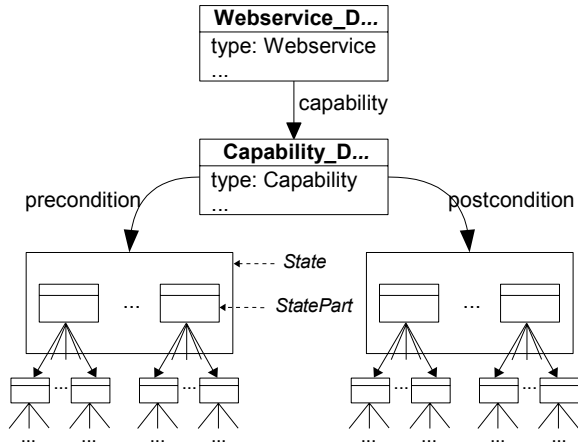


Figure 2. Service derivative in WSML-MX

The language WSML-MX allows for constraints on both relations and derivatives formulated in the full Horn fragment of F-logic. Hence, WSML-MX constraints are as expressive and, in general, only semi-decidable as are WSML-Rule axioms. In WSMO-MX, we use relative query containment for constraint matching (cf. section 3.2.3). However, matching of parts of WSML-MX expressions represented as acyclic object-oriented graphs without constraints is decidable in polynomial time. The emphasis of WSML-MX on these parts of service modelling is motivated not only by clear separation of computationally tractable elements but the fact that it allows the matchmaker for a more detailed explanatory feedback to the user in case the matching of given service and goal derivatives failed.

An example for a service in WSML-MX is shown in figure 3; the service offers tickets for any trip between any two German towns, but if the user departs from Berlin, her destination must be Hamburg.

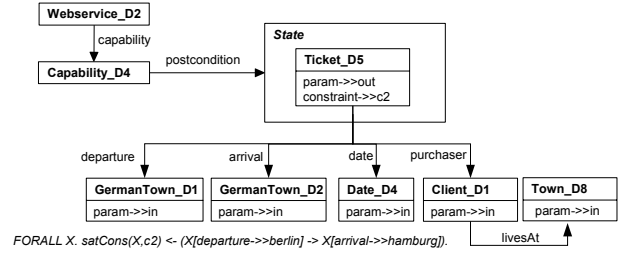


Figure 3. Example service in WSML-MX

### 3 Hybrid matching of derivatives

#### 3.1 Overview

The result of matching a derivative  $D_G$  from a goal description with a derivative  $D_W$  from a service description is a vector  $v \in R^7$  of aggregated valuations of ontology based type matching, logical constraint matching, relation matching, and syntactic matching. Each real-valued entry in the so called valuation vector  $v = (\pi_{\equiv}, \pi_{\sqsubseteq}, \pi_{\sqsupset}, \pi_{\sqcap}, \pi_{\sim}, \pi_{\circ}, \pi_{\perp})$  with  $\pi_i \in [0, 1]$  ( $i \in \{\equiv, \sqsubseteq, \sqsupset, \sqcap, \sim, \circ, \perp\}$ ) and  $\sum \pi_i = 1$ , denotes the extent to which both derivatives  $D_G$  and  $D_W$  match with respect to the hybrid semantic matching degrees  $\pi_i$  of WSMO-MX.

These degrees are the logical relations *equivalence*, *plug-in* known from software component retrieval [18] or the similar *rule of consequences* from Hoare logic [4], *inverse-plug-in*, *intersection* and *disjunction (fail)* as degrees of logic based semantic match. The degree of *fuzzy similarity* refers to a non-logic based semantic match such as syntactic similarity, while the degree *neutral* stands for neither match nor fail, hence declares the tolerance of matching failure. The set-theoretic semantics of the hybrid matching degrees are given in Table 1 based on the relations between the maximum possible instance sets of the derivatives  $D_G$  and  $D_W$ , denoted by  $\mathcal{G}$  and  $\mathcal{W}$ . Since we use the heuristic relative query containment for the constraint matching, these sets are restricted to instances in the matchmaker knowledge base which satisfy the constraints.

In order to compute the degrees of hybrid semantic matching of given goal and service derivative, WSMO-MX recursively applies different matching filters to their preconditions and postconditions, and returns not only the aggregated matching valuation vector but also annotations of the matching process results as a kind of explanatory feedback to the user. That facilitates a more easy iterative goal refinement by the user in case of insufficient matching results. The individual matching filters and their valuation for the degrees of hybrid semantic matching are described in subsequent sections, and exemplified in section 4.

order	symbol	degree of match	pre	post
1	$\equiv$	equivalence		$\mathcal{G} = \mathcal{W}$
2	$\sqsubseteq$	plugin	$\mathcal{G} \subseteq \mathcal{W}$	$\mathcal{W} \subseteq \mathcal{G}$
3	$\supseteq$	inverse-plugin	$\mathcal{G} \supseteq \mathcal{W}$	$\mathcal{W} \supseteq \mathcal{G}$
4	$\sqcap$	intersection		$\mathcal{G} \cap \mathcal{W} \neq \emptyset$
5	$\sim$	fuzzy similarity		$\mathcal{G} \sim \mathcal{W}$
6	$\circ$	neutral	<i>by derivative specific definition</i>	
7	$\perp$	disjunction (fail)		$\mathcal{G} \cap \mathcal{W} = \emptyset$

**Table 1. Degrees of hybrid semantic matching of WSMO service and goal derivatives**

## 3.2 Matching filters

### 3.2.1 Type matching

The matching of types  $T_G$  and  $T_W$  of the goal and service derivative  $D_G$  and  $D_W$  is performed by means of computing the degree of their semantic relation in the matchmaker ontology according to a requested type similarity relation  $TSR$  defined as meta-relation values in  $D_G[typeSimRel \rightarrow TSR]$ . WSMO-MX offers the following derivative type similarity relations (in F-Logic):

- *equivalent*:  $T_W = T_G \vee T_W :: T_G \wedge T_G :: T_W$
- *sub*:  $T_W :: T_G$  ( $T_W$  subtype of  $T_G$ ); *super*:  $T_G :: T_W$
- *sibling*:  $\exists T_P.T_G :: T_P \wedge T_W :: T_P \wedge \neg(\exists T_X.\exists T_Y.T_X \in \{T_G, T_W\} \wedge T_X :: T_Y \wedge T_Y :: T_P)$ ; types with one immediate common ancestor (parent).
- *spouse*:  $\exists T_C.T_C :: T_G \wedge T_C :: T_W \wedge \neg(\exists T_X.\exists T_Y.T_X \in \{T_G, T_W\} \wedge T_C :: T_Y \wedge T_Y :: T_X)$ ; types with one immediate common descendant (child)
- *comAnc* (common ancestor):  $\exists T_P.T_G :: T_P \wedge T_W :: T_P$
- *comDes* (common descendant):  $\exists T_C.T_C :: T_G \wedge T_C :: T_W$
- *relative*: exists a path in the undirected ontology graph between  $T_G$  and  $T_W$

The maximum distance  $TD \in \mathbb{N} \setminus \{0\}$  between types in the matchmaker ontology with respect to which each of the latter three relations gets evaluated to true is specified in the goal derivative in terms of  $D_G[typeDistance \rightarrow TD]$ .  $TD$  is the path length between both types in the undirected ontology graph; for the type relations *comAnc* and *comDes* it must hold that the addition of the path lengths from both derivatives to their nearest common child/parent type is at most  $TD$ . Optionally, the same restriction can be imposed on the type relations *sub* and *super* with  $TD$  greater or equal the path length from  $D_G$  to  $D_W$ .

The valuation of the type matching of  $D_G$  and  $D_W$  for each of the hybrid semantic matching degrees of WSMO-MX is listed in Table 2. If more than one type similarity relation  $TSR$  is specified in the goal, the maximum of the valuation vectors is selected as a result.

### 3.2.2 Relation matching

Given that the  $D_G$  and  $D_W$  are complex, the hybrid semantic matching must continue recursively with comparing their relations. Let the relation signatures of  $D_G$  and  $D_W$  be defined as follows:  $D_G[R_1 \Rightarrow E_1; \dots; R_k \Rightarrow E_k; S_1 \Rightarrow F_1; \dots; S_l \Rightarrow F_l; \dots; S_m \Rightarrow F_m]$ , and  $D_W[R_1 \Rightarrow G_1; \dots; R_k \Rightarrow G_k; T_1 \Rightarrow H_1; \dots; T_n \Rightarrow H_n]$ , where  $R_1, \dots, R_k, S_1, \dots, S_m, T_1, \dots, T_n$  are unique relation names with  $\bigcup_{i \in [1, m]} S_i \cap \bigcup_{j \in [1, n]} T_j = \emptyset$  and derivatives  $E_1, \dots, E_k, G_1, \dots, G_k, F_1, \dots, F_m, H_1, \dots, H_n$  the respective ranges of the relations.

The relations  $R_1, \dots, R_k$  of the goal derivative  $D_G$  for which equally named relations do exist in  $D_W$  are valued for the hybrid degree of matching by recursively matching their ranges with each other. That is, WSMO-MX attempts to match the (goal) derivatives  $E_\tau$  with the (service) derivatives  $G_\tau$  for all  $\tau \in [1, k]$  and compute the respective valuation vectors.

We assume that for all relations  $S_\mu, \mu \in [1, l]$  in  $D_G$  that cannot be paired with an equally named relation in  $D_W$  (under unique name assumption for shared namespaces) there exist one so called *missing strategy* which indicates the matchmaker how to cope with this problem. Such a missing relation strategy is specified in the goal in terms of  $D_G[missingStrat@(S_\mu) \rightarrow MS_\mu]$ , with  $MS_\mu \in \{assumeEquivalent, assumeFailed, ignore\}$ .

The valuations for relations with missing strategies are given in table 3. It lists also the valuations for the relations without missing strategy ( $S_1, \dots, S_m$  and  $T_1, \dots, T_n$ ), which depend on whether they are part of a pre- or postcondition.

The final valuation vector for the recursive relation matching between  $D_G$  and  $D_W$  is an aggregation of all valuation vectors computed for the missing relations, and those for the relation range derivative matchings. The cor-

type similarity relation	valuation vector	
	$val_{pre}$	$val_{post}$
	( $\pi_{\equiv}, \pi_{\sqsubseteq}, \pi_{\sqsupset}, \pi_{\sqcap}, \pi_{\sim}, \pi_{\circ}, \pi_{\perp}$ )	( $\pi_{\equiv}, \pi_{\sqsubseteq}, \pi_{\sqsupset}, \pi_{\sqcap}, \pi_{\sim}, \pi_{\circ}, \pi_{\perp}$ )
<i>equivalent</i>	( 1, 0, 0, 0, 0, 0, 0 )	( 1, 0, 0, 0, 0, 0, 0 )
<i>sub</i>	( 0, 0, 1, 0, 0, 0, 0 )	( 0, 1, 0, 0, 0, 0, 0 )
<i>super</i>	( 0, 1, 0, 0, 0, 0, 0 )	( 0, 0, 1, 0, 0, 0, 0 )
<i>sibling</i>	( 0, 0, 0, 1, 0, 0, 0 )	( 0, 0, 0, 1, 0, 0, 0 )
<i>comAnc</i>	( 0, 0, 0, 1, 0, 0, 0 )	( 0, 0, 0, 1, 0, 0, 0 )
<i>spouse</i>	( 0, 0, 0, 0, 1, 0, 0 )	( 0, 0, 0, 0, 1, 0, 0 )
<i>comDes</i>	( 0, 0, 0, 0, 1, 0, 0 )	( 0, 0, 0, 0, 1, 0, 0 )
<i>relative</i>	( 0, 0, 0, 0, 1, 0, 0 )	( 0, 0, 0, 0, 1, 0, 0 )

Table 2. Valuation of type matching for hybrid matching degrees

missing strategy	valuation vector	
	$val_{pre,webservice}/val_{post,goal}$	$val_{post,webservice}/val_{pre,goal}$
	( $\pi_{\equiv}, \pi_{\sqsubseteq}, \pi_{\sqsupset}, \pi_{\sqcap}, \pi_{\sim}, \pi_{\circ}, \pi_{\perp}$ )	( $\pi_{\equiv}, \pi_{\sqsubseteq}, \pi_{\sqsupset}, \pi_{\sqcap}, \pi_{\sim}, \pi_{\circ}, \pi_{\perp}$ )
<i>assumeEquivalent</i>	( 1, 0, 0, 0, 0, 0, 0 )	( 1, 0, 0, 0, 0, 0, 0 )
<i>none</i>	( 0, 1, 0, 0, 0, 0, 0 )	( 0, 0, 1, 0, 0, 0, 0 )
<i>ignore</i>	( 0, 0, 0, 0, 0, 1, 0 )	( 0, 0, 0, 0, 0, 1, 0 )
<i>assumeFailed</i>	( 0, 0, 0, 0, 0, 0, 1 )	( 0, 0, 0, 0, 0, 0, 1 )

Table 3. Valuation of relation matching with missing strategies for hybrid matching degrees

responding relation matching algorithm is outlined in the subsequent section (cf. algorithm 5).

### 3.2.3 Constraint matching

Let  $D$  a derivative,  $C$  a F-Logic rule body and  $X_D$  a free variable in  $C$ , then we call  $c$  a *constraint* of  $D$ , if  $D[\text{constraint} \rightarrow c]$ . and  $\forall X_D. \text{satCons}(X_D, c) \leftarrow C$ . holds. Variable  $X_D$  is bound with potential instances of  $D$ , and  $\text{satCons}$  verifies whether such an instance satisfies  $c$ . A derivative can have zero or many constraints including a special constraint for nominals; the respective meta-relation *oneOf* denoted as  $D[\text{oneOf} \rightarrow \{i_1, \dots, i_m\}]$  means that an instance of  $D$  has to be one of  $i_1, \dots, i_m$ .

In WSMO-MX, the matching of logical constraints of goal and service derivatives is performed by means of so called relative query containment. That is, any clause  $A$  is relatively contained in clause  $B$ , or  $B$  relatively implies  $A$ , with respect to a given knowledge base  $\mathcal{KB}$ , denoted by  $A \sqsubseteq_{\mathcal{KB}} B$ , if the answer set  $Q_{\mathcal{KB}}(A)$  of querying  $\mathcal{KB}$  with  $A$ , is a subset of  $Q_{\mathcal{KB}}(B)$ . Under the open world assumption,  $\mathcal{KB}$  does not contain all possible instances of a query (universal closure), hence relative query containment can only be considered as an approximation of logical implication (query containment) which is, in general, undecidable for first-order languages such as F-Logic [2]. An alternative would be to approximate logical implication by means of clause theta-subsumption [13] which is, in general, NP-complete decidable [3]. Since fast deterministic

algorithms for partial testing of theta-subsumption are also known [14], the correct but incomplete theta-subsumption relation is used as a consequence relation in many ILP systems [12], and the matchmaker LARKS [16].

However, for pragmatic reasons of implementation, WSMO-MX uses relative query containment for matching constraints over the instances stored in the matchmaker ontology. For each derivative  $D$  of type  $T$ , WSMO-MX determines a set of potential instances against which its constraints are evaluated as queries. This set comprises all instances of the concept  $T$  and instances of derivatives of type  $T$ :

$$\forall D, X_D. \text{potentialInstance}(D, X_D) \leftarrow \\ \exists T. D[\text{type} \rightarrow T] \wedge \\ (X_D : T \vee (\exists D_T. D_T[\text{type} \rightarrow T] \wedge X_D : D_T)).$$

The constraint matching filter then returns only those instances of this set which satisfy all constraints of  $D$ :

$$\forall X_D, D. \text{satAllCons}(X_D, D) \leftarrow \\ \text{potentialInstance}(D, X_D) \wedge \\ (\forall C. D[\text{constraint} \rightarrow C] \rightarrow \text{satCons}(X_D, C)) \wedge \\ ((\exists X. D[\text{oneOf} \rightarrow X]) \rightarrow D[\text{oneOf} \rightarrow X_D]).$$

The valuation of constraint matching is determined by the type of the set relation  $\rho$ , which is defined as  $\mathcal{I}_{\mathcal{KB}}(D_G) \rho \mathcal{I}_{\mathcal{KB}}(D_W)$  over the set  $\mathcal{I}_{\mathcal{KB}}(D) := \{X_D | \text{satAllCons}(X_D, D)\}$  of matching instances of derivative  $D$  with respect to the given knowledge base  $\mathcal{KB}$  of the matchmaker (cf. table 4).

set relation	valuation vector	
	$val_{pre}$	$val_{post}$
$\mathcal{I}_{KB}(D_G) \rho \mathcal{I}_{KB}(D_W)$	$( \pi_{\exists}, \pi_{\sqsubseteq}, \pi_{\sqsupset}, \pi_{\sqcap}, \pi_{\sim}, \pi_{\circ}, \pi_{\perp} )$	$( \pi_{\exists}, \pi_{\sqsubseteq}, \pi_{\sqsupset}, \pi_{\sqcap}, \pi_{\sim}, \pi_{\circ}, \pi_{\perp} )$
$\mathcal{I}_{KB}(D_G) = \mathcal{I}_{KB}(D_W)$	$( 1, 0, 0, 0, 0, 0, 0 )$	$( 1, 0, 0, 0, 0, 0, 0 )$
$\mathcal{I}_{KB}(D_G) \supseteq \mathcal{I}_{KB}(D_W)$	$( 0, 0, 1, 0, 0, 0, 0 )$	$( 0, 1, 0, 0, 0, 0, 0 )$
$\mathcal{I}_{KB}(D_G) \subseteq \mathcal{I}_{KB}(D_W)$	$( 0, 1, 0, 0, 0, 0, 0 )$	$( 0, 0, 1, 0, 0, 0, 0 )$
$\mathcal{I}_{KB}(D_G) \cap \mathcal{I}_{KB}(D_W) \neq \emptyset$	$( 0, 0, 0, 1, 0, 0, 0 )$	$( 0, 0, 0, 1, 0, 0, 0 )$
$\mathcal{I}_{KB}(D_G) \cap \mathcal{I}_{KB}(D_W) = \emptyset$	$( 0, 0, 0, 0, 0, 0, 1 )$	$( 0, 0, 0, 0, 0, 0, 1 )$

**Table 4. Valuation of constraint matching for hybrid matching degrees**

### 3.2.4 Syntactic matching

The filter of WSMO-MX for syntactic matching of goal and service derivatives,  $D_G$  and  $D_W$ , is intended to complement those for semantic matching as described above. For this purpose, it transforms the description of each derivative into a weighted keyword vector as known from information retrieval, and applies one of the selected syntactic similarity metrics cosine, extended Jaccard, loss-of-information (LOI), and weighted LOI [10], depending on the user preferences specified as instances of the following meta-relations of goal derivatives  $D_G$ .

- $D_G[synSimUsage \rightarrow U]$  with  $U \in \{alternative, compensative, complementary\}$  specifies whether syntactic matching shall be performed either as an exclusive alternative to semantic matching, or only in case of semantic matching failure, or in any case.
- $D_G[synSimScope \rightarrow S]$  with  $S \in \{scpType, scpRelation, scpDescription\}$  denotes whether only the types, or the relations, or the whole text of the description of the derivatives are used for syntactic matching. In case of *scpType*, all type names (no relation names) of the derivative are recursively unfolded in the matchmaker ontology and the resulting set of primitive components used to compute a weighted keyword vector, whereas for *scpRelation* only the relation names of the derivative are used for this purpose. Any combination of scopes is allowed.
- $D_G[synSimMetric \rightarrow M]$  with  $M \in \{cosine, loi, loiWeighted, jaccard\}$  specifies which IR similarity metric to use. For details of computation, we refer to [10].
- $D_G[synSimMinDegree \rightarrow \alpha]$  with  $\alpha \in [0, 1]$  specifies the minimum degree of syntactic similarity required (threshold).

The valuation of syntactic matching is considered only with respect to the degree of fuzzy similarity  $\pi_{\sim}$  and set to 0, if the computed syntactic similarity value does not exceed  $\alpha$ , and to 1 otherwise.

### 3.2.5 Parameter matching

A derivative can be tagged to be an input and/or output parameter by the meta-relation *param*. The parameter matching filter checks whether goal and service derivative are differently tagged and returns no valuation vector but an annotation indicating the deviations. This allows the service requester to understand the interface of the service and if needed to adjust the interface as it was expected and denoted by the parameters tags in the goal description.

### 3.2.6 Intentional matching

Optionally, WSMO-MX does perform a kind of intentional matching of goal and service derivatives. For this purpose, we adopt the approach proposed by Keller et al. [6]. In particular, the semantics of their notions of  $\exists$ -intention and  $\forall$ -intention correspond with the evaluation of our meta-relation *existentialIntention* to *true* and *false*, respectively. The valuation vector of hybrid semantic matching can be "intentionally recomputed" by its multiplication with the transformation matrix that corresponds to the requested combination of intended provision of relevant instances as it is declared for the goal and the service derivative by the requester and provider, respectively.

The case in which  $\forall$ -intentions are declared for both derivatives,  $D_G$  and  $D_W$ , is equal to not using intentions at all, hence can simply be ignored by WSMO-MX. As a consequence, there remain three cases for each pre- and postcondition matching. These are computed by means of six intentional matching matrices (to be multiplied with the valuation vector) of which we show only those for the postcondition matching cases <sup>1</sup>: (1)  $I_{post, \exists G, \forall W}$ : only  $D_G$  has an  $\exists$ -intention, (2)  $I_{post, \forall G, \exists W}$ : only  $D_W$  has an  $\exists$ -intention, (3)  $I_{post, \exists G, \exists W}$ : both derivatives have  $\exists$ -intentions. The matrices are defined as follows.

<sup>1</sup>For the cases of precondition matching the lines and columns for  $\pi_{\sqsubseteq}$  and  $\pi_{\sqsupset}$  in the matrices have to be inverted.

$$\begin{aligned}
I_{post,\exists G,\forall W} &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \\
I_{post,\forall G,\exists W} &= \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & 0 & 0 & \frac{1}{5} \\ 0 & 0 & \frac{1}{3} & \frac{1}{3} & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \\
I_{post,\exists G,\exists W} &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} \\ \frac{1}{3} & 0 & \frac{1}{3} & 0 & 0 & 0 & \frac{1}{3} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}
\end{aligned}$$

Due to space restrictions, we refer the interested reader for more details to [5].

### 3.3 WSMO-MX matching algorithm

The request for a semantically relevant service is specified by the user as a goal derivative in WSML-MX, together with a matching configuration  $Conf$ . The configuration contains default values for minimum syntactic similarity degree, weights for the aggregation of different matching filter results, and the minimum valuation of each degree of hybrid matching returned by the matchmaker. WSMO-MX takes the precondition state and the postcondition state of each advertised service from its local knowledge base (cf. algorithm 1), and then matches them pairwise with the states of the given goal (cf. algorithm 2). In case of no preconditions, the result of their matching is set to equivalence by default.

The state of the goal is matched with that of the service by matching their state part derivatives (cf. algorithm 3) and then recursively by the pairwise matching of relation range derivatives of equally named relations (cf. algorithm 5). Subsequently, WSMO-MX computes the *maximum* weighted bipartite graph match, where nodes of the graph correspond to the goal and service state parts and the computed valuation vectors act as weights of edges existing between matched state parts.

At each step in the recursion, the parameter matching filter is applied first, since its result, an annotation record, is not valued for any of the hybrid matching degrees. Then each of the semantic matching filters (type, constraint, and relation matching) is applied. Syntactic matching is per-

formed in case one of these filters fails (compensative), or complementary in any case, if not specified differently. The user can also ask for just a first coarse-grained filtering by means of exclusively syntactic matching without any semantic matching.

Finally, all valuation vectors computed during recursive matching of goal and service derivatives are aggregated into one single valuation vector. For aggregation, each individual valuation vector is weighted for the respective matching filter as specified in the configuration ( $Conf$ ) for the given goal; the weighting is assumed to be equal by default. This aggregated valuation of hybrid matching degrees is then recomputed with respect to the intentions of the considered derivatives (cf. in section 3.2.6).

The overall result of the matching process is a ranked list of services with their hybrid matching valuation vector, and annotations. Services are ranked with respect to the maximum value of hybrid semantic matching degrees in descending order (cf. table 1), starting with  $\pi_{\equiv}$ .

---

**Algorithm 1** WSMO-MX matching of query (goal  $G$ , configuration  $Conf$ ) with registered services in WSML-MX:  $matchGoal$

---

```

1: function MATCHGOAL( $G, Conf$ )
    $WS := GETREGISTEREDWEBSERVICES()$ 
2:    $\mathcal{S}_{G,pre} := GETPRECONDITION(G)$ 
3:    $\mathcal{S}_{G,post} := GETPOSTCONDITION(G)$ 
4:    $Conf_{pre} := Conf + (modus : pre)$ 
5:    $Conf_{post} := Conf + (modus : post)$ 
6:    $\mathcal{W}_{matched} := \text{empty set}$ 
7:   for all  $W \in WS$  do
8:      $\mathcal{S}_{W,pre} := GETPRECONDITION(W)$ 
9:      $\mathcal{S}_{W,post} := GETPOSTCONDITION(W)$ 
10:     $(Val_{W,pre}, Ann_{W,pre}) :=$ 
      MATCHSTATES( $\mathcal{S}_{G,pre}, \mathcal{S}_{W,pre}, Conf_{pre}$ )
11:     $(Val_{W,post}, Ann_{W,post}) :=$ 
      MATCHSTATES( $\mathcal{S}_{G,post}, \mathcal{S}_{W,post}, Conf_{post}$ )
12:     $\mathcal{W}_{matched} += (W, Val_{W,pre}, Val_{W,post},$ 
       $Ann_{W,pre}, Ann_{W,post})$ 
13:   end for
14:   return  $\mathcal{W}_{matched}$ 
15: end function

```

---

### 3.4 Implementation

WSMO-MX has been fully implemented in Java 5 and F-Logic using the F-Logic reasoner OntoBroker<sup>2</sup>. Its main components are a matching engine which is interfaced with an ontology manager communicating with the reasoner. Type and constraint matching is done directly within the OntoBroker, whereas the WSMO-MX matching engine

<sup>2</sup>developed by Ontoprise, <http://www.ontoprise.de>



---

**Algorithm 2** *matchStates*

---

```
1: function MATCHSTATES( $\mathcal{S}_G, \mathcal{S}_W, Conf$ )
2:   ▷ build bipartite weighted graph from
3:   ▷ matching state parts of goal and webservice
4:   Graph := empty graph
5:   for all StatePartG ∈  $\mathcal{S}_G$  do
6:     for all StatePartW ∈  $\mathcal{S}_W$  do
7:       (ValW, AnnW) := MATCHDERIVATIVES
           (StatePartG, StatePartW, Conf)
8:       if ¬ ISFAIL(ValW) then
9:         Graph += edge(StatePartG,
           StatePartW, ValW, AnnW)
10:      end if
11:    end for
12:  end for
13:
14:  ▷ find maximum weighted graph matching
15:  M := GETGRAPHMATCHING(Graph)
16:  (Val, Ann) := GETVALANN(M, Conf)
17:
18:  ▷ valueate not matched state parts
19:   $\mathcal{S}_{G-M}$  := NOTMATCHEDSTATEPARTS( $\mathcal{S}_G, M$ )
20:   $\mathcal{S}_{W-M}$  := NOTMATCHEDSTATEPARTS( $\mathcal{S}_W, M$ )
21:  for all StatePartG ∈  $\mathcal{S}_{G-M}$  do
22:    Val += VALSTATEPART(goal, Conf)
23:    Ann += (G, W, state,
           (StatePartG, notMatched, goal))
24:  end for
25:  for all StatePartW ∈  $\mathcal{S}_{W-M}$  do
26:    Val += VALSTATEPART(webservice, Conf)
27:    Ann += (G, W, state,
           (StatePartW, notMatched, webservice))
28:  end for
29:
30:  ▷ normalize cumulated valuation
31:  Val /= |M| +  $\mathcal{S}_{G-M}$  +  $\mathcal{S}_{W-M}$ 
32:  return (Val, Ann)
33: end function
```

---

---

**Algorithm 3** *matchDerivatives*

---

```
1: function MATCHDERIVATIVES( $D_G, D_W, Conf$ )
2:
3:   if  $D_G = D_W$  then return ((1, 0, 0, 0, 0, 0, 0), ())
4:   end if
5:
6:   AnnParams := MATCHPARAMS( $D_G, D_W$ )
7:   synMatchUsage :=
           GETSYNMATCHINGUSAGE( $D_G, Conf$ )
8:
9:   if synMatchUsage = alternative then
10:    (ValSyn, AnnSyn) :=
           MATCHSYNTACTIC( $D_G, D_W, Conf$ )
11:    if ¬ ISFAIL(ValSyn) then
12:      Ann += AnnParams + AnnSyn
13:      return (ValSyn, Ann)
14:    end if
15:  end if
16:
17:  (ValSem, AnnSem) :=
           MATCHSEMANTIC( $D_G, D_W, Conf$ )
18:
19:  if ISFAIL(ValSem) then
20:    if synMatchUsage = compensative then
21:      (ValSyn, AnnSyn) :=
           MATCHSYNTACTIC( $D_G, D_W, Conf$ )
22:      if ¬ ISFAIL(ValSyn) then
23:        Ann += AnnParams + AnnSyn
24:        return (ValSyn, Ann)
25:      end if
26:    end if
27:  else if synMatchUsage = complementary then
28:    (ValSyn, AnnSyn) :=
           MATCHSYNTACTIC( $D_G, D_W, Conf$ )
29:    if ¬ ISFAIL(ValSyn) then
30:      Ann += AnnParams + AnnSyn + AnnSem
31:      Val :=
           AGGREGATEVAL(ValSem, ValSyn, Conf)
32:      return (Val, Ann)
33:    end if
34:  else
35:    Ann += AnnParams + AnnSem
36:    Val :=
           AGGREGATEVAL(ValSem, null, Conf)
37:    return (Val, Ann)
38:  end if
39: end function
```

---

---

**Algorithm 4** *matchSemantic*

---

```
1: function MATCHSEMANTIC( $D_G, D_W, Conf$ )
2:   ( $Val_{Type}, Ann_{Type}$ ) :=
      MATCHTYPES( $D_G, D_W, Conf$ )
3:   ( $Val_{Cons}, Ann_{Cons}$ ) :=
      MATCHCONSTRAINTS( $D_G, D_W, Conf$ )
4:   ( $Val_{Rel}, Ann_{Rel}$ ) :=
      MATCHRELATIONS( $D_G, D_W, Conf$ )
5:
6:    $Val := (Val_{Type}, Val_{Cons}, Val_{Rel})$ 
7:    $Ann := Ann_{Type} + Ann_{Cons} + Ann_{Rel}$ 
8:
9:   return ( $Val, Ann$ )
10: end function
```

---

---

**Algorithm 5** *matchRelations*

---

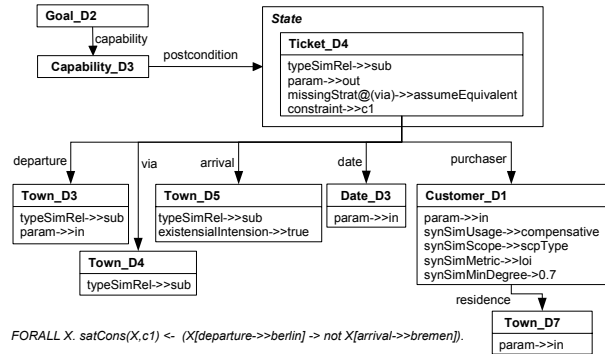
```
1: function MATCHRELATIONS( $D_G, D_W, Conf$ )
2:    $\triangleright Rels_G$  - relations defined only for  $D_G$ 
3:    $\triangleright Rels_W$  - relations defined only for  $D_W$ 
4:    $\triangleright Rels_{G,W}$  - relations defined for both
5:   ( $Rels_G, Rels_W, Rels_{G,W}$ ) :=
      GETRELATIONS( $D_G, D_W$ )
6:
7:    $\triangleright$  for all relations defined in  $D_G$  and  $D_W$ 
8:    $\triangleright$  match the derivatives in their range
9:   for all  $R \in Rels_{G,W}$  do
10:      $Range_G := GETRELRange(D_G)$ 
11:      $Range_W := GETRELRange(D_W)$ 
12:     ( $Val_{Range}, Ann_{Range}$ ) :=
        MATCHDERIVATIVES( $Range_G, Range_W$ )
13:      $Val += Val_{Range}$ 
14:      $Ann += Ann_{Range}$ 
15:   end for
16:    $\triangleright$  evaluate relations defined only for  $D_G$ 
17:   for all  $R \in Rels_G$  do
18:      $MS_R :=$ 
        GETMISSINGSTRATEGY( $D_G, R, Conf$ )
19:      $Val +=$ 
        VALUATEMISSREL( $webservice, MS_R, Conf$ )
20:      $Ann += (D_G, D_W, rel,$ 
        ( $R, missing, webservice, MS_R$ ))
21:   end for
22:    $\triangleright$  evaluate relations defined only for  $D_W$ 
23:   for all  $R \in Rels_W$  do
24:      $Val += VALUATEMISSREL(goal, null, Conf)$ 
25:      $Ann += (D_G, D_W, rel, (R, missing, goal))$ 
26:   end for
27:
28:    $Val /= |Rels_G| + |Rels_W| + |Rels_{G,W}|$ 
29: end function
```

---

takes the results and does the rest, that is relation matching, syntactic matching, aggregation of valuation vectors, state matching including the computation of maximum weighted bipartite graph matching. The OntoBroker loads the matchmaker ontology from a given set of F-Logic files that contain the types, derivatives (including goals and services), instances, and constraints, as well as the rules for type and constraint matching, unfolding and some auxiliary tasks. In an upcoming version of WSMO-MX, the goals will be passed by the matching engine to the ontology manager only at the time of the respective request to the matchmaker.

## 4 Example

**Goal, service, ontology.** Suppose the user defines a goal derivative *Ticket\_D4* as shown in figure 4. That is, she is looking for any ticket for a trip between two arbitrary towns, but if it starts in Berlin, then it must not end in Bremen. Please note, that the user may specify matching relaxations for any object of the goal as exemplified, but also different weights for the matching filters to be applied. In this example, we assume the filters to be equally weighted.

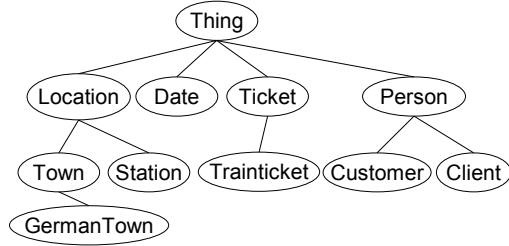


**Figure 4.** Example goal in WSML-MX

The part of the type hierarchy in the matchmaker ontology and all instances used in this example are shown in figure 5.

For reasons of efficiency and data privacy, mediation between service providers and requesters by means of an autonomous matchmaker is not appropriate for constraint matching over different instance bases. Alternatively, an autonomous WSMO-MX matchmaker could perform constraint matching without instance sets by polynomial means of theta-subsumption reasoning for restricted set of Horn clauses like in LARKS [16]. This is part of our future work on WSMO-MX.

In this example, the service derivative *Ticket\_D5* given in section 2 will be matched against the goal derivative *Ticket\_D4*. Please note, that the service offers tickets for any trip between any two German towns, but if the user



```

t1:Ticket_D4[departure->>berlin;
arrival->>leipzig; ...].
t2:Ticket_D4[departure->>berlin;
arrival->>kiel; ...].
t3:Ticket_D5[departure->>hamburg;
arrival->>bremen; ...].
t4:Ticket_D5[departure->>hamburg;
arrival->>hannover; ...].
t5:Ticket_D5[departure->>berlin;
arrival->>hamburg; ...].
t6:Ticket_D6[departure->>berlin;
arrival->>bremen; ...].

```

**Figure 5. Example ontology (type hierarchy and instances)**

departs from Berlin, her destination must be Hamburg.

**Matching.** Since the capabilities of both goal and service derivatives do not include any precondition, the hybrid semantic matching of them is restricted to the matching of their postcondition states as follows.

1. **match types:** the types of Ticket\_D4 and Ticket\_D5 are equal. Hence the valuation is  $v_1 = (1, 0, 0, 0, 0, 0, 0)$ .
2. **match parameters:** both are output parameters, no annotation necessary
3. **match relations**
  - (a) *departure:* the types of Town\_D3 and GermanTown\_D1 are not equivalent, but Town\_D3 allows subtypes. Since GermanTown is a subconcept of Town, the valuation is  $v_2 = (0, 1, 0, 0, 0, 0, 0)$ .
  - (b) *via:* this relation is not defined for Ticket\_D3, but the missingStrategy for this relation is *assumeEquivalent* yielding a valuation  $v_3 = (1, 0, 0, 0, 0, 0, 0)$ .
  - (c) *arrival:* analogous to *departure* types of the ranges of *arrival* are subtypes and yield the valuation  $v_4 = (0, 1, 0, 0, 0, 0, 0)$ .
  - (d) *date:* is equal in goal and service, hence valued as  $v_5 = (1, 0, 0, 0, 0, 0, 0)$

- (e) *purchaser:* type matching fails for Customer\_D1 and Client\_D1, but compensative syntactic matching is allowed using loss of information (LOI) metric. For the unfolding only the types of the derivatives should be used (*scpType*), yielding the term vectors ( $Customer : 1, Town : 1, Person : 1, Location : 1, Town : 1$ ) and ( $Client : 1, Town : 1, Person : 1, Location : 1, Town : 1$ ) for Customer\_D1 and Client\_D1, respectively. The similarity degree is 0.75, and therefore greater than the declared minimum of 0.7. The resulting valuation vector is  $v_6 = (0, 0, 0, 0, 0, 0, 1, 0)$ .

The aggregated relation valuation is  $v_7 = \frac{v_2 + \dots + v_6}{5} = (0.4, 0.4, 0, 0, 0, 0.2, 0)$

4. **match constraints:** Ticket\_D4 has the constraint c1. This is satisfied by the instances  $t1, \dots, t5$ . The constraint c2, which is imposed on Ticket\_D5 is satisfied by the instances  $t3, \dots, t5$ . That means the instances for Ticket\_D5 are a subset of those of Ticket\_D4 and hence the valuation is  $v_8 = (0, 1, 0, 0, 0, 0, 0, 0)$

Finally, the aggregated valuation for the derivative matching of Ticket\_D4 and Ticket\_D5 is

$$v_9 = \frac{v_1 + v_7 + v_8}{3} = (\frac{7}{15}, \frac{7}{15}, 0, 0, 0, \frac{1}{15}, 0)$$

## 5 Related work

To the best of our knowledge, WSMO-MX is the first implemented full-fledged matchmaker for WSMO-oriented services. It borrows the approach to recursive object-oriented structure matching from [9], the notion of intentional matching from [6], and the hybrid semantic matching from [10]. The mediator based discovery approaches presented in [7, 17] do not allow for a general goal-service matching, but require problem specific mapping, or construction rules. Besides, like in [15], they define their notions of match on the assumption that an advertisement postcondition has to subsume the goal's postcondition for a full match, which is diametrically opposed to our approach and to the original idea of how to match program capabilities initially proposed in [4, 18].

Other relevant approaches to automated selection of semantic web services include those for retrieving relevant OWL-S services [11, 15]. Most of them rely on DL based subsumption reasoning. However, OWL still lacks the support of rules and subsumption reasoning in the underlying description logic  $\mathcal{SHOIN}(\mathcal{D})$  is NEXPTIME. Besides, unlike WSMO, there is no way in OWL-S to link I/O parameters in the signature with preconditions and effects as shared variables. Thus, most OWL-S matchmakers perform

signature matching only. OWLS-MX [10] complements the logic based semantic matching of OWL-S service signatures with syntactic matching, which is also rudimentary performed in LARKS [16]. For WSMO-MX, we did improve on this idea of OWLS-MX by allowing for a more fine-grained parametrisation, and integrated interleaving of syntactic and semantic matching.

## 6 Conclusions

In this paper we presented the general purpose matchmaker WSMO-MX for services described in WSML-MX which is a LP based variant of WSML-Rule that facilitates matching of pre- and postconditions of object-oriented descriptions of goals and services. WSMO-MX applies different matching filters to retrieve WSMO services that are semantically relevant to a given goal with respect to seven degrees of hybrid matching. Each of these degrees are recursively computed by aggregated valuations of ontology based type matching, logical constraint and relation matching, and syntactic similarity of goal and service derivatives. It integrates signature matching with state matching, and returns not only the final aggregated valuation vector for the hybrid matching degrees but an annotation of the matching results for interactive goal refinement by the user. Currently, relation cardinalities are not considered by WSMO-MX but will be integrated as soon as they become standardised<sup>3</sup> and supported by an F-Logic reasoner. Though the matchmaker has been fully implemented, the evaluation of its performance is ongoing work with generating the required WSMO service retrieval test collection first. Like with OWLS-MX, we intend to make WSMO-MX (without OntoBroker<sup>4</sup>) available to the semantic community under GPL-like license at the semwebcentral.org portal.

## References

[1] J. Angele and G. Lausen. Ontologies in f-logic. In S. Staab and R. Studer, editors, *Handbook on Ontologies*, pages 29–50. Springer, 2004.

[2] E. Dantsin, T. Eiter, G. Gottlob, and A. Voronkov. Complexity and expressive power of logic programming. *ACM Computing Surveys*, 33(3):374–425, September 2001.

[3] G. Gottlob and A. Leitsch. On the efficiency of subsumption algorithms. *Journal of the ACM (JACM)*, 32(2):280 – 295, April 1985.

[4] C. Hoare. An axiomatic basis for computer programming. *Communications of the ACM (CACM)*, 12(10):576–580, 10 1969.

[5] F. Kaufer. *WSMO-MX: A Logic programming based hybrid semantic web service matchmaker*. Computer Science Dept., University of the Saarland, Saarbruecken, Germany, 2006.

[6] U. Keller, R. Lara, H. Lausen, A. Polleres, and D. Fensel. Automatic location of services. In *Proceedings of the 2nd European Semantic Web Symposium (ESWS2005)*, Heraklion, Crete, June 2005.

[7] M. Kifer, R. Lara, A. Polleres, C. Zhao, U. Keller, H. Lausen, and D. Fensel. A logical framework for web service discovery. In *Proceedings of the ISWC 2004 Workshop on Semantic Web Services: Preparing to Meet the World of Business Applications*, volume 119, Hiroshima, Japan, November 2004. CEUR Workshop Proceedings.

[8] M. Kifer, G. Lausen, and J. Wu. Logical foundations of object-oriented and frame-based languages. *Journal of the ACM*, 42, 1995.

[9] M. Klein and B. König-Ries. Coupled signature and specification matching for automatic service binding. In *Proceedings of European Conference on Web Services (ECOWS 2004)*, LNCS 3250, page 183, Erfurt, Germany, September 2004. Springer.

[10] M. Klusch, B. Fries, M. Khalid, and K. Sycara. Owls-mx: Hybrid owl-s service matchmaking. In *Proceedings of 1st Intl. AAI Fall Symposium on Agents and the Semantic Web*, Arlington VA, USA, November 2005.

[11] L. Li and I. Horrocks. A software framework for matchmaking based on semantic web technology. In *Proceedings of the Twelfth International Conference on World Wide Web*, pages 331–339. ACM Press, 2003.

[12] S. Muggleton and L. D. Raedt. Inductive logic programming: Theory and applications. *Logic Programming*, 19(20):629–679, 1994.

[13] J. Robinson. A machine-oriented logic based on the resolution principle. *Journal of the ACM*, 12(1), 1965.

[14] T. Scheffer, R. Herbrich, and F. Wyszotki. Efficient algorithms for theta-subsumption. *JAIR*, 1997.

[15] K. Sycara, M. Paolucci, A. Ankolekar, and N. Srinivasan. Automated discovery, interaction and composition of semantic web services. *Journal of Web Semantics*, 1(1):28, 2003.

[16] K. Sycara, S. Widoff, M. Klusch, and J. Lu. Larks: Dynamic matchmaking among heterogeneous software agents in cyberspace. *Autonomous Agents and Multi-Agent Systems*, 5:173–2003, June 2002.

[17] E. D. Valle and D. Cerizza. Cocoon glue: a prototype of wsmo discovery engine for the healthcare field. In *Proceedings of the WIW 2005 Workshop on WSMO Implementations*, volume 134, Innsbruck, Austria, June 2005. CEUR Workshop Proceedings.

[18] A. M. Zaremski and J. M. Wing. Specification matching of software components. In *3rd ACM SIGSOFT Symposium on the Foundations of Software Engineering*, 10 1995.

<sup>3</sup>For more information see <http://forum.projects.semwebcentral.org/>

<sup>4</sup>research licences can be obtained from Ontoprise

## Semantic Service Discovery in Pure P2P Networks

M. Klusch and U. Basters: Risk Driven Semantic P2P Service Retrieval. Proceedings of the 6th IEEE International Conference on Peer-To-Peer Computing (P2P), Cambridge, UK, IEEE CS Press, 2006.

# Risk Driven Semantic P2P Service Retrieval

Matthias Klusch, Ulrich Basters  
German Research Center for Artificial Intelligence (DFKI)  
Saarbruecken, Germany  
klusch@dfki.de, ulrich@basters.de

## Abstract

*In this paper, we present a novel approach, named RS2D, to risk driven semantic service query routing in unstructured, so called pure P2P networks. Following the RS2D protocol, each peer dynamically learns about the query answering behavior of its direct neighbours. without prior knowledge on the semantic overlay. The decision to whom to forward a given service request is then driven by the estimated mixed individual Bayes' conditional risk of routing failure in terms of both semantic loss and high communication costs. The results of our experimental evaluation of retrieval performance and robustness show that RS2D top performs compared to other relevant systems.*

## 1. Introduction

The retrieval of relevant services is one key to service oriented computing in the web and semantic web. As of today, web services are supposed to be still discovered mainly by means of central repositories or registries such as UDDI [1]. In contrast, unstructured P2P service networks are expected to be robust against dynamic changes in the underlying network topology at the very expense of administrative communication overhead in due course of the self-regulation of the peers. The major challenge of decentralized semantic Web service retrieval in unstructured P2P service networks is to keep the communication costs of service retrieval low with reasonably high precision of the returned results.

Different approaches to solve this problem have been proposed in the literature; an accessible survey is provided in [2]. Whereas broadcast-based approaches are very robust with high precision they typically suffer from poor scalability due to their high communication overhead. Randomized routing usually keeps the communication effort low, but at the expense of low precision of the returned results. Our solution to this problem is the first risk assessment driven semantic service query routing protocol,

named RS2D. Key idea is to let the peers dynamically learn the average query-answer behavior of their direct neighbours in the network for making individual probabilistic risk based routing decisions with respect to both semantic gain and communication costs. In contrast to other existing approaches, RS2D does not require any prior knowledge on the environment including service distribution, global ontologies, or network topology. We implemented the RS2D protocol and experimentally evaluated its performance and robustness in randomly generated unstructured P2P networks in different scenarios.

The remainder of this paper is organized as follows: After brief discussion of related work in section 2, we provide the outline of the RS2D protocol in section 3, and provide the details of the underlying Bayesian risk based routing decision rule in section 4. Section 5 provides and discusses the results of the experimental evaluation of RS2D compared with other relevant approaches. Section 6 presents some insights into the implementation of the RS2D system and its simulation, and section 7 concludes the paper.

## 2. Related Work

The GSD-algorithm by Chakraborty et al.[4] takes advantage of the hierarchical structure of a global underlying ontology of the semantic network. Peers advertise their services not as service description, but with an ontological classification. When a peer gets a request for a service, it uses this classification to determine the distance between the requested and the offered services in the ontology tree. This approach has the clear disadvantage that only a static and commonly known ontology can be used. Additionally, sometimes the ontological classification might not be sufficient to find matching services, e.g. if they differ in their input and output parameters.

Another approach was proposed by Haase et al. with Bibster [8]. Their system relies on service advertisements that build up a semantic topology overlay. This is done by a special advertisement caching policy: peers add advertised services of neighbours to their list of known services

only if they are semantically close to at least one of their own services. This way, peers become experts for semantically similar services. When a query then asks for a certain service, Bibsters routing mechanism chooses those two neighbours whose expertise is closest to the query. Thus the query travels along a path of peers with similar expertise what increases the result precision and decreases communication overhead. However, the message traffic induced by the initial exchange of service advertisements is rather high. Also, prior knowledge about other peers' ontologies as well as their mapping to local ontologies is assumed.

To the best of our knowledge, there exist no other relevant and implemented solutions to the problem of decentralized semantic service retrieval in unstructured P2P networks.

### 3 RS2D Routing Protocol Overview

One major challenge of decentralized service retrieval in unstructured P2P networks, is to achieve a reasonably high retrieval performance with low communication costs without any prior knowledge about the environment including services, ontologies, or network topology. There is no central directory or repository in the system. The basic idea of our solution to the problem is to allow each peer to quickly learn which of its direct neighbours in the network will probably return relevant semantic web services for a given query with minimal risk of both semantic loss and high communication in total. We first outline the RS2D protocol to be followed by each peer, and then provide the details of it in subsequent sections.

Let be for each peer  $v$ ,  $q$  a service request (query);  $S$  set of locally known services,  $S_q$  the current top- $k$  relevant services (URIs) retrieved;  $a \in R$  the communication effort of propagating  $q$ , that is the number of messages in the routing subtree for  $q$  in the network graph;  $TS$  the individual training set of a peer consisting of information about previous queries and their results;  $hop \in \mathbb{N}$  the distance from  $v$  in the network. Then, each peer  $v$  performs the following steps:

- Determine the set  $S'_q$  of services that are semantically relevant to  $q$ :  $S'_q = S_q \cup \{\forall s \in S : \sigma(s, q)\}$ .

The function  $\sigma(s, q) \in [0, 1]$  maps the matching results of the used semantic web service matchmaker to  $[0, 1]$ , where  $\sigma(s, q) = 0$  and  $\sigma(s, q) = 1$  represent a matching failure and exact match, respectively. For our experiments with RS2D, we used the hybrid OWL-S service matchmaker OWLS-MX [9] which renders RS2D independent from any fixed global ontology, as this matchmaker dynamically maintains a local matchmaker ontology by means of logic based rea-

soning upon provided service advertisements and requests (see also sect. 6).

- For each peer  $v_k$  in the direct neighbourhood of  $v$  ( $hop = 1$ ):
  - Estimate the expected semantic gain  $E(y)$ , and communication costs  $E(a)$  of forwarding request  $r = (q, S_q, S'_q, a)$  to  $v_k$  based on the actual training set  $TS$ .
  - Compute the individual Bayes' conditional risk of routing  $r$  to  $v_k$ , or not (cf. section 4).
  - Send  $r$  to  $v_k$ , if the risk of forwarding is minimal, or if (initially)  $TS = \emptyset$  then multicast  $r$  to all neighbours.
- Observe the query answer behavior of neighbour peers  $v_k$  by storing received replies with a semantic score  $L(S''_q)$  of intermediate results  $S''_q$  returned and communication costs  $a$  per query in the local training set  $TS$ . The semantic score measures the quality of the set of retrieved services with respect to  $q$  by means of  $L(S_q) := \sum_{s \in S_q} \sigma(s, q)$ .
- Reject a received request  $r$ , if it has been already processed locally, or a fixed number of forwarding steps (hops) is reached, or the risk of further forwarding is maximal for each of its neighbours.
- Return set of top- $k$  semantically matching services in a priority queue if the semantic gain is positive, that is  $L(S''_q) - L(S'_q) > 0$ .

Each peer collects the replies on query  $q$  it receives from its neighbours and merges them together with its local results set which is then returned to the one who did forward  $q$  to it. This way, the result set for a query is created while being propagated back to its origin. At the same time, each peer involved in this process continuously learns about the query answering behaviour of each of its neighbours in general. It caches the individual observations in its local training set each time it receives a reply. This, in turn, enables each peer to estimate the corresponding risk of forwarding a query to individual peers.

### 4 Bayes Risk of Query Routing

The decision of each peer to route a given query  $q$  to any of its neighbours  $v_k$  is based on the individual estimated mixed risk of doing so in terms of both semantic gain and communication costs. The estimated semantic gain  $E(y)$ , the estimated communication costs  $E(a)$  as well as the probability with which a neighbour will answer are computed from the training set  $TS$  by means of a naive Bayes

approach [5]. More concrete, the risk assessment driven routing decision bases on the computation of the individual Bayes' conditional risk defined as:

$$R(\alpha_i|x) = \sum_{j=1}^{|C|} \lambda(\alpha_i, c_j) \cdot P(c_j|x) \quad (1)$$

with

- Binary routing alternatives  $\alpha_0$  and  $\alpha_1$  for not routing, respectively, routing the query.
- Query answer class set  $C = \{c_0, c_1\}$  with classes  $c_0$  (= query rejected because it already was processed by  $v_k$ ) and  $c_1$  (=  $v_k$  answers to the query with a semantic gain, i.e. with  $L(S''_q) - L(S'_q) > 0$ ).
- Observation  $x$  of query answering behavior of  $v_k$  for past queries
- Mixed semantic and communication loss  $\lambda(\alpha_i, c_j)$  for routing alternative  $\alpha_i$  and query answer class  $c_j$ .
- Conditional probability  $P$  of query answering class  $c_j$  for given observation  $x$ .

Having computed the mixed risk values for each binary routing alternative for each of its neighbours, the peer then routes the query  $q$  only to those peers for which the corresponding alternative with minimal risk

$$\alpha^* = \operatorname{argmin}\{R(\alpha_0|x), R(\alpha_1|x)\} \quad (2)$$

is  $\alpha_1$ , otherwise rejects. This minimizes the overall risk  $R = \int R(\alpha(x)|x)P(x)dx$  in compliance with the known Bayes Decision Rule, in other words a decision for the alternative with minimal overall risk is optimal.

What does an individual peer observe in concrete terms? From each reply to a given query  $q$  it received from some neighbour  $v_k$ , it extracts data into a training record  $t = (q, S'_q, S''_q, L(S'_q), L(S''_q), fid, tid, c_j, a)$  and stores it in a local training set  $TS$ . These observation data are as follows:

$q$ : Request in terms of the description of a desired service written in a semantic web service description language such as OWL-S.

$S'_q$ : Set of top- $k$  relevant services retrieved *before* forwarding the request.

$S''_q$ : Set of top- $k$  relevant services retrieved *after* forwarding the request.

$L(S'_q), L(S''_q)$ : Semantic score of  $S'_q, S''_q$

$fid$ : Identifier of the peer from which the request was received.

$tid$ : Identifier of the peer to which the request was forwarded.

$c_j$ : Query answer result class ( $c_0$  or  $c_1$ ).

$a$ : Communication effort entailed by the decision to route the request to  $v_k$ , i.e. the number of message hops in the routing subtree of the request.

The observation vector  $x \in \mathbb{N}^2$  used for risk estimations is defined as  $x = (fid, tid)$ . Our experiments showed, that already the use of these two parameters yield an reasonably well prediction. To be able to predict the values of  $\lambda, E(y), E(a)$  and  $P(c_j|x)$ , we filter the training set in different ways. Let  $TS_{p_1, \dots, p_z} \subset TS$  denote the set of training records  $t$  with parameters  $p_1$  to  $p_z$  set to given values, for example,  $TS_{fid, tid}$  the subset which has the given values for  $fid$  and  $tid$  (here:  $z = 2$  having  $p_1 = fid$  and  $p_2 = tid$ ).

The estimated semantic loss of routing  $q$  to a peer  $v_k$  (alternatives  $\alpha_0, \alpha_1$ ) for possible query answer classes ( $c_0, c_1$ ) based on its average Q/A behavior according to the actual training set is computed as follows:

$$\begin{array}{c|c|c} & \lambda(\alpha_0|\cdot) & \lambda(\alpha_1|\cdot) \\ \hline c_0 & -E(a)\kappa & 2\kappa \\ c_1 & \tau E(y) & -\tau E(y) \end{array} \quad (3)$$

The average message transmission costs are denoted by  $\kappa$ , and assumed to be constant. In addition, the average expected semantic gain  $E(y)$  and average number of messages  $E(a)$  are defined as follows:

$$E(y) := \frac{1}{|TS_{fid, tid}|} \sum_{t \in TS_{fid, tid}} [L(S''_q)]_t - [L(S'_q)]_t \quad (4)$$

$$E(a) := \frac{1}{|TS_{fid, tid}|} \sum_{t \in TS_{fid, tid}} [a]_t \quad (5)$$

with  $[x]_t$  extracting the parameter  $x$  from observation record  $t$  in the training set  $TS$ . The real-valued user preference parameter  $\tau$  denotes the weighted relation between maximum semantic gain ( $y = 1$ ) and communication costs the user is willing to accept; in our experiments, we obtained the best results with  $\tau = 1000$ . Each of the above defined cases of semantic loss of a routing decision by an individual peer  $v$  with respect to forwarding a given request to one of its neighbor peers  $v_k$  is justified as follows:

$\lambda(\alpha_0|c_0)$  No routing of the request to the targeted peer  $v_k$  takes place, but it would have been rejected by this peer anyway. As a consequence, the risk based decision is of benefit for  $v$  in terms of saved communication efforts ( $-E(a)\kappa$ ).



$\lambda(\alpha_0|c_1)$  In this case, peer  $v$  does not forward the query to  $v_k$  but would have received a positive reply with semantic gain. Hence, the loss is computed in terms of the costs of the lost opportunity, that is the semantic gain weighted by its relation to individually preferred upper bound of communication costs ( $\tau E(y)$ ).

$\lambda(\alpha_1|c_0)$  Peer  $v$  decides to route the query to  $v_k$  which rejects it. Hence, the decision was not beneficial for  $v$  in that it produced unnecessary communication costs of the specific request and reply.

$\lambda(\alpha_1|c_1)$  The request of peer  $v$  will be answered by  $v_k$  with some expected semantic gain for  $v$ . Hence, the decision is beneficial for  $v$  in terms of negative loss (utility  $-\tau E(y)$ ).

Using  $\lambda$  for computing the risk of routing alternatives  $\alpha_0, \alpha_1$  does reflect the classical loss relation between utility and costs:

$$R(\alpha_0|x) = -E(a) \cdot \kappa \cdot P(c_0|x) + \tau \cdot E(y) \cdot P(c_1|x) \quad (6)$$

$$R(\alpha_1|x) = 2 \cdot \kappa \cdot P(c_0|x) - \tau \cdot E(y) \cdot P(c_1|x) \quad (7)$$

Alternatively, one could have defined the semantic loss of the query answering class  $c_1$  directly as difference between expected semantic gain and average communication costs in terms of number and volumes of messages exchanged:

	$\lambda'(\alpha_0 \cdot)$	$\lambda'(\alpha_1 \cdot)$	
$c_0$	$-E(a)\kappa$	$2\kappa$	
$c_1$	$E(y) - E(a)\kappa$	$-E(y) + E(a)\kappa$	

(8)

However, according to the results of our experimental evaluation, this option can be significantly improved by the one chosen in terms of retrieval performance with only slightly increased communication efforts.

Then, the conditional probability  $P(c_j|x)$  of possible answering result classes of the considered peer  $v_k$  based on its observed Q/A behavior in the past is computed as usual based on the prior probability  $P(x|c_j)$ , the likelihood  $P(c_j)$ , and the normalizing evidence factor  $P(x)$  from the training set  $TS$ :

$$P(c_j|x) = \frac{P(x|c_j) \cdot P(c_j)}{P(x)} \quad (9)$$

with

$$cP(c_j) = \frac{|TS_{c_j}|}{|TS|} \quad (10)$$

$$P(x|c_j) = \prod_{l=1}^n P(x_l, c_j) \quad (11)$$

$$P(x) = \sum_{j=1}^{|C|} P(x|c_j) \cdot P(c_j) \quad (12)$$

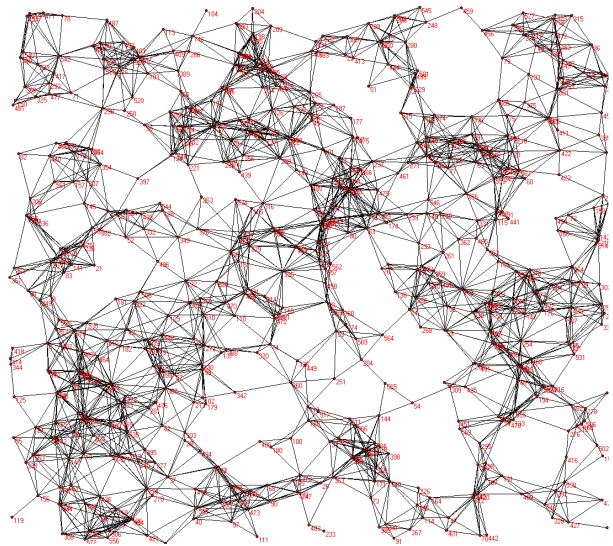
with the probability  $P(x_l, c_j)$  of the occurrence of the observation feature component  $x_l$  together with class  $c_j$  defined as

$$P(x_l, c_j) = \frac{|TS_{x_l, c_j}|}{|TS_{c_j}|} \quad (13)$$

The decision making process heavily relies on the training set  $TS$  that each peer maintains individually. Initially, when a peer joins the network, its training set  $TS$  is empty; in this case, it sends its queries to all its direct neighbours until the size ( $\theta(TS)$ ) of its training set, more specifically  $TS_{fid, tid}$  is sufficiently large for continuing with risk assessment driven routing decisions from this point. Our experiments provide evidence in favor of  $\theta(TS_{fid, tid}) = 1$  ( $\theta_{TS} = 8$  when using the alternative semantic loss definition in equ.(8)).

## 5 Evaluation

We have implemented the RS2D protocol and evaluated it by means of simulation. For this purpose, we randomly generated unstructured, sparsely connected P2P networks of different size with 50, 100, 200, and 576 peers, and used the OWLS-TC2 service retrieval test collection [6] which contains 576 OWL-S services, 36 queries with relevance sets, and the OWLS-MX matchmaker [7] for semantic matching by each peer.



**Figure 1. Example of unstructured network of 576 peers used in our experiments**

In each simulation run, the queries are sequentially processed by each peer to generate the training set, and the top  $k$  ( $k = 20$ ) services are returned by each peer only. The

P2P service retrieval performance is measured in terms of micro-averaged precision and recall against communication overhead with respect to the maximum hop count for query propagation.

We evaluated the performance of the RS2D service query routing mechanism against the following relevant alternative approaches:

**BCST** Classic broadcast based routing forwards the query to all direct neighbours until a maximal number of hops is reached, or all neighbours reject the query. BCST always yields optimal precision but at the very expense of communication efforts.

**RND2** Random peer selection: This method randomly selects two direct neighbours of a peer  $v_m$  to which the query is forwarded. RND2 has low communication costs but low precision as well. It is also used by developers of BIBL in [8] for the comparison of performance.

**BIBL** Bibster-like routing: This routing mechanism simulates the one used in the P2P system Bibster [8]. In particular, peers have prior knowledge about a fixed semantic overlay network that is initially built by means of a special advertisement caching policy. Each peer only stores those advertisements that are semantically close to at least one of their own services, and then selects for given queries only those two neighbours with top ranked expertise according to the semantic overlay it knows in prior.

### 5.1 Service retrieval performance

In our experiments, we evaluated two essential aspects of P2P service retrieval performance measurement:

1. Service distribution to peers: Uniformly at random Vs. Single peer hosts all relevant services per query
2. Query distribution to peers by the user: Random querying of peers Vs. One central Q/A peer, as a single entry point to the system for the user

For reasons of space limitation, we present only representative results of selected experiments.

**Experiment 1:** As figure 2 shows, in a network of 576 peers with evenly distributed 576 services, and random querying of peers, RS2D outperforms BIBL as well as RND2 in terms of precision with lesser number of hops which yields a better response time. The same results can be obtained for RS2D in smaller networks.

However, unlike BCST this nearly optimal performance of RS2D does not come at the very expense of communication but only almost one third and twice of that of BCST and BIBL, respectively, as shown in figure 3.

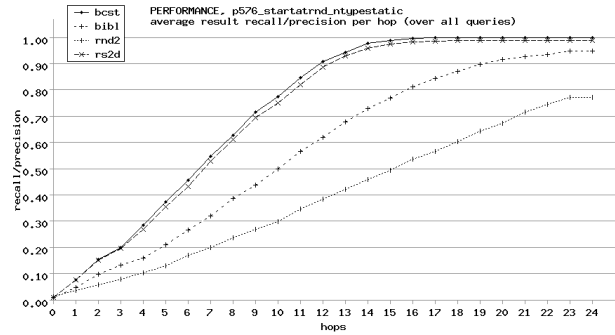


Figure 2. Experiment 1, Precision: RS2D outperforms BIBL and RND2, and performs close to optimal BCST.

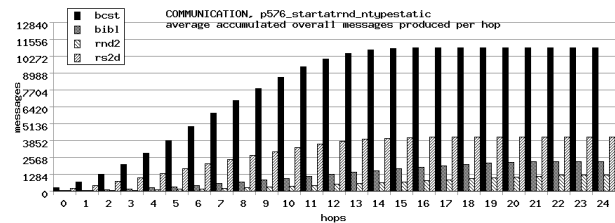


Figure 3. Experiment 1, Communication.

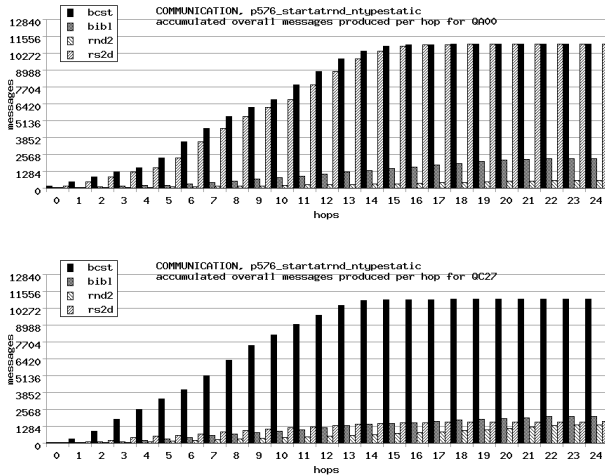
In more detail, RS2D performs as bad as broadcast in its initial training phase while in case of processing the last query of the test collection in one simulation run, RS2D outperforms even the more savvy BIBL system (see fig.4).

Please note that this provides evidence in favor of mixed risk-driven routing based on learned average Q/A behavior rather than query-specific routing only. It would be interesting to investigate a mix of both approaches in future.

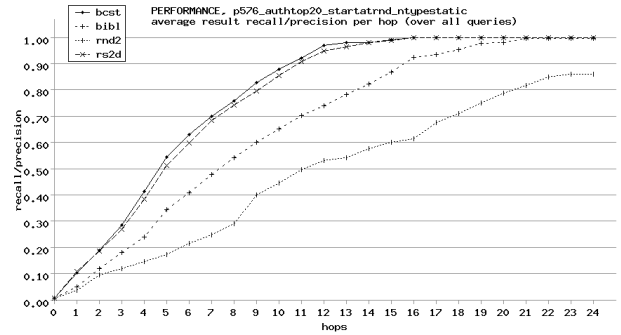
**Experiment 2:** We also simulated the case of single query authorities and random querying of peers. In this case, one peer hosts all services that are relevant to a specific query, thus possesses the complete relevance set of this particular query. For each query a different peer was chosen at random. Then the queries were executed uniformly at random from different peers as in the first experiment.

In this case, BIBL is more efficient than its competitors as it heavily benefits from the exploitation of the given semantic overlay structure for optimal routing. RS2D is outperformed by BIBL because it is difficult to find the authority for a query when only the *average* query answer behaviour is considered (see fig.5).

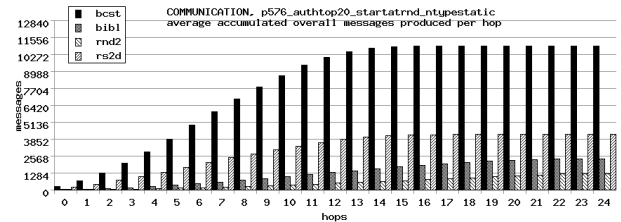
Not surprising, in this case the communication costs of RS2D are higher than that of BIBL with given semantic



**Figure 4. Experiment 1, Communication (first and last query):** While in the initial training phase RS2D produces as much traffic as BCST does, it even outruns BIBL’s traffic on the last executed query due to the learned average query answering behaviour. BIBL is more efficient in communication in the first run because of its exploitation of given semantic overlay knowledge for routing.



**Figure 5. Experiment 2, Precision:** RS2D still is almost optimal. BIBL exploits its given semantic topology to its maximum. Both outperform RND2.



**Figure 6. Experiment 2, Communication.**

topology but still significantly lower than BCST as shown in figure 6.

**Experiment 3:** In case where one centrally located peer executes all 36 queries for the user, thereby acting as a single point of entry, with 576 services distributed uniformly at random in a 576 peer network, and  $\theta_{TS} = 1$ , RS2D performed as well as BCST in terms of precision (BCST-/RS2D curves are overlapping in fig.7) but drastically reduced communication overhead.

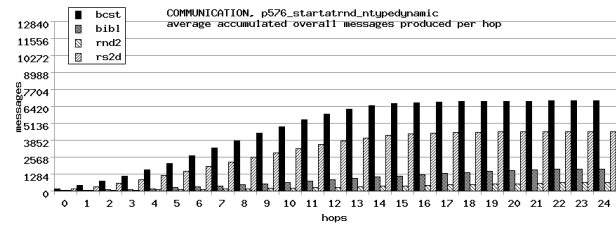
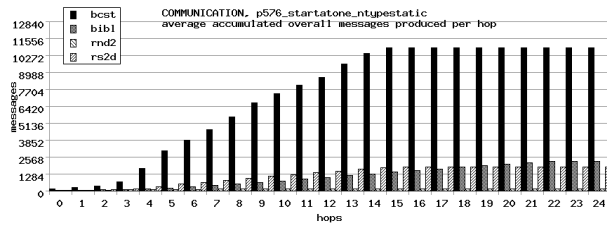
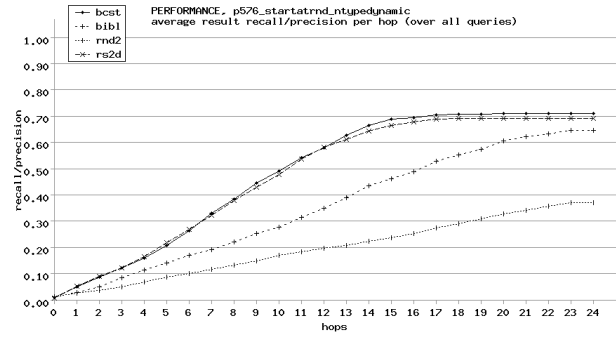
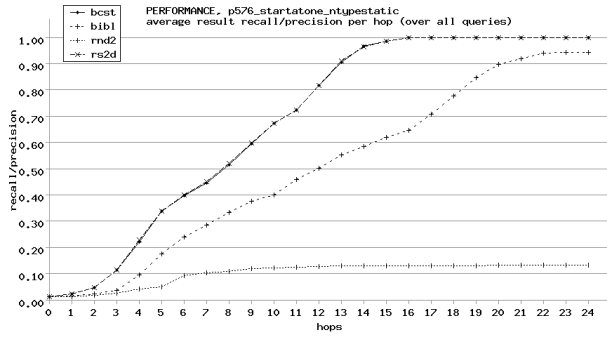
## 5.2 Robustness

The remaining question is how robust RS2D enabled unstructured P2P service networks are against dynamic changes, when peers can enter or leave the network at any time. For this purpose, we tested RS2D in a 576 peer network where each peer is hosting exactly one service but with only 80% of all the peers (= 460) being online. During the simulation run, we randomly let new peers join and leave the network with a rate of about one peer joining/leaving each 5 simulation steps (about 400 join/leave-operations per simulation run). In case of incomplete return paths for a query due to relevant peers having left the network, the peer in question tries to find the subsequent peers in the path. If this strategy fails, it sends out a limited 2-hop

broadcast of the answer to its neighbours. If this last fallback also fails, the answer to the query is discarded yielding a total loss of all related intermediate results.

The join operation for a single peer in RS2D enabled P2P networks is implemented as a simple handshake-advertisement: Each peer that wants to join the network, broadcasts a one-hop advertisement to all peers in its direct (one-hop) neighbourhood and then waits for acknowledgement-messages. If at least one peer answers, the requesting peer considers itself to be online, and both peers mutually take themselves for new routing decisions into account. The leave operation is completely passive: A peer just drops out and stops answering to messages. Each of its neighbouring peers will detect this as soon as it wants to send a new message to the dropped peer, and removes all training records that relate to this peer from the local training set.

As shown in figure 8, RS2D turned out to be reasonably robust against dynamic changes in the network topology. However, the precision went down for all tested routing mechanisms due to the following reasons. First, some of the relevant services are provided by offline peers, hence



**Figure 7. Experiment 3, precision: RS2D performs optimal (curve is on that of BCST). Central peer searches minimal spanning tree for all queries after initial multicast; Communication: RS2D outperforms BIBL with significantly lower communication effort.**

**Figure 8. Experiment 4, dynamic: Precision goes down for all routing methods but RS2D significantly outruns both Bibster and random selection, hence is more robust to network-fluctuation. However, this comes at the very expense of communication overhead compared to Bibster, though half of that of Broadcast.**

were unreachable. Second, some query answers were not propagated to the querying peer due to broken links in the network. Please note that the precision of BCST is optimal for this scenario, and RS2D is close to it but with only half of its communication efforts. This is because the initial training phase is only repeated for recently joined peers - all stable peers are still risk-evaluated when taking the forward-decision.

Not surprising, Bibster-like routing performs poor in dynamic environments since its semantic topology breaks to the same extent the network topology changes. Building up a semantic topology is a very costly process as each peer has to advertise its hosted services at the cost of one advertisement message propagated over 3 hops in our experiments leading to a traffic load of about 212.000 messages in a 576 peer network, and about 5.800 messages for each of the 36 queries.

For more details on the RS2D performance and robustness experiments, we refer the interested reader to the RS2D experiment web page [3].

## 6 Implementation

The RS2D approach to OWL-S service retrieval in unstructured P2P networks has been implemented in Java 1.5, and evaluated by simulation on one server. The architecture of the simulator is shown in figure 9; the simulator also provides PHP-script based online visualization of the experimental evaluation results.

For computing the numeric semantic score  $LS(S_q)$  used by RS2D for its risk based routing decision (cf. section 3), we defined a simple linear mapping  $(\sigma(s, q))$  of the output of the semantic Web service matchmaker OWLS-MX [9] to the interval  $[0, 1]$  as shown in figure 10.

The OWLS-MX code is available at [7]. OWLS-MX takes any OWL-S service description as a query, and returns an ordered set of relevant services that match the query in terms of both crisp logic based and syntactic similarity according to five different filters and selected IR similarity metric. Logical subsumption failures produced by the integrated description logic reasoner of OWLS-MX are tolerated, if the computed syntactic similarity value is sufficient. What makes OWLS-MX particularly suitable to ser-

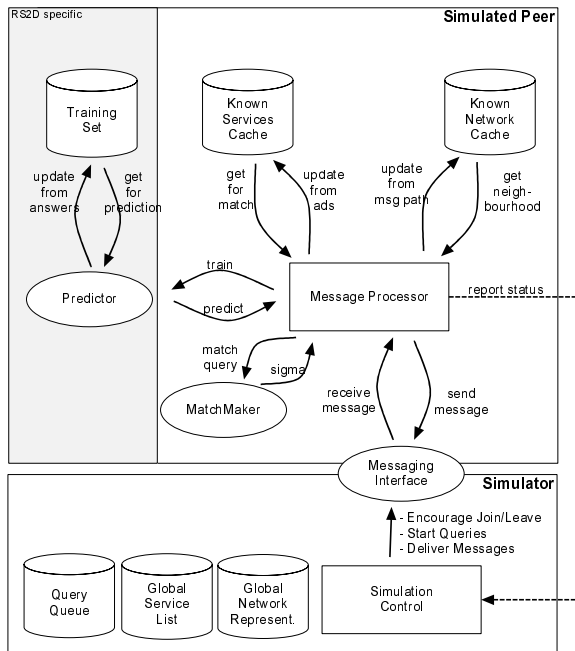


Figure 9. Architecture of RS2D Simulator.

vice retrieval in unstructured semantic P2P service networks is its capability to dynamically maintain a local ontology, hence renders RS2D independent from the use of any fixed global ontology like in GSD. It classifies arbitrary query concepts into its dynamically evolving ontology based on a commonly shared minimal basic vocabulary of primitive components instead of limiting query I/O concepts to terminologically equivalent service I/O concepts in a shared static ontology as, for example, the OWLS-UDDI matchmaker does.

For our experiments, we used the OWL-S service retrieval test collection OWLS-TC which is available at [6]. The collection consists of 576 OWL-S 1.1 services; its size in particular limited the maximum size of the unstructured P2P networks of our experiments as we neither did extend the collection nor distributed dummy service copies to peers for simulation.

## 7 Conclusion

We presented a first approach, named RS2D, to risk assessment driven semantic service retrieval in unstructured P2P networks without prior knowledge on the semantic overlay. It relies on the dynamic learning of averaged query-answer behavior of peers for minimal mixed routing risk. Experimental evaluation of RS2D showed that it is very robust and fast with reasonably high precision compared to

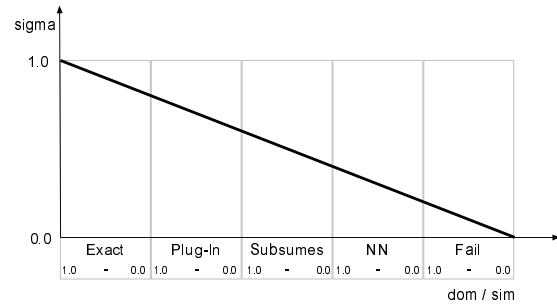


Figure 10. Used mapping of the degrees of semantic service matching returned by the OWLS-MX matchmaker to [0,1] for computing the numeric semantic score.

other existing relevant approaches. It is, however, weak in finding single query authority peers, and requires initial training, though only for an acceptable amount of time. We intend to make RS2D publicly available under GPL-like license at [semwebcentral.org](http://semwebcentral.org).

## References

- [1] OASIS Standard consortium. *Universal Description, Discovery and Integration (UDDI) protocol*. <http://www.uddi.org/>.
- [2] S. Androutsellis-Theotokis and D. Spinellis. *A survey of peer-to-peer content distribution technologies*. ACM Computing Surveys, 36(4):335-371, December 2004.
- [3] U. Basters. RS2D experimental evaluation results online: <http://www.basters.org/rs2d/>.
- [4] D. Chakraborty, A. Joshi, T. Finin, and Y. Yesha. *GSD: A novel groupbased service discovery protocol for MANETS*. 4th IEEE Conference on Mobile and Wireless Communications Networks (MWCN), 2002.
- [5] C. Elkan. *Naive Bayesian Learning*, [cite-seer.ist.psu.edu/30545.html](http://www.ist.psu.edu/30545.html).
- [6] B. Fries and M. Klusch. OWL-S service retrieval test collection OWLS-TC v2, download at <http://projects.semwebcentral.org/projects/owl-s-tc/>.
- [7] B. Fries and M. Klusch. OWLS-MX matchmaker source code, download at <http://projects.semwebcentral.org/projects/owl-s-mx/>.
- [8] P. Haase, R. Siebes, and F. van Harmelen. *Expertise-based Peer selection in Peer-to-Peer Networks*. Knowledge and Information Systems, Springer Verlag, 2006.
- [9] M. Klusch, B. Fries, and K. Sycara. *Automated Semantic Web Service Discovery with OWLS-MX*. Proc. 5th Intl. Conference on Autonomous Agents and Multiagent Systems (AAMAS), Hakodate, Japan. ACM Press., 2006.



**Semantic Service Composition**





---

## Introduction

Semantic Web service composition is the act of taking several semantically annotated component services, and bundling them together to meet the needs of a given customer. Automating this process is desirable to improve speed and efficiency of customer response, and, in the semantic Web, supported by the formal grounding of service and data annotations in logics. The following very brief introduction to Semantic Web service composition planning with comments on its interrelation to Semantic Web service discovery, open problems and further readings are taken from the comprehensive book chapter on semantic service coordination (Klusch, 2008a)[196].

### Web Service Composition

In general, Web service composition is similar to the composition of workflows such that existing techniques for workflow pattern generation, composition, and management can be partially reused for this purpose (Henocque & Kleiner, 2007)[153]. Typically, the user has to specify an abstract workflow of the required composite Web service including both the set of nodes (desired services) and the control and data flow between these nodes of the workflow network. The concrete services instantiating these nodes are bound at runtime according to the abstract node descriptions, also-called "search recipes" (Casati & Shan, 2001)[58].

As mentioned in the first chapter, the mainstream approach to Web service composition is to have a single administrator responsible for (semi-)manually scripting such workflows (orchestration and choreography) between WSDL services of different business partners in BPEL (Papazoglou, 2007; Alonso et al., 2003)[284, 6]. This is largely motivated by industry to work for service composition in legally contracted business partner coalitions - in which there is, unlike in open service environment, only very limited need for automated service composition planning, if at all. Besides, neither WSDL nor BPEL nor any other workflow languages like UML2 or YAWL have formal semantics which would allow for an automated logic-based composition. This is differ-

ent with Semantic Web service composition. In fact, the majority of Semantic Web service composition tools draws its inspiration from the vast literature on logic-based AI planning (Peer, 2005)[290].

### **AI-Planning-Based Web Service Composition**

The service composition problem roughly corresponds to the state-based planning problem  $(I, A, G)$  in AI to devise a sound, complete, and executable plan which satisfies a given goal state  $G$  by executing a sequence of services as actions in  $A$  from a given initial world state  $I$ . Classical AI planning-based (planning from first principles) composition focuses on the description of services as merely deterministic state transitions (actions) with preconditions and state altering (physical) effects. Actions are applicable to actual world states based on the evaluation of preconditions and yield new (simulated) states where the effects are valid. Further, classical AI planning is performed under the assumption of a closed world with complete, fully observable initial (world) state. This is not necessarily appropriate for service composition planning in the dynamic and open-ended Semantic Web (Srivastava & Koehler, 2003)[344]. However, all existing SWS composition planners are closed-world planners of which some are able to cope with uncertainties about the domain. A given logical goal expression and set of logic-based definitions of semantic service signature (I/O) concepts together with logic preconditions and effects from a DL-based ontology (domain or background theory) can be converted into one declarative (FOL) description of the planning domain and problem - which can serve a given logic-based AI planner as input. In particular, service outputs are encoded as special non-state altering knowledge effects, and inputs as special preconditions. The standard target language for the conversion is PDDL (Planning Domain Description Language) but alternative representation formalisms are, for example, the situation calculus [271], linear logic [309], high-level logic programming languages based on this calculus like GOLOG [252], Petri nets, or HTN planning task and method description format[338].

In the following, we classify existing Semantic Web service composition planners and comment on the principled interrelation between composition, discovery, and execution. Approaches to interleaved composition and negotiation are discussed in the introduction to part four. Please note that the set of presented examples of Semantic Web service composition planners is representative but not exhaustive.

### **Classification of Semantic Web Service Composition Planners**

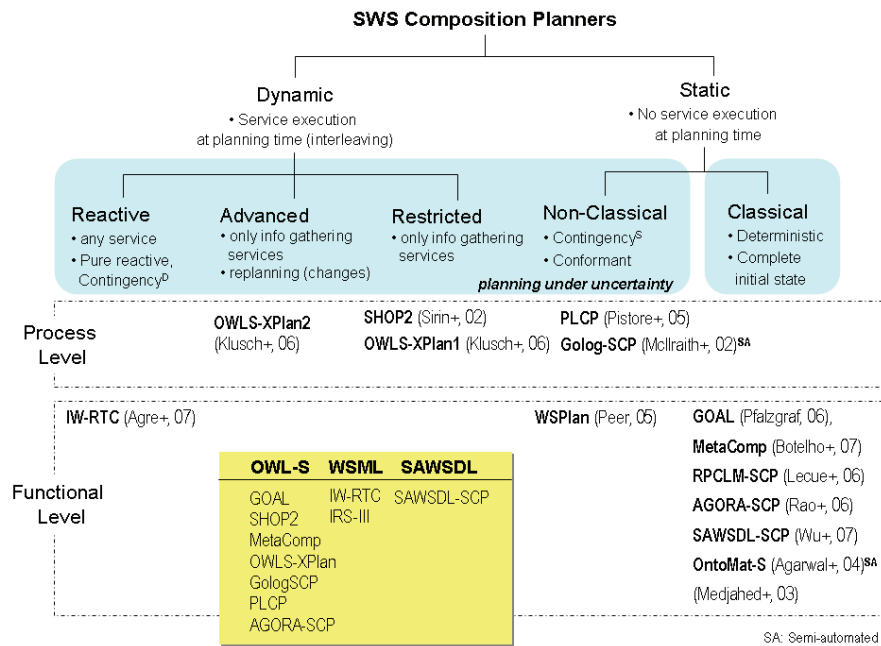
In general, any AI-planning framework for Semantic Web service composition can be characterized by

- the kind of representation of the planning domain and problem to allow for automated reasoning on actions and states,
- the planning method applied to solve the given service composition problem in the domain, and
- the parts of service semantic descriptions that are used for this purpose.

We can classify existing Semantic Web service composition planners according to the latter two criteria, which yields the following classes.

- Dynamic or static Semantic Web service composition planners depending on whether the plan generation and execution are inherently interleaved in the sense that actions (services) can be executed at planning time, or not.
- Functional-level or process-level Semantic Web service composition planners depending on whether the plan generation relies on the service profile semantics only, or the process model semantics in addition (data and control flow) [229].

Figure 7.1 shows representative examples of implemented Semantic Web service composition planners for each of these categories.



**Fig. 7.1.** Classes of Semantic Web service composition planners.

### *Static and dynamic composition planning*

The majority of Semantic Web service composition planners such as GOAL (Pfalzgraf, 2006)[294], MetaComp (Botelho et al., 2007), PLCP (Pistore et al., 2005), RPCLM-SCP (Lecue & Leger, 2006)[229] and AGORA-SCP (Rao et al., 2006)[309] are static classical planners. Approaches to dynamic composition planning with different degrees of interleaving plan generation and execution are rare. Unlike the static case, restricted dynamic composition planners allow the execution of information gathering but no world state altering services, hence are capable of planning under uncertainty about action outcomes at planning time. Examples of such composition planners are SHOP2 (Sirin et al., 2002) [339, 338], GOLOG-SCP (McIlraith et al., 2002)[252] and OWLS-XPlan1 (Klusck et al., 2005)[206].

Advanced and reactive dynamic composition planners in stochastic domains even take non-deterministic world state changes into account during planning. While advanced dynamic planners like OWLS-XPlan2 (Klusck & Renner, 2006)[210] are capable of heuristic replanning subject to partially observed (but not caused) state changes that affect the current plan at planning time, their reactive counterparts like INFRAWESBS-RTC (Agre & Marinova, 2007)[3] fully interleave their plan generation and execution in the fashion of dynamic contingency and real-time planning.

### *Functional- and process-level composition planning*

As shown in figure 7.1, most Semantic Web service composition planners perform functional-level, also called service profile-based, composition (FLC) planning. FLC planning considers services as atomic or composite black-box actions which functionality can solely be described in terms of their inputs, outputs, preconditions, and effects, and which can be executed in a simple request-response without interaction patterns. Examples of FLC planners are GOAL (Pfalzgraf, 2006)[294], SAWSDL-SCP (Wu et al., 2007)[386] and OntoMat-S (Agarwal et al., 2004)[4].

Process-level composition (PLC) planning extends FLC planning in the sense that it also takes the internal complex behavior of existing services into account. Prominent examples are SHOP2 (Sirin et al., 2004)[338], PLCP (Giunchiglia & Traverso, 1999; Pistore et al., 2001, 2005) [295, 296] and OWLS-XPlan (Klusck et al., 2005, 2006)[206, 210]. Both kinds of composition planning perform, in particular, semantic service profile or process matching which is either inherent to the underlying planning mechanism, or achieved by a connected stand-alone Semantic Web service matchmaker. We will discuss the interrelation between composition and semantic matching later.

### *Support of Semantic Web service description frameworks*

Remarkably, most implemented Semantic Web service composition planners support OWL-S like GOAL, OWLS-XPlan, SHOP2, GologSCP and MetaComp, while there is considerably less support of the standard SAWSDL

and WSML available to date. In fact, the SAWSDL-SCP planner (Wu et al., 2006)[386] is the only one for SAWSDL, while the IW-RTC planner (Agre & Marinova, 2007)[3] is, apart from the semi-automated orchestration of WSML services in IRS-III, the only fully automated FLC planner for WSML yet.

Most composition planner feature an integrated conversion of Semantic Web services, goals and ontologies into the internally used format of the planning domain and problem description, though a few others like the framework WSPlan (Peer, 2005)[291] for static PDDL-based planning under uncertainty, and the recursive, progression-based causal-link matrix composition planner RPCLM-SCP (Lecue & Leger, 2006)[229] do not.

In the following, we discuss each of the above mentioned categories and selected examples of Semantic Web service composition planners in more detail.

### **Functional-Level Semantic Web Service Composition Planners**

Intuitively, FLC planning generates a sequence of Semantic Web services based on their profiles that exact or plug-in matches with the desired (goal) service. In particular, existing services  $S_i, S_{i+1}$  are chained in this plan such that the output of  $S_i$  matches with the input of  $S_{i+1}$ , while the preconditions of  $S_{i+1}$  are satisfied in the world state after execution of  $S_i$ . Depending on the considered Semantic Web service description format (cf. chapter 3), different approaches to logic based, non-logic-based or hybrid semantic service profile IOPE matching are available for this purpose (cf. figure 3.13).

In order to automatically search for a solution to the composition problem, FLC planners can exploit different AI planning techniques with inherent logic-based semantic profile IOPE or PE matching like WSPlan (Peer, 2005), respectively, MetaComp (Botelho et al., 2006). The recursive forward-search planner GOAL (Pfalzgraf, 2006) as well as the SAWSDL-SCP (Wu et al., 2007) apply non-logic-based semantic profile IO matching of OWL-S, respectively, SAWSDL services.

In AGORA-SCP (Rao et al., 2006), theorem proving with hybrid semantic profile IO matching is performed for OWL-S service composition: Both services and a request (theorem) are described in linear logic, related to classical FOL, while the SNARK theorem prover is used to prove that the request can be deduced from the set of services. The service composition plan then is extracted from the constructive proof.

The FLC planner in (Medjahed, 2003)[255] uses proprietary composability rules for generating all possible plans of hybrid semantic IO-matching services in a specific description format (CSSL). From these plans the requester has to select the one of best quality (QoS).

### **Process-Level Semantic Web Service Composition Planners**

Though FLC planning methods can address conditional outputs and effects of composite services with dynamic planning under uncertainty, considering

services as black-boxes does not allow them to take the internal complex service behaviour into account at planning time. Such behavior is usually described as subservice interactions by means of control constructs including conditional and iterative steps. This is the domain of process level composition (PLC) planning that extends FLC planning in the aforementioned sense. However, only few approaches to process-level composition planning for Semantic Web services exist to date. For example, orchestration of WSML services in IRS-III (Domingue et al., 2005)[101] synthesizes abstract state machines to compose individual services in a given process flow defined in OCML<sup>1</sup>. Though, the functionality of the WSMX orchestration unit has not been completely defined yet.

Other automated PLC planners of OWL-S services exploit different AI planning techniques such as

- HTN (Hierarchical Task Network) planning of OWL-S process models converted to HTN methods like in SHOP2 (Sirin et al., 2004)[338],
- Neo-classical GRAPHPLAN-based planning mixed with HTN planning of OWL-S services converted to PDDL in OWLS-XPlan (Klusch et al., 2005, 2006)[206, 210],
- Value-based synthesis of OWL-S process models in a given plan template of situation calculus-based GOLOG programs (McIlraith et al., 2002)[252, 271],
- Planning as model checking of OWL-S process models converted to equivalent state transition systems (STS) in the PLCP (Giunchiglia & Traverso, 1999; Pistore et al., 2001, 2005) [295, 296].

In the following, we discuss each class of static and dynamic Semantic Web service composition planners together with selected examples.

### Static Semantic Web Service Composition Planners

The class of static AI planning-based composition covers approaches to both classical and non-classical planning under uncertainty.

#### *Static classical planning*

As mentioned above, classical AI planners perform (off-line) planning under the assumption of a closed, perfect world with deterministic actions and a complete initial state of a fully observable domain at design time. For example, Graphplan is a prominent classical AI planning algorithm that first performs a reachability analysis by constructing a plan graph, and then performs logic-based goal regression within this graph to find a plan that satisfies the goal. Classical AI planners are static since their plan generation and execution is strictly decoupled.

---

<sup>1</sup> [kmi.open.ac.uk/projects/ocml/](http://kmi.open.ac.uk/projects/ocml/)

One example of a static classical Semantic Web service composition planner is GOAL (Pfalzgraf, 2006)[294] developed in the SmartWeb project. GOAL composes extended OWL-S services by means of a classical recursive forward-search (Ghallab et al., 2004)[130]. Both, the initial state and the goal state are derived from the semantic representation of the user's question (goal) obtained by a multimodal dialogue system in SmartWeb. At each stage of the planning process the set of services which input parameters are applicable to the current state is determined by signature (IO) matching through polynomial subgraph isomorphism checking (Messmer, 1995)[253]: The instance patterns of input parameters are matched against the graph representation of the state, and a service is applied to a plan state (simulated world state) by merging the instance patterns of its output parameters with the state.

As a result, GOAL does not exploit any logical concept reasoning but structural service I/O graph matching to compose services. If plan generation fails, GOAL detects non-matching paths within instance patterns and consequently produces a clarification request (ako information gathering service) conveyed to the user by the dialogue system; on response by the user the planning process is restarted in total.

Static service composition in the AGORA-SCP service composition system (Rao et al., 2006)[309] relies on linear logic (LL) theorem proving. The profiles of available DAML-S services are translated in to a set of LL axioms, and the service request is formulated as a LL theorem to be proven over this set. In case of success, the composition plan can be extracted from the proof, transformed to a DAML-S process model and executed as a BPEL script. The AGORA planner is the only approach to decentralized composition planning in structured P2P networks (Küngas & Matskin, 2006)[223].

An example of a static classical Semantic Web service composition planner based on a special logic-based PDDL planner is MetaComp (Botelho et al., 2007).

#### *Static planning under uncertainty*

Work on planning under uncertainty in AI is usually classified according to (a) the representation of uncertainty, that is whether uncertainty is modeled strictly logically, using disjunctions, or is modeled numerically (e.g. with probabilities), and (b) observability assumptions, that is whether the uncertain outcomes of actions are not observable via sensing actions (conformant planning); partially or fully observable via sensing actions (conditional or contingency planning) [93]. As mentioned above, we can have uncertainty in the initial states and in the outcome of action execution. Since the observation associated to a given state is not unique, it is also possible to model noisy sensing and lack of information. Information on action outcomes or state changes that affect the plan can be gathered either at planning time (dynamic) or

thereafter (static) for replanning purposes.

**Static conditional or contingency planning.** Static conditional or contingency planners like Cassandra (Pryor & Collins, 1996) and DTPOP (Peot 1998) devise a plan that accounts for each possible contingency that may arise in the planning domain. This corresponds to an optimal Markov policy in the POMDP framework for planning under uncertainty with probabilities, costs and rewards over a finite horizon (Puterman, 1994). The contingency planner anticipates unexpected or uncertain outcomes of actions and events by means of planned sensing actions, and attempts to establish the goals for each different outcome of these actions through conditional branching of the plan in advance<sup>2</sup>. The plan execution is driven by the outcome of the integrated sensing subplans for conditional plan branches, and decoupled from its generation which classifies these planners as static.

**Static conformant planning.** Conformant planners like the Conformant-FF (Hoffmann and Brafman, 2007), Buridan (Kushmerick et al., 1995), and UDTPOP (Peot 1998) perform contingency planning without sensing actions. The problem of conformant planning to search for the best unconditional sequence of actions under uncertainty of initial state and action outcome can be formalized as fully non-observable MDP, as a particular case of POMDP, with a search space pruned by ignoring state observations in contingency planning. For example, conformant Graphplan planning (CGP) [340] expresses the uncertainty in the initial state as a set of completely specified possible worlds, and generates a plan graph for each of these possible worlds in parallel. For actions with uncertain outcomes the number of possible worlds is multiplied by the number of possible outcomes of the action. It then performs a regression (backward) search on them for a plan that satisfies the goal in all possible worlds which ensures that the plan can be executed without any sensory actions. Conformant planners are static in the sense that no action is executed at planning time.

*Examples of static Semantic Web service composition planners under uncertainty*

The PLCP (Pistore et al., 2005)[297, 296] performs static PLC planning under uncertainty for OWL-S services. OWL-S service signatures and process models together with a given goal are converted to non-deterministic and partially observable state transition systems (STS) which are composed by a symbolic

---

<sup>2</sup> Examples of decision criteria according to which contingency branches are inserted in the (conventional) plan, and what the branch conditions should be at these points, are the maximum probability of failure, and the maximum expected future reward (utility) as a function of, for example, time and resource consumption. Uncertainty is often characterized by probability distributions over the possible values of planning domain predicates.



model checking-based planner (MBP)[295] to a new STS which implements the desired composed service. In other words, possible plans modeled as finite STSs (sequences of applicable actions in the state space) are verified against the goal specification. This STS eventually gets transformed to an executable service composition plan (in BPEL) with possible conditional and iterative behaviors. No action is executed at planning time, and uncertainty is resolved by sensing actions during plan execution.

An example of static FLC planning under uncertainty is the WSPlan framework (Peer, 2005)[291] which provides the user with the option to plug in his own PDDL-based planner and to statically interleave planning (under uncertainty) with plan execution. Static interleaving refers to the cycle of plan generation, plan execution, and replanning based on the result of the executed sensing subplans (in the fashion of static conditional planning) until a sequential plan without sensing actions is generated that satisfies the goal. There are no static classical PLC planner for Semantic Web services with deterministic (sequential) process models of composite services only available.

### Dynamic Semantic Web Service Composition Planners

The class of dynamic AI planning-based composition covers approaches to restricted, advanced and reactive dynamic planning under uncertainty.

#### *Restricted dynamic planning*

Dynamic planning methods allow agents to inherently interleave plan generation and execution. In restricted dynamic planning, action execution at planning time is strictly restricted to information gathering about uncertain action outcomes of services. These special actions (also called book-keeping services or callbacks) add new knowledge in form of ground facts to the partial observable initial state under the so-called IRP (Invocation and Reasonable Persistence) assumption[252] to ensure conflict avoidance<sup>3</sup> only. The incremental execution of callbacks under IRP assumption during planning has the same effect when executing them prior to planning in order to complete the initial state for closed world planning. Like in classical planning, however, world state altering services with physical effects (in opposite to the so-called knowledge effects of service outputs) are only simulated in local planning states and never get executed at planning time.

#### *Examples of restricted dynamic Semantic Web service composition planners*

Prominent examples of restricted dynamic composition planners are SHOP2, and OWLS-XPlan1 (Klusch et al., 2005)[206] for OWL-S. SHOP2 (Sirin et al.,

---

<sup>3</sup> The IRP assumption states that (a) the information gathered by invoking the service once cannot be changed by external or subsequent actions, and (b) remains the same for repeating the same call during planning.

2003, 2004)[339, 337] converts given OWL-S service process models into HTN methods and applies HTN planning interleaved with execution of information-gathering actions (callbacks) to compose a sequence of services that satisfies the given task. By mapping any OWL-S process model to a situation calculus-based GOLOG program, the authors prove that the plans produced are correct in the sense that they are equivalent to the action sequences found in situation calculus. HTN planning is correct and complete but undecidable due to possibly infinite recursive decomposition of given methods to executable atomic tasks. SHOP2 detects and breaks decomposition cycles. OWLS-XPlan1 will be described in chapter 8.

#### *Advanced dynamic planning*

Advanced dynamic planning methods allow in addition to react on arbitrary changes in the world state that may affect the current plan already during planning such as in OWLS-XPlan2. This is in contrast to static planning under uncertainty where sensing subplans of a plan are executed at run time only. However, in both restricted and advanced dynamic planning the interleaved execution of planning with world state altering services is prohibited to prevent obvious problems of planning inconsistencies and conflicts.

#### *Examples of advanced dynamic Semantic Web service composition planners*

To the best of our knowledge, OWLS-XPlan2 (Klusch & Renner, 2006)[210] is the only one implemented example of an advanced dynamic composition planner. OWLS-XPlan2 will be described in chapter 9.

#### *Reactive dynamic planning*

Finally, reactive dynamic planning like in Brooks's subsumption architecture, RETE-based production rule planners, and the symbolic model checking based planner SyPEM (Bertoli et al., 2004)[33] allows the execution of arbitrary actions at planning time. Pure reactive planner produce a set of condition-action (if-then) or reaction rules for every possible situation that may be encountered, whether or not the circumstances that would lead to it can be envisaged or predicted. The inherently interleaved planning and execution is driven through the evaluation of state conditions at every single plan step to select the relevant if-then reaction rule and the immediate execution of the respective, possibly world state altering action; This cycle is repeated until the goal is hopefully reached.

A variant of reactive dynamic planning is dynamic contingency planning like in XII (Golden, Etzioni & Weld, 1994) and SAGE (Knoblock, 1995). In this case, a plan that is specified up to the information-gathering steps gets executed to that stage, and, once the information has been gathered, the rest of the plan is constructed. Interleaving planning and execution this way has the advantage that it is not necessary to plan for contingencies that do not actually arise.

In contrast to pure reactive planners, reasoning is only performed at branch points predicted to be possible or likely.

In any case, reactive dynamic planning comes at the possible cost of plan optimality, and even plan existence, that is suboptimality and dead-end action planning or failure (Olawsky & Gini, 1990). The related ramification problem<sup>4</sup> is usually addressed either by restrictive assumptions on the nature of service effects on previous planning states (Bertoli et al., 2004)[33] in safely explorable domains (Koenig and Simmons 1995; 1998; Koenig 2001), or by integrated belief revision (TMS) in the planners knowledge base at severe computational costs.

#### *Examples of reactive dynamic Semantic Web service composition planners*

One example of an implemented reactive dynamic composition planner is the real-time composition planner IW-RTC (Agre & Marinova, 2007) developed in the European research project INFRAWEB. It successively composes pairs of keyword-based IO matching services, executes them and proceeds with planning until the given goal is reached. Unfortunately, the authors do not provide any detailed description of the composition and matching process nor complexity analysis.

#### *Problems of Semantic Web service composition planning under uncertainty*

One problem with adopting planning under uncertainty for service composition is that the execution of information gathering (book keeping) or even world state altering services at design or planning time might not be charge free, if granted by providers at all. That is, the planning agent might produce significant costs for its users even without any return value in case of plan generation or execution failure. Another problem is the known insufficient scalability of conditional or conformant planning methods to planning domains at Web scale or business application environments with potentially hundreds of thousands of services and vast instance bases. Research on exploiting conditional or conformant planning methods for Semantic Web service composition has just started.

### **FLC Planning of Monolithic DL-Based Services**

Research on FLC planning with monolithic DL-based descriptions of services has just started. Intuitively, the corresponding plan existence problem for the composition of such services is as follows. Given an acyclic TBox  $T$  describing the domain or background theory in a DL, ABoxes  $S$  and  $G$  which interpretations  $I$  (consistent wrt  $T$ ) over infinite sets of individual (object) names are

---

<sup>4</sup> The problem of ensuring the consistency of the planners knowledge base and the reachability of the original goal in spite of (highly frequent) world state altering service execution during plan generation.

describing, respectively the initial and goal state, and a set  $A$  of operators describing deterministic, parameterized actions  $\alpha$  which precondition and effects are specified in the same DL and transform given interpretations of concepts and roles in  $T$  ( $I \rightarrow_{\alpha}^T I'$ ), is there a sequence of actions (consistent with  $T$ )<sup>5</sup> obtained by instantiating operators with individuals which transforms  $S$  into  $G$ ?

It has been shown in (Baader et al., 2005)[20] that the standard reasoning problems on actions, that are executability<sup>6</sup> and projection<sup>7</sup>, are decidable for description logics between ALC and ALCOIQ. Furthermore, it has been shown in (Milicic, 2007)[257] only recently that the plan existence problem for such actions in ALCOIQ is co-NEXPTIME decidable for finite sets of individuals used to instantiate the actions, while it is known to be PSPACE-complete for propositional STRIPS-style actions. In addition, the extended plan existence problem for actions specified in  $ALC_U$  (with universal role  $U$  for assertions over the whole domain with infinitely countable set of individuals) was proven undecidable by reduction to the halting problem of deterministic Turing machines.

However, there is no implemented composition planner for monolithic DL-based services available to date.

## Interrelations

In the following, we briefly comment on the principled relations between semantic service composition planning, discovery, and execution. Selected approaches to interleaved semantic service composition planning with negotiation are presented in the introduction to the next part of this thesis.

### *Semantic Web service composition planning and discovery*

From the view of semantic service discovery, the composition of complex services is of importance if no available service satisfies the given request. In this case, the matchmaker or requester agent can interact with a composition planner to successfully generate a composite service that eventually satisfies the query.

On the other hand, semantic service composition planning agents require a description of the planning domain and goal to start their planning. Both can be semi-automatically generated from the set of available semantic service descriptions together with related logic-based ontologies, the so-called

---

<sup>5</sup> An action is consistent with TBox  $T$ , if for every model  $I$  of  $T$  there exists  $I'$  such that  $I \rightarrow_{\alpha}^T I'$ .

<sup>6</sup> Action executability is equal to the satisfaction of action preconditions in given world states:  $I \models pre_1, \forall i, 1 \leq i \leq n, I'.I \rightarrow_{\alpha_1 \dots \alpha_i}^T I' : I' \models pre_{i+1}$ .

<sup>7</sup> Satisfaction of assertion  $\phi$  as a consequence or conjunctive effect of applying actions to a given state: For all models  $I$  of  $S$  and  $T, I'.I \rightarrow_{\alpha_1 \dots \alpha_n}^T I' : I' \models \phi$

background theories. In fact, from the view of composition planning, semantic service discovery is of importance for the following reasons: A semantic service matchmaker can be used to

- Prune the initial search space of the composition planner with respect to given application-specific preferences of available services, and
- Select semantically equivalent or plug-in, and execution compatible services during planning as alternative (substitute) services in case of planning failures (replanning).

There is no agreed-upon strategy for pruning the search space of Semantic Web service composition planners. Such pre-filtering of services by a matchmaker can be heuristically performed against non-functional and functional service semantics in order to speed up the corresponding planning process - but at the cost of its incompleteness. That is, composition planning over heuristically pruned search space does not, in general, solve the plan existence problem.

Another source of correct but incomplete composition planning is the naive interleaving of planning with semantic service matching. For example, the sequential composition of stateful services from a given initial state by consecutive calls of a logic-based semantic service matchmaker by the planner only does not guarantee to find a solution if it exists: Any (not specifically planning-oriented) matchmaker usually

- does not maintain any planning state information, thus ignores variable bindings that hold for service signatures (IO) and specifications (PE) according to the actual state reached by the calling (closed-world) planner, and
- performs pairwise service matching only, hence would not return services to the calling planner which combined effects (even with provided state-based instantiation) would eventually lead to a solution.

To the best of our knowledge, all available Semantic Web service matchmakers (cf. introduction to part two) are implemented as a stand-alone tool for mere semantic service matching without any composition planning support. However, functional-level composition planning is a kind of state-based semantic plug-in matching of the generated service plan with the given goal: Any FLC-planner generates a sequence of Semantic Web services based on their profiles that exact or plug-in matches with the desired (goal) service, whereas for each consecutive pair of planned services  $S$  and  $S'$  the output of  $S$  semantically matches with the input of  $S'$ , and the preconditions of  $S'$  are satisfied in the planning state including the effects of  $S$ .

#### *Examples of interleaved Semantic Web service matching and composition planning*

There are only a few approaches implemented that explicitly interleave semantic matching with composition planning. These composition planners ex-

plicitly interact with matchmaking modules, and/or the user (semi-automated composition) during planning.

In (Lecue et al., 2007)[230], logic-based service matching is extended with concept abduction to provide explanations of mismatches between pairs of service profiles that are iteratively used as constructive feedback during composition when searching for alternative services to bridge identified semantic gaps between considered IOPE profiles of services in the current plan step. A similar abduction-based matchmaking approach is presented in [99]. This scenario of explicitly interleaved discovery and composition has been implemented and tested in a non-public France Telecom research project.

In (Küstners et al., 2007)[224], the functional-level composition of services specified in the DIANE service description language DSD is explicitly integrated with a DSD matchmaker module that matches service requests asking for multiple connected effects of configurable services. By using a value propagation mechanism and a cut of possible (not actual) parameter value fillings for service descriptions that cover multiple effects the authors avoid exponential complexity for determining an optimal configuration of plug-in matching service advertisements used for a composition.

In (Binder et al., 2004)[36], the syntactic functional-level service composition is based on partial matching of numerically encoded service IO data types in a service directory. Unfortunately, the justification of the proposed numeric codings for matching services appears questionable, though it was shown to efficiently work for certain applications.

The composition planner OWLS-XPlan2 (Klusch et al., 2006)[310] integrates planning-specific service IOPE matching on the grounding level: At each plan step, the planner calls the component OWLS-MXP of the matchmaker OWLS-MX 1.1 to check the compatibility of XMLS types of input and output parameters of consecutive services. This ensures the principled executability of the generated sequential plan at the service grounding level in WSDL.

The interactive OWL-S service composer developed at UMBC (Sirin et al., 2004)[337] uses the OWLS-UDDI matchmaker to help users filter and select relevant services while building the composition plan. At each plan step, the composer provides the user with advertised services which signatures (IO) plug-in or exact match with that of the last service in the current plan. This leads to an incremental forward chaining of services which does not guarantee completeness without respective user intervention.

The Agora-P2P service composition system (Küngas & Matskin, 2006)[223] is the only approach to decentralized Semantic Web service composition planning. It uses a Chord ring to publish and locate OWL-S service descriptions keyword-based while linear logic theorem proving and logic-based semantic service IO matching is applied to compose (and therefore search for relevant subservices of) the desired service.

The semantic compatibility of subsequent services in a plan does not guarantee their correct execution in concrete terms on the grounding level. A plan is called correct, if it produces a state that satisfies the given goal (Lecue & Leger, 2006)[229]. The principled plan executability, also-called execution composability of a plan, requires the data flow between chained services of a plan to be ensured during plan execution on the service grounding level (Medjahed et al., 2003)[255]. This can be verified through complete (XMLS) message data type checking of semantically matching I/O parameters of every pair of subsequent services involved in the plan. For example, OWLS-XPlan2 calls a special matchmaker module (OWLS-MXP) that checks plan execution compatibility at each plan step during planning.

The consistent, central or decentral plan execution can be achieved by means of classical (distributed) transaction theory and systems. An implemented approach to distributed Semantic Web service composition plan execution is presented, for example, in [46, 265]. However, the availability of non-local services that are not owned by the planning agent can be, in principle, refused by autonomous service providers without any prior commitment at any time. This calls for effective replanning based on alternative semantic matching services delivered by a planning-specific matchmaker to the composition planner prior to, or during planning such as in OWLS-XPlan2.

### **The Contributions**

In the following two chapters, we present the dynamic OWL-S service composition planner OWLS-XPlan which comes in two variants: OWLS-XPlan1 performs restricted dynamic planning, while OWLS-XPlan2 is capable of advanced dynamic planning in stochastic environments. Both variants are available to the public at the software portal [semwebcentral.org](http://semwebcentral.org). Both variants have been successfully used in the European research project CASCOM and the national basic research project SCALLOPS for composing medical assistance services written in OWL-S for selected real-world use case scenarios in the e-health domain. These contributions are joint work with the Master students Marcus Schmidt and Kai-Uwe Renner, the software engineer Patrick Kapahnke, and the PhD students Andreas Gerber and Bastian of my research team at DFKI. For more details on OWLS-XPlan, I refer to the co-authored technical report [310].

*Chapter 8: OWL-S service composition planning with OWLS-XPlan.* In this chapter, we present the restricted dynamic functional-level composition planner OWLS-XPlan1 for OWL-S services. Its state-based AI planner XPlan1 is hybrid in the sense that it integrates the relaxed GraphPlan-based FF planner developed by Hoffmann and Nebel (2003)[156] with HTN (Hierarchical Task Network) planning of domain-specific complex actions. XPlan1 uses enforced

hill-climbing search to select the best helpful action for reaching the next state in the plan graph, that is an applicable action in the relaxed plan graph (no delete lists) with minimal heuristic goal distance, and applies breadth-first search otherwise to ensure completeness of planning. To start planning, the user describes an initial world and goal state in OWL-DL supported by given OWL-S service and respective ontologies. Both states are translated into PDDL 1.2 by the converter OWLS2PDDL 1.0 of OWLS-XPlan1, and then fed into its planner XPlan1 (supporting the ADL part of STRIPS) which generates a plan, if it exists (completeness).

Like SHOP2, XPlan1 allows the real execution of services via callbacks under IRP assumption at planning time (closed world) for services without any (world state altering) effects only. This ensures the correctness of plans generated by XPlan1 over deterministic states. It does not allow to cope with conditional effects of service outcomes (inCondition in OWL-S; if-then-else in PDDL) in partially observable world states by respective callbacks on output values. Sequenced planning states cannot be inconsistent by definition; any single state inconsistency caused by multiple services with contradictory effects in the state is inherently resolved by XPlan1 through serialization with heuristic choice (of one helpful service with minimal goal distance in the final plan sequence). Each generated plan is minimal and eventually translated by the converter of OWLS-XPlan1 to a composite OWL-S service with sequential process model. It has been shown by limited experimental evaluation that XPlan1 top performs compared to selected planners of the planning competition IPC3.

Other coauthored publications on OWLS-XPlan1 not included in this chapter are (Hutter et al., 2006; 2006b)[174, 175]. They describe the extension and use of OWLS-XPlan1 with an information flow analysis component that is checking whether the generated composition plan preserves the individual user data privacy based on given service policies and clearances.

*Chapter 9: Advanced dynamic OWL-S service composition with OWLS-XPlan 2.0.* OWLS-XPlan2 is an advanced dynamic composition planner since it takes world state changes into account at planning time. This is achieved through extending XPlan1 with a module that allows the resulting planner XPlan2 to perform a heuristic-based re-planning that is driven by periodic observation of state changing events. Such events are service unavailability and fact changes. In contrast to contingency planning or reactive planning, XPlan2 first checks whether a certain event affects the current plan, and heuristically determines the approximately optimal re-entry point for a partial restart of the actual planning process.

Dynamic planning by XPlan2 turned out to be particularly well suited to the application environment considered in the projects CASCOM and SCALLOPS. In these environments, any execution of services during planning is prohibited for reasons of costs and autonomy. Like with XPlan1, planning with XPlan2 is complete and correct. Besides, XPlan2 calls the matchmaker



module OWLS-MXP for checking the execution composability of subsequent services on their WSDL grounding level at each plan step. This ensures that the final composition plan is correctly executable in principle at the data flow level. Preliminary experiments show the efficiency of advanced dynamic planning over iterative full replanning.

While XPlan1 is much less expressive than XPlan2, it is in average faster by a magnitude. Though both variants do not scale well in general, they have been successfully used in small-scale application scenarios of the above mentioned e-health projects. An optimized version XPlan3 is planned that relies in part on the CONFORMANT-FF planner by Hoffmann and Brafman (2006)[155] which is similar to ConformantGraphPlan (Smith and Weld, 1998).

### **Open Problems**

Some major challenges of research and development in the domain of Semantic Web service composition are as follows.

- Scalable and resource efficient approaches to service composition planning under uncertainty coupled with semantic service selection at plan execution time in the extremely resource constrained environments like the so-called Internet of Things interlinking all kinds of computing devices without limit on the global scale.
- Efficient means of composition planning of Semantic Web services in unstructured or hybrid peer-to-peer and grid computing environments.
- Interleaving of service composition planning with service negotiation in competitive settings.
- Easy to use tools for the common user to support discovery, negotiation, composition and execution Semantic Web services in one framework for different Semantic Web service formats like the standard SAWSDL, OWLS, WSML, and SWSL (cf. chapter 3).



## Service Composition Planning with OWLS-XPlan1

M. Klusch, A. Gerber, M. Schmidt: Semantic Web Service Composition Planning with OWLS-XPlan. Proceedings of the 1st International AAI Fall Symposium on Agents and the Semantic Web, Arlington VA, USA, pages 55 - 62, AAAI Press, 2005

M. Klusch and A. Gerber: Fast Composition Planning of OWL-S Services and Application. Proceedings of the 4th IEEE European Conference on Web Services (ECOWS), Zurich, Switzerland, pages 181 - 190, IEEE CS Press, 2006

# Semantic Web Service Composition Planning with OWLS-Xplan\*

**Matthias Klusch, Andreas Gerber**

German Research Center for Artificial Intelligence,  
Stuhlsatzenhausweg 3, 66123 Saarbruecken, Germany,  
{klusch, agerber}@dfki.de

**Marcus Schmidt**

DiaLOGIKa GmbH,  
Albertstrasse, 66125 Saarbruecken, Germany  
Markus.Schmidt@dialogika.de

## Abstract

We present an OWL-S service composition planner, called OWLS-Xplan, that allows for fast and flexible composition of OWL-S services in the semantic Web. OWLS-Xplan converts OWL-S 1.1 services to equivalent problem and domain descriptions that are specified in the planning domain description language PDDL 2.1, and invokes an efficient AI planner Xplan to generate a service composition plan sequence that satisfies a given goal. Xplan extends an action based FastForward-planner with a HTN planning and re-planning component.

## Introduction

One of the striking advantages of web service technology is the fairly simple aggregation of complex services out of a library of simpler or even atomic services. The same is expected to hold for the domain of semantic web services such as those specified in WSMO or OWL-S. The composition of complex services at design time is a well-understood principle which is nowadays supported by many broadly available tools and other composition planners such as SHOP2.

Hierarchical task network (HTN) planners such as SHOP2 perform well in domains for which complete and detailed knowledge on at least partially hierarchically structured action execution patterns is available, such as, for example, in scenarios of rescue planning. In domains in which this is not the case, i.e., no concrete set of methods and decomposition rules that lead to an executable plan are provided, an HTN planner would not find the solution due to the fixed structure of hierarchical action decompositions stored in its database. That inherently limits the degree of quality of any HTN planner to that of its used methods that are created by human experts.

In contrast, action based planners are able to find a solution based on atomic actions as they are described in the methods, but without using the structure of the latter. Atomic actions can be combined in multiple ways to solve a given planning problem. But how to cope with planning problems that are in part hierarchically structured according

to decomposition rules and methods but not solvable exclusively by means of HTN planning?

For this purpose, we developed a hybrid AI planner Xplan (Schmidt 2005) which combines the benefits of both approaches by extending an efficient graph-plan based FastForward-planner with a HTN planning component. To use Xplan for semantic Web-Service composition, XPlan is complemented by a conversion tool that converts OWL-S 1.1 service descriptions to corresponding PDDL 2.1 descriptions that are used by Xplan as input to plan a service composition that satisfies a given goal. In contrast to HTN planners, Xplan always finds a solution if it exists in the action/state space over the space of possible plans, though the problem is NP-complete. Xplan also includes a re-planning component to flexibly react to changes in the world state during the composition planning process. Together the implementations of Xplan and OWLS2PDDL converter make up the semantic Web service composition planner OWLS-Xplan.

The remainder of this paper is structured as follows. Section 2 provides an overview of the OWLS-Xplan system architecture, followed by a brief description of the integrated converter module OWLS2PDDL in section 3. The core of OWLS-Xplan, the hybrid planner Xplan, is presented and compared with SHOP2 in section 4 and 5, respectively. We conclude in section 6.

## OWLS-Xplan Overview

OWLS-Xplan consists of several modules for preprocessing and planning. It takes a set of available OWL-S services, a domain description consisting of relevant OWL ontologies and a planning query as input, and returns a plan sequence of composed services that satisfies the query goal.

For this purpose, OWLS-Xplan first converts the domain ontology and service descriptions in OWL and OWL-S, respectively, to equivalent PDDL 2.1 problem and domain descriptions using its OWLS2PDDL converter. The domain description contains the definition of all types, predicates and actions, whereas the problem description includes all objects, the initial state, and the goal state. Both descriptions are then used by the AI planner Xplan to create a plan (representing one composed web service) in PDDL that solves the given problem in the actual domain and initial state. For reasons of convenience, we developed a XML dialect of PDDL,

\*This work has been supported by the German Ministry of Education and Research (BMBF 01-IW-D02-SCALLOPS), and the European Commission (FP6 IST-511632-CASCOM).  
Copyright © 2005, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

called PDDXML, that simplifies parsing, reading, and communicating PDDL descriptions using SOAP. Table 1 shows the corresponding notions of both the AI planning and semantic web service domain.

planning domain	semantic web service domain
(atomic) operator	service profile
(atomic) action	atomic web service, atomic process
complex action	service model
method	composed web service, workflow, composite process

Table 1: Correlating notions of the planning and semantic web service domain

An operator of the planning domain corresponds to a service profile in OWL-S: Both operator and profile, in essence, describe a pattern or template of how an action or web service as an instance should look like. A method is a special type of operator, that allows the user to describe workflows or composed web services. The planner may use methods as a hierarchical task network during its planning process.

### Converter OWLS2PDDL

The conversion of OWL-S 1.1 service descriptions to PDDXML requires not only the straight forward transcription of types and properties to PDDL predicates but the mapping of services to actions (cf. figure 1). Due to space limitations, we only describe the essential translation process.

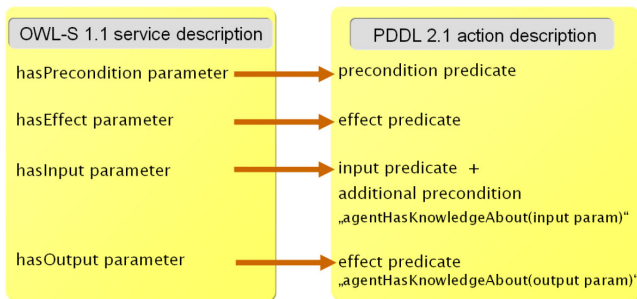


Figure 1: Mapping between OWL-S service and PDDL action description

Any OWL-S service profile input parameter correlates with an equally named one of a PDDL action, and the hasPrecondition service parameter can directly be transformed to the precondition of the action by use of predicates. The same holds for the hasEffect condition parameter. Figure 3 provides an example of such a mapping of an OWL-S 1.1 service that calculates the route from given GPS-position of an accident to the nearest hospital for an emergency physician to the equivalent PDDL 2.1 action description. Either this service already exists, hence its translation is part of the planning domain description, or, as a requested

service (query) becomes part of the planning problem description.

```

Part of OWL-S 1.1 service description
- <service:Service rdf:ID="CalculateRoute">
- <service:presents>
- <profile:Profile rdf:ID="CalculateRoute_Profile">
  <service:presentedBy rdf:resource="#CalculateRoute" />
  <profile:serviceName>
    rdf:datatype="http://www.w3.org/2001/XMLSchema#string">CalculateRoute</profile:serviceName>
  <profile:textDescription rdf:datatype="http://www.w3.org/2001/XMLSchema#string">calculates the
    routes from the emergency station (where the ambulance is at) to the accident position and
    back to the chosen hospital.</profile:textDescription>
  </profile:Profile>
</service:presents>
- <service:describedBy rdf:resource="#CalculateRoute_Model" />
- <service:supports>
- <grounding:WsdGrounding rdf:ID="CalculateRoute_Grounding">
  <service:supportedBy rdf:resource="#CalculateRoute" />
  </grounding:WsdGrounding>
</service:supports>
</service:Service>
Service I/O
- <process:hasInput>
- <process:Input rdf:ID="CalculateRoute_AccidentGpsPosition">
  <process:parameterType
    rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">http://www.dfki.de/scallops/health
    scallops/EMAOntology.owl#GpsPosition</process:parameterType>
</process:Input>
- <process:hasOutput>
- <process:Output rdf:ID="CalculateRoute_RouteToAccident">
  <process:parameterType
    rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">http://www.dfki.de/scallops/health
    scallops/EMAOntology.owl#Route</process:parameterType>
</process:Output>
- <process:name rdf:datatype="http://www.w3.org/2001/XMLSchema#string">CalculateRoute</process:name>
- <process:hasInput>

```

Figure 2: Part of OWL-S 1.1 service description

```

PDDXML 2.1 action description
serviceName → <action name="CalculateRoute">
hasInput → <param type="GpsPosition">?CalculateRoute_AccidentGpsPosition</param>
- <parameters>
  <param type="GpsPosition">?CalculateRoute_EmergencyPhysicianGpsPosition</param>
  <param type="ListOffHospitals">?CalculateRoute_ListOffHospitals</param>
  <param type="Route">?CalculateRoute_RouteToNearestHospital</param>
  <param type="GpsPosition">?CalculateRoute_AccidentGpsPosition</param>
</parameters>
- <precondition>
- <and>
  <pred name="agentHasKnowledgeAbout">
    <param>?CalculateRoute_EmergencyPhysicianGpsPosition</param>
  </pred>
  <pred name="agentHasKnowledgeAbout">
    <param>?CalculateRoute_ListOffHospitals</param>
  </pred>
  <pred name="agentHasKnowledgeAbout">
    <param>?CalculateRoute_AccidentGpsPosition</param>
  </pred>
  <pred name="agentHasKnowledgeAbout">
    <param>?CalculateRoute_EmergencyPhysician</param>
  </pred>
  <pred name="Person_IsAT">
    <param>?CalculateRoute_EmergencyPhysician</param>
    <param>?CalculateRoute_EmergencyPhysicianGpsPosition</param>
  </pred>
  <not>
    <pred name="Person_IsAT">
      <param>?CalculateRoute_EmergencyPhysician</param>
      <param>?CalculateRoute_AccidentGpsPosition</param>
    </pred>
  </not>
</and>
</precondition>
- <effect>
- <and>
  <pred name="agentHasKnowledgeAbout">

```

Figure 3: Part of action description in PDDXML converted by OWLS2PDDL

However, the conversion of the output of an individual OWL-S service, that is the information the service offers to the world, to PDDL turns out to be more problematic. The problem is that the service hasEffect condition explicitly describes how the world state will change while this is not necessarily the case for an hasOutput parameter value, though it indeed could implicitly influence the composition planning process. However, PDDL does not allow to describe non-physical knowledge such as train connections produced as an output of a service.

This problem can be solved by mapping the service output parameter X to a special type of the service hasEffect pa-

parameter. In particular, every output variable  $X$  is described in, and added to the current (physical) planning world state by means of a newly created add-effect predicate in PDDL uniquely named "agentHasKnowledgeAbout( $X$ )". Similarly, each input variable  $Y$  is mapped to an input parameter  $Y$  of an PDDL action complemented by precondition predicate "agentHasKnowledgeAbout( $Y$ )". OWLS-Xplan would only use a service description during its planning process, if the additional precondition predicate "agentHasKnowledgeAbout( $Y$ )" on available knowledge about service input data is satisfied such that  $X = Y$  holds. Otherwise the service execution could fail since checking the service preconditions may reveal that they are not satisfied in the actual world state.

```

<Route rdf:ID="Route_Transport2"/>
<DateTime rdf:ID="DateTime_ArrivalFlight2"/>
<PersonName rdf:ID="PersonName_EmergencyPhysician"/>
<Route rdf:ID="Route_Transport3"/>
<GpsPosition rdf:ID="Position_TransportCompany2"/>
<EmergencyPhysician rdf:ID="Physician_EmergencyPhysician">
  <Person_hasName rdf:resource="#PersonName_EmergencyPhysician"/>
  <Person_hasCreditCard>
    <CreditCard rdf:ID="CreditCard_EmergencyPhysician">
      <CreditCard_hasCreditCardNumber>
        <CreditCardNumber rdf:ID="CreditCardNumber_EmergencyPhysician"/>
      </CreditCard_hasCreditCardNumber>
    </CreditCard>
  </Person_hasCreditCard>
  <Person_isAt>
    <GpsPosition rdf:ID="Position_EmergencyPhysician"/>
  </Person_isAt>
</EmergencyPhysician>

```

„Emergency physician is at some (GPS) position“

Figure 4: Part of initial world state semi-automatically built by OWLS-Xplan editor



Figure 5: Part of problem description in PDDLXML converted by OWLS2PDDL

Figure 4 shows an example of an initial world state that has been semi-automatically built by the OWLS-Xplan editor. In particular, it currently provides application-oriented templates to the user that allow her to quickly enter, modify, and validate the initial world state and the query, i.e., the goal state, depending on the specific situation and problem at hand. If the user wants to query the agent for a

medical transportation service, she only has to fill in a few pre-given templates, thereby setting the values of default parameters of world state and one requested service with related OWL ontologies attached to the template. This initial state and request description is then automatically converted to the corresponding PDDLXML problem description by OWLS2PDDL (cf. figure 5). This, in turn, is fed into the planner Xplan to find a solution, i.e. a plan sequence of services or actions on the initial world state that satisfy the given goal.

## The AI planner Xplan

Xplan is a heuristic hybrid search planner based on the FF-planner developed by Hoffmann and Nebel (Hoffmann & Nebel 2001). It combines guided local search with graph planning, and a simple form of hierarchical task networks to produce a plan sequence of actions that solves a given problem. This yields a higher degree of flexibility compared to pure HTN planners like SHOP2 (Sirin *et al.* 2004) whereas the use of predefined workflows or methods improves the efficiency of the FF-planner. In contrast to the general HTN planning approach, a graph-plan based planner is guaranteed to always find a solution independent from whether the given set of decomposition rules for HTN planning would allow to build a plan that contains only atomic actions. In fact, any graph-plan based planner would test every combination of actions in the search space to satisfy the goal which, of course, can quickly become prohibitively expensive.

Xplan combines the strengths of both approaches. It is a graph-plan based planner with additional functionality to perform decomposition like a HTN planner. Figure 6 shows an example of how Xplan of OWLS-Xplan uses only those parts of a given method for decomposition that are required to reach the goal state with a sequence of composed services  $WS_1$  and  $WS_3$ . In contrast, HTN planning would completely decompose  $M$  into  $WS_1$  followed by  $WS_2$ , hence output also  $WS_2$  which is of no use for reaching the goal.

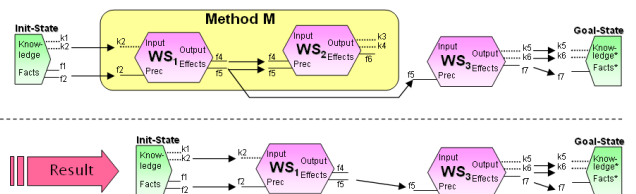


Figure 6: Using parts of methods to reach a goal state in OWLS-Xplan

## Overview

The Xplan system consists of one XML parsing module, and following preprocessing modules. First, required data structures for planning are created and filled, followed by the generation of the initial connectivity graph and goal agenda. The latter two actions are interleaved with replanning. The core (re-)planning modules concern the heuristically relaxed



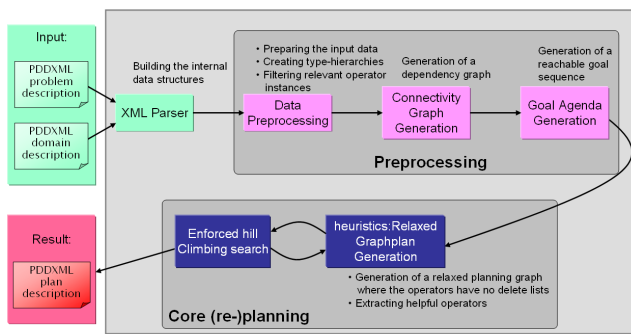


Figure 7: Architecture of Xplan

graph-plan generation and enforced hill-climbing search (cf. figure 7).

After the domain and problem definitions have been parsed, Xplan compiles the information into memory efficient data structures. A connectivity graph is then generated and efficiently realized by means of a look up table, which contains information about connections between facts and instantiated operators, as well as information about numerical expressions which can be connected to facts. This connectivity graph is maintained during the whole planning process and used for the actual search. In case of a replanning situation, the connectivity graph is adjusted according to the changed new world state.

Xplan uses an enforced hill-climbing search method to prune the search space during planning, and a modified version of relaxed graph-planning that allows to use (decomposition) information from hierarchical task networks during the efficient creation of the relaxed planning graph, if required, such as in partially hierarchical domains. Information on the quality of an action (service) are utilized by the local search to decide upon two or more steps that are equally weighted by the used heuristic.

In addition, Xplan includes a replanning component which is able to re-adjust outdated plans during execution time. Therefore, the execution engine informs the planning module about changed world states, and the Xplan replanning component decides whether the remaining plan fragment to execute is still valid or whether a re-planning has to be initiated. Figure 9 shows a fragment of the plan description produced by Xplan, i.e., a sequence of actions, that is the composed sequence of corresponding OWL-S services, that can be executed by the agent.

We implemented Xplan modularly in C++, using the Microsoft MSXML Parser for reading PDDLXML definitions and generating plans in XML format. Alternatively, Xplan also provides an interface for direct interchange of planning data without having to use PDDLXML as interchange format. Each component of Xplan will be described in more detail in subsequent sections.

### Data preprocessing component

Solving a planning process can be viewed as a search problem in the space of all possible combinations of action se-

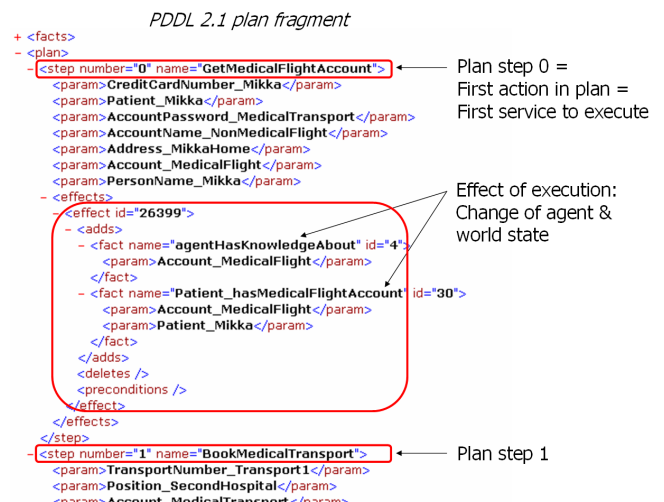


Figure 8: Part of plan description in PDDLXML.

quences. Xplan starts off with preprocessing the input data assigning initial values to each predicate of the given (problem and domain) state description in PDDL.

**Type relation, conversion and simplification of formulas.** In the second step, Xplan creates a matrix, that describes all type relations and type inclusions. Predicates which are neither negative nor positive in the effect list of an operator are considered static for the complete planning process, hence are removed from all preconditions and effect lists. Then, the preconditions and effects are converted to disjunctive normal form.

**Operator-templates, instantiation and reduction of search space.** Xplan creates templates from these simplified operators which are instantiated by all possible combinations of input data based on object instances as described in the PDDL problem description. The set of instantiated operators is then reduced by means of fixed point computation leading to useable and relevant operators. This is achieved by iteratively starting with applying all operators to the initial state. Facts that are added to the state by operators will be stored in a *potential positive facts* list. The respective operators are marked as relevant. This process is repeated until either no new facts nor operators are added to the lists. Operators and facts that are neither reachable nor able to be fulfilled, are removed from the basis set of instantiated operators. Relations between instantiated methods, complex actions and atomic actions are built, to speed up the search and decomposition later on. Furthermore, to guarantee completeness while searching, all negative facts that have a corresponding fact in the potential positive facts list are also stored in the list of relevant facts. Both relevant facts and operators are used to build the connectivity graph.

### Generation of the connectivity graph and goal agenda

The connectivity graph is built upon the list of relevant facts, and relevant operators in an iterative process that detects the

dependencies between the precondition, add- and delete lists of operators and facts. Once created, the connectivity graph remains static during the search and planning process. In contrast to traditional plan graph algorithms, Xplan does not consider the complete set of goals as a whole but computes an ordered list of goals, the so called goal agenda. The corresponding goal graph is generated based upon this agenda and the FALSE-sets of each goal. Finally, the transitive hull over the goal graph is being computed which is then used to classify goals into goal sets.

Let  $(\mathcal{O}, \mathcal{I}, \mathcal{G})$  be a planning problem for which a goal agenda with  $n$  goal-sets  $G_{s_0}, \dots, G_{s_n}$  exists. The search algorithm starts with the initial state  $I_0 = \mathcal{I}$  and the first goal-set  $G_{s_0}$  as the planning goal  $G$ . If a solution  $P_1$  is found which leads from  $I_0$  to  $G_{s_0}$ , then the plan is used on  $P_1$  and  $I_0$ . The resulting state  $I_1 = Result(I_0, P_0)$  is then used as the starting point for the search using  $I_1$  as initial state and planning goal  $G = G_{s_0} \cup G_{s_1}$ . Thus all reached goals  $G_{s_0}$  to  $G_{s_{k-1}}$  remain valid while searching for a solution for  $G_{s_k}$ . For the current planning goal  $G_k$  in iteration  $k$  it holds that

$$G_k = \bigcup_{i=0}^k G_{s_i}$$

The Xplan search algorithm uses a *no-ops first*-strategy, i.e., goals achieved in previous iterations are marked and only temporally deleted if they will be generated again later on. This guarantees that the planner generates no sub-optimal plans with loops.

### The Relaxed Graphplan heuristic

After the goal-agenda has been generated, the search process starts. The search consists of two interleaved processes. The Relaxed Graphplan heuristic (Hoffmann 2000) approximates the distance between the initial state  $\mathcal{I}$  to all reachable states  $S$ . These distance values are then used to guide the forward directed search. After each successful step the distance values are updated again using the heuristic.

**Definition 0.1** A state  $S = (F_S, h_S, N_S)$  is defined as

- $F_S$  is a set of all facts which are true in state  $S$ .
- $h_S$  is distance to the goal given by a heuristic.
- $N_S$  is a set of helpful action which can be used in state  $S$ .

**Complex actions and hierarchical task networks within relaxed Graphplan** We have expanded the Relaxed Graphplan heuristic based algorithm by adding an HTN planner component, and utilization of numerical and boolean facts that can be updated online during the planning phase by external function calls. As a consequence, not only atomic operators but also complex actions and methods are allowed during planning. If a complex action is used while generating a plan graph of which preconditions on some graph layer  $E_i$  are satisfied, the HTN component then tries to decompose the complex action using a method-structure element or complex task. A relevant method is searched for by looking up the connectivity graph. Since more than one method could be relevant for decomposition, a heuristic  $h_{htn}^d$  is used to determine the most useful one. The selected

partial task network itself may contain complex actions that have to be recursively decomposed. Through selection of useable operators  $O_i$  of plan graph layer  $E_i$  the algorithm first tries to select complex actions. If a solution cannot be found by decomposition, Xplan tries to find a solution without using the HTN component.

**External procedure calls** Many planners offer the possibility to use numerical values with the standard operators  $+$ ,  $-$ ,  $*$ ,  $\div$ ,  $\dots$  during the planning process. In most cases these functions are not only bounded in their number but rather hard-coded in the planner such as in the Metric-FF planner (Hoffmann 2003). This is a drawback because the system cannot be expanded without having to change the code. In contrast, Xplan offers the use of so-called external call-back functions. A call-back function is linked to a predicate by means of a fluent variable which contains its return value. This way, Xplan is able to obtain actual information on the value of predicates during the planning process by calling the linked call-back function. The function returns a boolean value which indicates whether the linked predicate is set or removed from the world state in the next layer of the plan graph. New call-back functions can be added without changing the code of the planner itself.

The fluents are utilized by the planner on both the global fluent-layer of the plan graph, that represents the current state in the computed plan, and the local fluent-layer storing the changed new states of the fluents for their use in the next planning steps. An update of the global fluent-layer is performed each time the fast-forward search finds a better state with respect to a given utility function. The values of the local fluent-layer are used for calculating those facts that are satisfied by executable actions of some layer of the plan graph. The pseudo-code of the algorithm 1 for generating the relaxed plan graph is provided in the annex of this paper.

### Extraction of a relaxed plan from the planning graph

Let  $(\mathcal{O}', \mathcal{I}_x, \mathcal{G})$  the planning problem to solve and  $PG$  the relaxed plan graph with  $k$  layers. Figure 9 shows a simple example of problem and domain description together with initial part of a corresponding plan graph.

The search for a relaxed plan starts at the top most layer  $E_{k-1}$ . For every goal of the current layer  $E_i$ ,  $i < k$ , an action of  $E_{i-1}$  is selected that satisfies one or multiple goals. Let  $F_i$  be the set of facts of layer  $E_i$ , then the selection of goals is done by use of a heuristic  $h_d$  that measures the barrier for executing an action.

$$h_d(o) := \sum_{p \in pre(o)} \min\{i \mid p \in F_i\} + (1 - QoS(o))$$

with  $QoS(o) \in ]0, 1]$  quality of service of action  $o$ . The intersection of the set of selected actions' preconditions of  $E_{i-1}$  and the facts of  $E_{i-1}$  makes up the goal set  $G_{i-1}$  of layer  $E_{i-1}$ . This goal set then has to be fulfilled by use of actions of the subsequent layer  $E_{i-2}$ . The recursion is going on until the lowest layer of the planning graph with initial facts is reached. This is then the relaxed plan, that consists of an action sequence  $A_0 \dots A_{k-1}$ .  $k-1$  is the index of the first layer which contains all goals of the original problem.



#### PDDL problem and domain description

```

Init (Have(X))
Goal(Have(X) ∧ Used(X))
Action(Use1(X)
  Precondition: Have(X)
  Effect: ¬ Have(X) ∧ Used(X))
Action(Use2(X)
  Precondition: ¬ Have(X)
  Effect: Have(X))

```

#### Plan graph PG

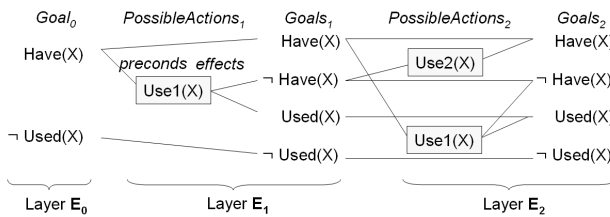


Figure 9: Part of a plan graph with  $k = 3$  layers

To get an approximation how far away it is from state  $\mathcal{I}_x$  to goal  $\mathcal{G}$ , a heuristic  $h(\mathcal{I}_x)$  (described in (Hoffmann 2000)):

$$h(\mathcal{I}_x) := \sum_{i=0}^{k-1} |A_i|$$

The value  $h(\mathcal{I}_x)$  indicates the length of the relaxed, sequential solution of  $\mathcal{O}$  starting with  $\mathcal{I}_x$ . This value in combination with the state information  $\mathcal{I}_x$  constitutes the new search state  $S$  which is included into the search space of the fast forward search.

### Detecting helpful actions

In addition to the heuristic, Xplan computes the set  $H_S$  of helpful executable actions for every search state  $S$  such that the goal  $\mathcal{G}$  eventually can be reached.

**Definition 0.2** A helpful action of a search state  $S$  is an action, that satisfies at least one proposition of the goal set  $G_1$  of the first layer in the plan graph. The set of helpful actions is described as follows:

$$H_S(S) := \{o \mid (pre(o) \subseteq S) \cap (add(o) \cap G_1(S) \neq \emptyset)\}$$

If there are many helpful actions, then actions of an HTN decomposition are preferred. The reason is, that such actions are more highly probable to be succeeded by an useful action in the task network as part of the relaxed plan.

### Local search with enforced hill-climbing

Xplan uses an enforced hill-climbing search algorithm to search for best reachable states during generation of the global plan to satisfy a given goal. It combines the standard search strategy with a breadth search for a better state than the given one not only in its direct neighborhood but within the set of successor states of  $S$  that are reachable by applying a helpful action of  $N_S$ . This search strategy performs as follows.

- Compute the distance between the starting state  $\mathcal{I}$  and the goal state  $\mathcal{G}$  by use of Relaxed Graphplan, and the set of helpful actions  $I$ .
- Initialize the enforced hill-climbing with  $\mathcal{I} = (F_{\mathcal{I}}, h_{\mathcal{I}}, N_{\mathcal{I}})$  as input.
- Enforced Hill-Climbing analyzes all reachable states that have been computed. It assigns each state with its approximative distance to the goal by use of Relaxed Graphplan.
- If a better state is found, then include this state into the current plan, and use it as a basis for further search. Update all fluents on the current layer by invoking the respective call-back functions.
- Terminate if a state  $S' = \mathcal{G}$  is reached in which all given goals are satisfied. Otherwise, if not at least one goal has been achieved, the search failed. In this case, a new complete breadth search is instantiated on  $\mathcal{I}$  to find a solution, if it exists.

The pseudo-code of the local search by enforced hill-climbing algorithm is shown in algorithm 2.

### Re-planning component

During plan execution, the agent has to check for each action of the plan whether its preconditions hold, or not. If at least one precondition is not satisfied, Xplan gets informed about which facts are invalid, at which position in the plan this problem occurs, and then checks whether the original plan still can be executed. Otherwise, it tries to fix the problem by searching for an alternative path in the connectivity graph from the actual position in the plan to the goal state. In addition, it may temporally block unnecessary actions to reduce the search space, thereby avoiding a complete preprocessing phase.

### Plan patterns

Xplan builds ordered service composition plan sequences only whereas OWL-S allows for more complex plan structures such as unordered, choice, if-then-else, iterations, repeat-until loops, repeat-while loops, split, and split+join. However, complex plan structures can be formed out of the produced plan sequence based on its appropriate analysis and interpretation in posterior. For example, figure 10 shows how a plan sequence looks like, that can be transformed into a split+join structure. In this case, the input of  $WS_2$  does not depend on the output of  $WS_1$ , hence both services may be executed concurrently. Since both are required to achieve the given goal, their results have to be joined after execution.

Figure 11 shows an example of how to realize a split structure. Like in previous example, both services neither influence each other, nor share a common goal to reach, thus can be executed in parallel.

However, the plan structures "choice" and "unordered sequence" are not realizable by proper interpretation of plan sequences created by Xplan. Though, the latter problem is a hard problem for any AI planner in general, including, for example, Shop2 (Sirin *et al.* 2004).

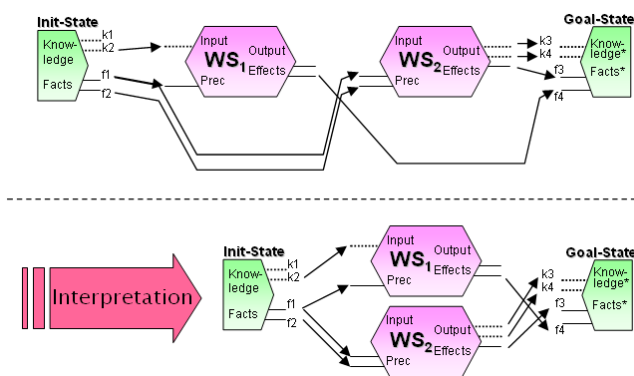


Figure 10: Split + Join interpretation

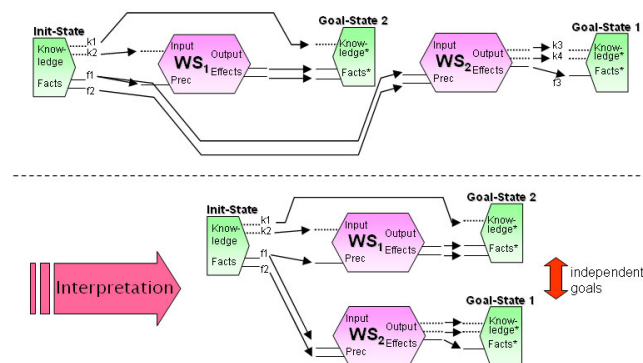


Figure 11: Split structure of an OWLS-Xplan plan

### Related work

A logic-based DAML-S composition planner has been developed at the UMBC, USA (Sheshagiri, desJardins, & Finin 2003). This planner uses STRIPS-style services to compose a plan, given the goal and set of basic services. It is implemented with JESS (Java Expert System Shell), and uses a set of JESS rules to translate DAML-S descriptions of atomic services into planning operations.

One of the currently most prominent service composition planners is Shop2 (Simple Hierarchical Ordered Planner 2) developed at the University of Maryland, USA (Wu *et al.* 2003). It is a hierarchical task network (HTN) planner well-suited for working with the hierarchically structured OWL-S process model. The authors proved the correspondence between the semantics of Shop2 and situation calculus semantics of the OWL-S process model. The implemented Shop2 soundly and completely plans over sets of OWL-S descriptions, and treats the output of a web service as effects that either change the planning agent's knowledge, or the world state. Shop2, like HTN planner in general, replaces those elements of the provided methods (workflows) by special methods or atomic actions until the composition plan contains only atomic actions that correspond to available web services. During planning, web services are not executed,

hence do not affect the world state.

Both Xplan and Shop2 base on the closed world assumption, use PDDL for problem description, allow external (call-back) functions to be bounded to variables and executed during planning, and generate total ordered, instantiated plan sequences for a given initial state, goal and planning domain. Among others, the main difference between Shop2 and Xplan is inherent to the individual planning processes. In essence, Shop2 plans are generated by use of given decomposition rules (methods), hence a solution to the planning problem is not always guaranteed to be found (Lotem, Nau, & Hendler 1999). In contrast, hybrid Xplan as part of OWLS-Xplan tries to plan by means of (a) method decomposition using only relevant parts of it, discarding useless actions, thereby reducing the plan size, and (b) if this is not successful, uses its relaxed graph plan algorithm to find a solution, if it exists.

### Conclusion

We presented an OWL-S service composition planner, called OWLS-Xplan, that allows for fast and flexible off-line composition of OWL-S services by use of an OWLS2PDDL converter, and a hybrid AI planner that combines relaxed Graphplan FF-planner with local search and HTN based planning, and a re-planning component. OWLS-Xplan has been implemented in C++ and Java, and is currently in use in a prototyped medical health information service system. It is intended to make the OWLS-Xplan code package available to the community at [www.semwebcentral.org](http://www.semwebcentral.org).

### References

- Hoffmann, J., and Nebel, B. 2001. The FF Planning System: Fast Plan Generation Through Heuristic Search. *Journal of Artificial Intelligence Research (JAIR)* (14):253–302.
- Hoffmann, J. 2000. A heuristic for domain independent planning and its use in an enforced hill-climbing algorithm. *Proceedings of 12th Intl Symposium on Methodologies for Intelligent Systems, Springer Verlag*.
- Hoffmann, J. 2003. The Metric-FF planning system: Translating Ignoring Delete Lists to Numeric State Variables. *Artificial Intelligence Research (JAIR)*, vol 20.
- Lotem, A.; Nau, D.; and Hendler, J. 1999. Using planning graphs for solving HTN problems. *Proceedings of AAAI/IAAI conference, USA*.
- Schmidt, M. 2005. Ein effizientes Planungsmodul fuer die lokale Planungsebene eines InterRaP Agenten. Master's thesis, Universitaet des Saarlandes.
- Sheshagiri, M.; desJardins, M.; and Finin, T. 2003. A planner for composing services described in DAML-S. *Proceedings of AAMAS 2003 Workshop on Web Services and Agent-Based Engineering*.
- Sirin, E.; Parsia, B.; Wu, D.; Hendler, J.; and Nau, D. 2004. HTN planning for web service composition using SHOP2. *Journal of Web Semantics*, 1(4) 377–396.
- Wu, D.; Parsia, B.; Sirin, E.; Hendler, J.; and Nau, D. 2003. Automating DAML-S web services composition using SHOP2. *Proceedings of the 2nd International Semantic Web Conference (ISWC2003)*, pages 20-23, Sanibel Island, Florida, USA.

```

function BuildRelaxedPlanningGraph() computes
relaxedPlanningGraph or fails
input: InitialFacts[] : List of Facts
input: GoalFacts[] : List of Facts
local: CurrentLayerFacts[], NextLayerFacts[] : List of
Facts
local: CurrActivActions[] : List of Actions
local: CurrentLayer : int
begin
  CurrentLayerFacts = InitialFacts; CurrentLayer = 0;
  while ! AllGoalsActive(GoalFacts) do
    foreach Fact f in CurrentLayerFacts do
      Increment precondition counter of actions
      which f is a precondition of;
    end
    foreach Fact f in CurrentLayerFacts do
      /* First collect all action,
      that are a result of a
      method decomposition and
      compute the layer, when it
      is earliest executed. */
      CurrActivHTNActions +=
      GetActiveHTNActions(CurrentLayer, f);
      /* Select all remaining
      executable actions, that
      are part of the
      current-layer */
      CurrActivActions +=
      GetActivePrimitiveActions(CurrentLayer, f);
    end
    foreach Action a in CurrActivHTNActions do
      if all preconditions of a are satisfied AND
      Layer of a == CurrentLayer then
        /* a is executable, all
        preconditions are
        fulfilled and is
        executable in the layer
        */
        NextLayerFacts +=
        GetAddedFactsFromAction(a);
        RemoveFromList(a, CurrActivHTNActions);
      end
    end
    foreach Action a in CurrActivActions do
      if all preconditions of a are satisfied AND
      Layer of a == CurrentLayer then
        NextLayerFacts +=
        GetAddedFactsFromAction(a);
        RemoveFromList(a, CurrentActivateActions);
      end
    end
    CurrentLayerFacts = NextLayerFacts;
    NextLayerFacts = <>;
    /* Increasing layer counter and
    continue with next layer. */
    CurrentLayer = CurrentLayer + 1;
    if CurrentLayerFacts == <> then
      /* If a fix point is reached
      regarding facts and actions
      and the goal isn't fulfilled,
      the problem isn't solvable.
      */
      if ! AllGoalsActive(GoalFacts) then
        return FAILURE;
      end
    end
    return CurrentLayer;
  end

```

```

function DoEnforcedHillClimbing() computes
validPlan: Plan or fails
input: InitialState : State input: GoalState : State
  local: S : State /* the current computed
  state */
local: S' : State /* possible successor of S
  */
local: currPlan : Plan /* current plan */
local: hS : int /* the distance of S to a
  goal computed by use of Relaxed
  Graphplan */
local: hS' : int local: NS[] : List of Actions /* List
  of helpful action based on state S
  */
local: NS'[] : List of Actions
begin
  /* The initial plan is empty */
  currPlan = <>;
  S = InitialState; /* Compute the distance
  from starting state to goal */
  hS = BuildRelaxedPlangraph(S, GoalState);
  /* Compute helpful actions for S
  */
  NS = GetHelpfulActions(S);
  while hS ≠ 0 do
    /* Searching with breadth search
    for a state S' with HS' < HS
    within NS and their
    successors. BFS_Expand
    computes for every relevant
    state the distance between
    goal and helpful actions.
    This is done by
    BuildRelaxedPlangraph and
    GetHelpfulActions */
    S' = BFS_Expand(S, NS);
    if S' == NULL then
      return FAILURE;
    else
      /* If a state S' is found,
      the action sequence is
      attached to the end of the
      current plan, that enables
      to get from S to S'. */
      currPlan =
      currPlan + ActionsPath(S, S');
      /* Update fuent of the
      global fluent-layer */
      UpdateGlobalFluents(S, S');
      /* The search goes on
      beginning with S'. NS' is
      computed before by
      BFS_Expand and can still be
      use. */
      S = S';
      NS = NS';
    end
  end
  return currPlan;
end

```

Algorithm 2: Local Search by enforced hill-climbing

# Fast Composition Planning of OWL-S Services and Application<sup>1</sup>

Matthias Klusch and Andreas Gerber  
German Research Center for Artificial Intelligence  
Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany  
{klusch@dfki.de, andreas.gerber@x-aitment.net}

## Abstract

*In this paper, we present the implementation, evaluation, and application of our OWL-S service composition planner OWLS-XPlan. Medical services described in OWL-S 1.1 and ontologies are converted to initial state and goal descriptions in PDDL 2.1, which are then used by the fast heuristic FF planner XPlan for generating an execution complete composition plan. Results of experimental evaluation of XPlan shows its top performance compared with other selected AI planners. OWLS-XPlan is used in an agent based mobile e-Health system for emergency medical assistance planning tasks.*

## 1. Introduction

Though the composition of complex Web services attracted much interest in different fields related to service oriented computing, there are only a few implemented composition planning tools publicly available for the semantic Web such as the HTN composition planner SHOP2 [11, 12] for OWL-S services. One problem with pure HTN planners is that they require task specific decomposition rules and methods developed at design time, hence are not guaranteed to solve arbitrary planning problems. That, in particular, motivated the development of our hybrid off-line composition planner for OWL-S 1.1 services, called OWLS-XPlan [13], which is guaranteed to find a solution if it exists, though the corresponding planning problem remains to be NP-complete.

OWLS-XPlan is integral part of the prototypically implemented agent based mobile eHealth system, called Health-SCALLOPS, for secure emergency medical assistance planning tasks, such as patient repatriation and relocation to selected hospital. An extended version of OWLS-XPlan, called OWLS-XPlan+, that allows for heuristic quasi-online re-planning of composite OWL-S services has also been implemented and is currently in use in a different e-health application scenario within the European project CASCOM.

The remainder of this paper is structured as follows. Section 2 briefly introduces OWLS-XPlan, followed by the results of the performance evaluation of its core planning module XPlan, and its implementation in sections 3 and 4, respectively. The use of OWLS-XPlan in the Health-SCALLOPS application is described in section 5. We briefly refer to related work and conclude in sections 6 and 7, respectively.

## 2. OWLS-XPlan Overview

The semantic web service composition planner OWLS-XPlan consists of several modules for pre-processing and planning of composite OWL-S services (cf. figure 1). It takes a set of available OWL-S 1.1 services, related OWL ontologies, and a planning request (goal) as input, and returns a planning sequence of relevant OWL-S services that satisfies the goal.

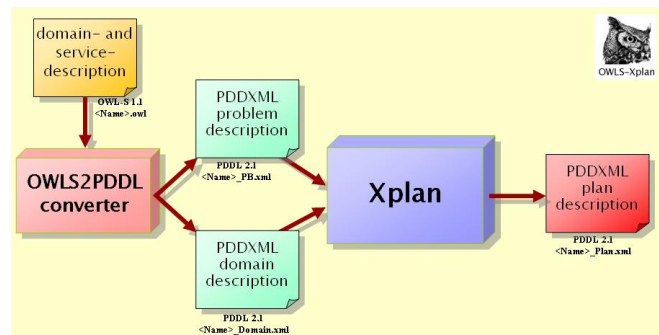


Figure 1. OWLS-XPlan overview

For this purpose, it first converts a given domain ontology and service descriptions in OWL and OWL-S 1.1, respectively, to equivalent PDDL 2.1 problem and domain descriptions using an integrated OWLS2PDDL converter. The domain description contains the definition of all types, predicates and actions, whereas the problem description includes all objects, the initial state, and the goal state. Both descriptions are then used by the AI

<sup>1</sup> This work has been supported by the German Ministry of Education and Research (BMBF 01-IW-D02-SCALLOPS), and the European Commission (FP6 IST-511632-CASCOM).



planner XPlan to create a plan in PDDL that solves the given problem in the actual domain. An operator of the planning domain corresponds to a service profile in OWL-S, while a method is a special type of operator for fixed complex services that OWLS-XPlan may use during its planning process. For reasons of convenience, we developed a XML dialect of PDDL, called PDDXML, that simplifies parsing, reading, and communicating PDDL descriptions using SOAP.

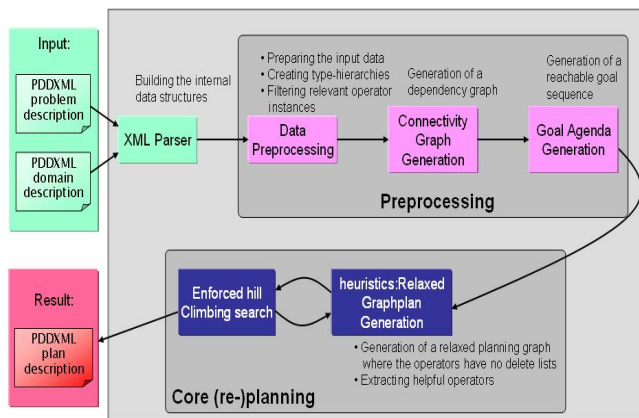
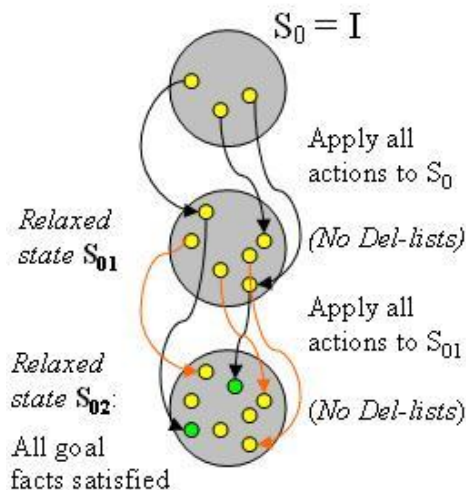


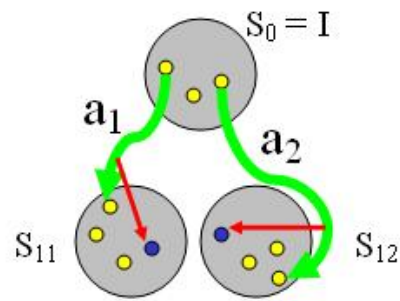
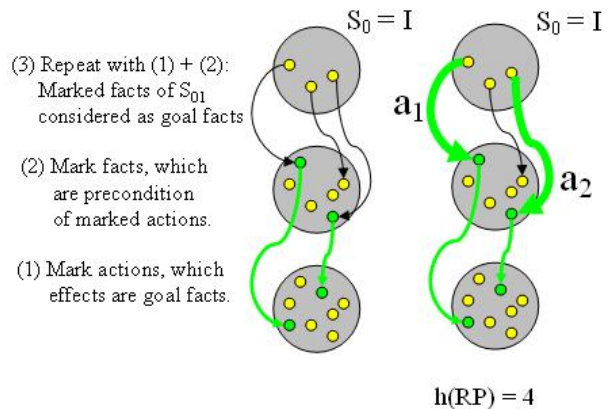
Figure 2. XPlan planning module

The planning module XPlan (cf. figure 2) is a heuristic hybrid FF planner based on the FF planner developed by Hoffmann and Nebel [4, 5, 6]. It combines guided local search with relaxed graph planning, and a simple form of hierarchical task networks (HTN) to produce a plan sequence of actions that solves a given problem. If equipped with HTN methods (composed services), XPlan uses only those parts of decomposed methods that are required to reach the goal state

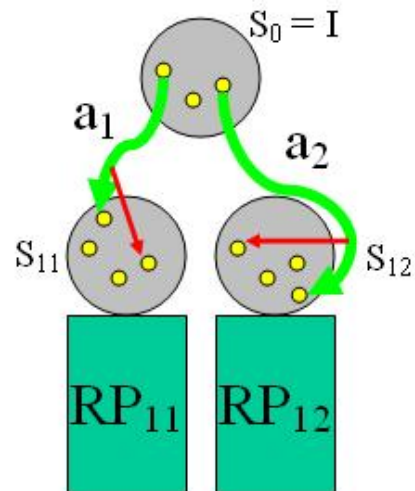
(a) Create RPG



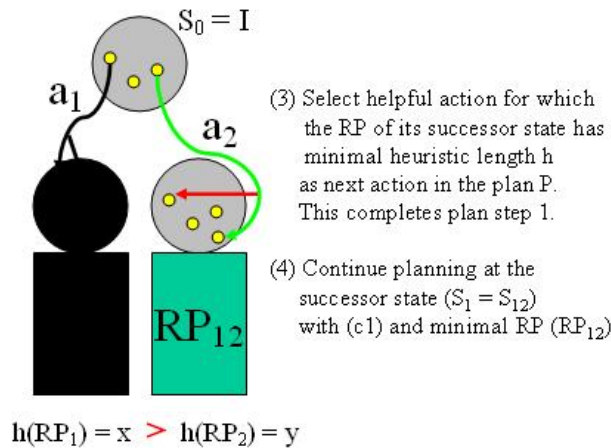
(b) Extract RP From RPG (Backward)



(1) Apply all helpful actions to  $S_0$  including previously ignored *Del-lists* (negative effect factors ●) which yields complete states.



(2) Create RPG for each successor state of helpful action (s. (a))



**Figure 3. XPlan planning step example: (a) Create relaxed planning graph RPG, (b) extract relaxed plan RP from RPG, (c) Select heuristically optimal helpful action (bold green) as action in the plan sequence;  $h(RP)$  = Number of actions in the relaxed plan heuristically equals the length of RP and P.**

For each sub-goal  $g$  of the determined goal agenda, at each planning step  $i$ , XPlan quickly builds a relaxed planning graph  $RPG(i)$  in a fast goal reachability test heuristically ignoring negative effects of actions, and the corresponding relaxed plan  $RP(i)$  in a backward pass from  $g$  to  $S_i$ .

The relaxed plan contains all paths of applicable actions that lead from  $g$  to  $S_i$ , of which only those in its first action-layer 0 are called helpful. In the following, XPlan focuses on the helpful actions of  $RP(i)$  only, hence reduces the search space. Please note that the relaxed plan is not necessarily correct.

In order to decide which helpful action to select as the next action in a valid plan sequence, it applies each of them to  $S_i$  and adds the previously ignored Del-list facts yielding the complete state  $S_{ij}$ , where  $j$  in  $\{1, \dots, l\}$ , denotes the  $j$ -th helpful action applied to state  $S_i$ .

For each of these states the relaxed plan  $RPG(i,j)$  is built to heuristically search for the relaxed plan  $RP(i,j)$  with heuristically minimal length  $h(RP(i,j))$ . In this context, the "plan length"  $h(RP(i,j))$  just denotes the sum of all actions in all action-layers of the RP. Finally, XPlan retains the action  $A_{ij}$  with heuristically minimal goal-distance and starts the next planning step  $i+1$  with  $S_{ij}$ . If there are multiple RPs of equal length, it repeats the same decision process starting at state  $S_{i1}$  (like a breadth first search restricted on helpful actions), and then  $S_{i2}, \dots, S_{il}$  until a minimum is found.

Eventually, all created plans for sub-goals  $g$  of the goal agenda are respectively concatenated which yields the final plan sequence  $P$ . The plan then gets executed,

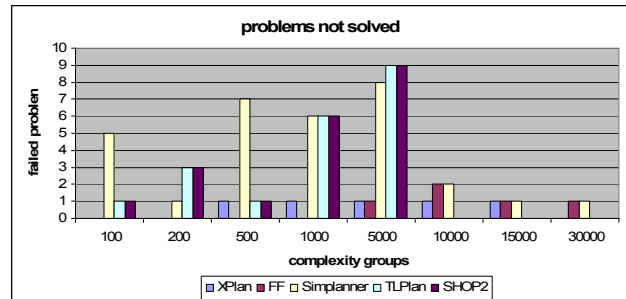
and if it fails, XPlan allows re-planning from the most recent valid state produced by action execution, to avoid a total re-planning, if possible. For more details on OWLS-XPlan in general, and XPlan in particular, together with examples of service translation from OWL-S to PDDLXML we refer the reader to [13]. The software package OWLS-XPlan 1.0 is available at [15].

### 3. Evaluation of XPlan

We evaluated the performance of XPlan, using the publicly available benchmark of the international planning competition IPC3 [2], and compared the results with that of the four top performing IPC3 participants, i.e. FF planner, Sim-Planner, and the HTN planners TLPlan, and Shop2. XPlan was tested without task specific methods. Planning performance was measured in terms of

- the planning completeness, i.e. the total percentage of solved problems (cf. figure 4),
- the average plan length (cf. figure 5), and
- the average plan quality, i.e. the average distance of individual plans from the optimal plan length (cf. figure 6)

in relation to the complexity of the given problems. The complexity of a planning problem is defined as the number of objects of the type definitions specified in the given planning problem domain description. We grouped all test cases of the IPC3 test scenarios leading to 122 problems in total into complexity classes with an increasing number of objects.



**Figure 4. Completeness**

First, we tested the completeness of planning (cf. figure 4). It turned out that XPlan fails to find a solution for problems of mid range complexity only, whereas the FF planner failed to solve the most complex problems. There were no results reported for TLPlan and SHOP2 for the last six test cases, they failed a lot in solving problems of low and mid range complexity, but performed very well in solving more complex problems. Main reason is that the HTN planners turned out to be equipped with methods that better enabled them to solve highly complex problems in most domains.

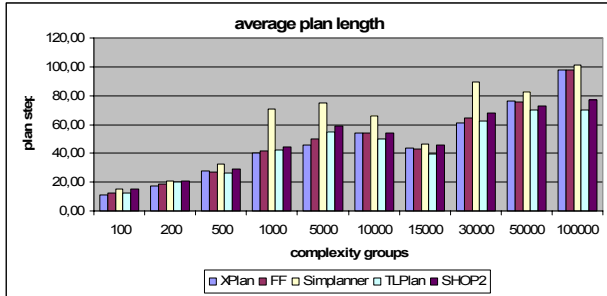


Figure 5. Average plan length

Figure 5 summarizes the results of testing the average plan length in relation to the complexity of the problem definition. The HTN planners produced shorter plans than their competitors with increasing complexity of the problem, whereas XPlan outperformed all other planners for given problems of low and mid range complexity.

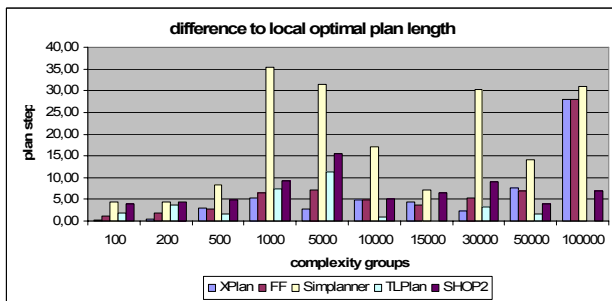


Figure 6. Average plan quality

Finally, we measured the average plan quality in terms of the average distance of individual plans from the optimal solution of a given problem (cf. figure 6). That is, we counted the number of additional plan steps of a solution generated by an individual planner compared to that of the shortest plan created for the given problem, averaged over all test cases per complexity class. In this respect, except for the most complex problems, XPlan outperformed the other planners

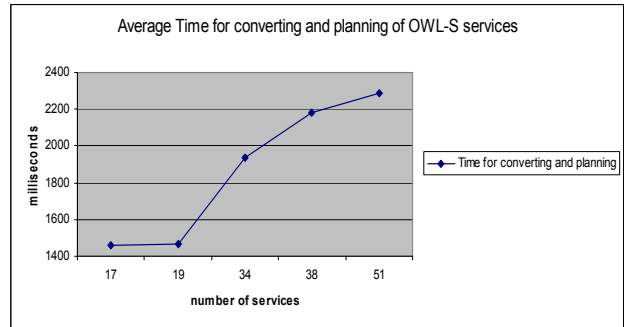
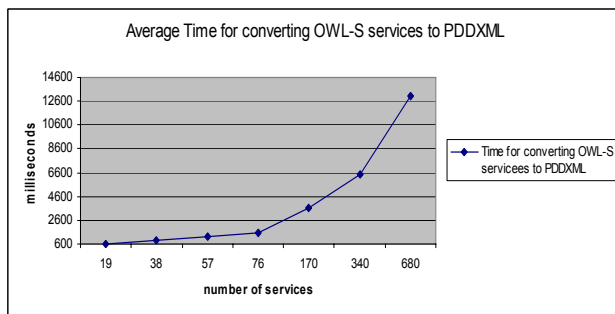


Figure 7. Average run time for conversion and planning by OWLS-XPlan

We did not have specific information about the underlying computing hardware used in the IPC3 competition for run time measurement. Figure 7 shows the reasonably fast run time of converting and planning by OWLS-XPlan on a Siemens-Fujitsu Amelio 1425 notebook with 1.8 Ghz Intel Centrino, and 1 GB RAM.

#### 4. Implementation

OWLS-XPlan has been implemented in Java and C++, and provides an integrated graphical user interface (cf. figure 8 and 9).



Figure 8. OWLS-XPlan graphical user interface (1)

The planning component XPlan is implemented in C++, uses the Microsoft MSXML parser for PDDXML definitions and generating plans in XML format. Besides,

XPlan also provides an interface for direct interchange of planning data without having to use PDDXML as interchange format. In addition, OWLS-XPlan provides an integrated PDDXML editor that allows the experienced user to directly change the planning goal, and edit the initial state ontology for a given planning problem.

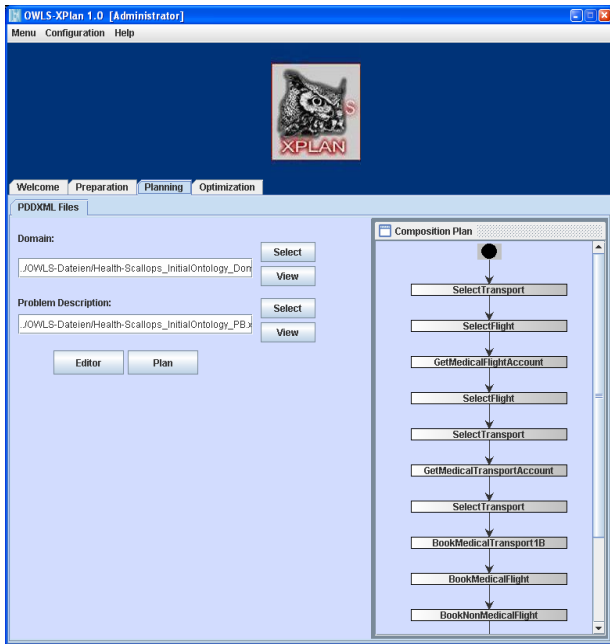


Figure 9. OWLS-XPlan graphical user interface (2)

This initial state ontology is assumed to be provided to the system for planning; we acknowledge that this assumption might be a major hurdle for inexperienced users to actually use the OWLS-XPlan software, so we are currently working on a novice user query interface for OWLS-XPlan version 2. The resulting plan is being displayed (cf. fig. 17) and can be further optimized with respect to given QoS parameters by means of ILP based optimization with newly available equivalent services. OWLS-XPlan v1 has been made publicly available to the semantic web society at the portal semwebcentral.org on December 16, 2005 [15].

## 5. Application Health-SCALLOPS

The service composition planner OWLS-XPlan is used in an agent based mobile eHealth system for emergency medical assistance (EMA) planning tasks, called Health-SCALLOPS. OWLS-XPlan runs on the server of a national EMA centre to support the planning of patient relocation to selected hospitals, or patient repatriation. Medical transport services are offered on line by a variety of medical transport companies in the internet.

**Use case scenario.** For the use case of Health-SCALLOPS as sketched in figure 10, we developed 33 appropriate business application services in OWL-S 1.0 with imported OWL ontologies. We distinguish between the following five roles Health-SCALLOPS users can take

1. Patient, or someone acting on his/her behalf,
2. EMA centre for medical mission planning,
3. Emergency physician with an ambulance car,
4. Hospital physician for local treatment and triggering of relocation to a selected hospital, and
5. Health insurance of the patient.

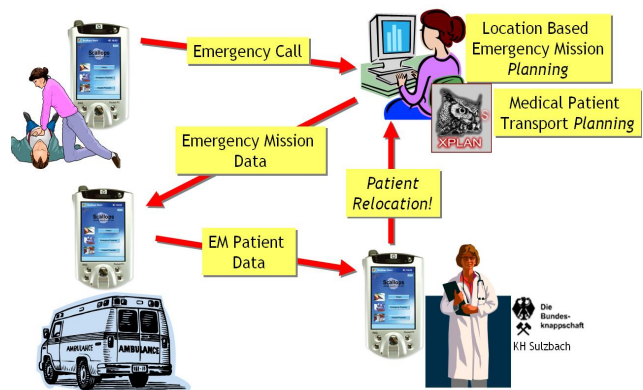
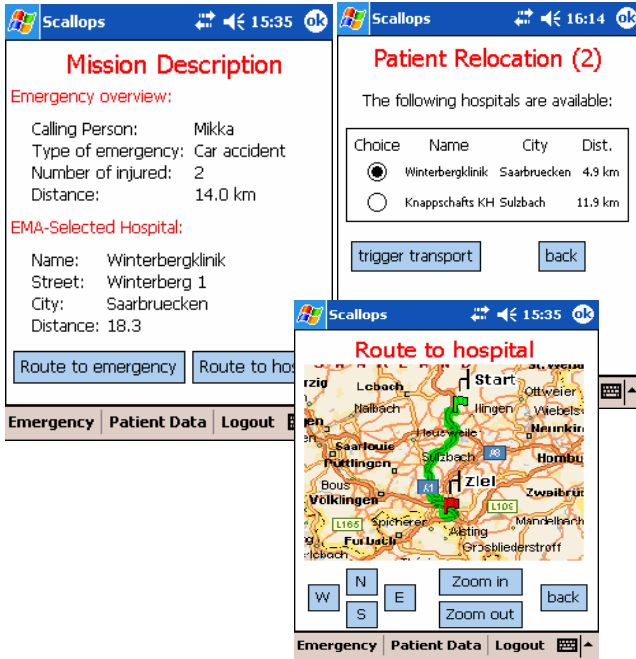


Figure 10. Health-SCALLOPS use case (overview)

Suppose, for example, that Mika meets with an accident. The emergency call to the emergency medical assistance centre (EMA) is given by Mika using his personal Health-SCALLOPS software agents on his PDA. In response to this call, EMA is planning an appropriate medical mission by use of OWLS-XPlan, and alerts the emergency medical doctor in an ambulance car requesting him/her to accomplish this mission. The mission data contain not only the exact GPS position of and summary of the situation given by the patient but the route to the scene and the nearest selected hospital. After providing first aid, the emergency doctor sends the mission data and pre-diagnosis to the hospital doctor to allow for preparation of emergency treatment. Upon arrival at the hospital, the doctor recognizes an additional fissure of Mika's eye that requires further special treatment in an eye-clinic abroad. On behalf of Mika, the doctor asks the EMA centre to plan for this mission of medical patient transport planning. Given that a variety of different medical and non-medical transport companies provide their services on the semantic web, EMA is composing an appropriate transport plan for Mika which a distinguished EMA assistant is then executing.



**Mobile Health-SCALLOPS.** The mobile graphical user interface of Health-SCALLOPS provides role-based functionality to the individual user, and has been implemented in Java for running under WinMobile 2003 on HP's iPAQ PocketPC series 5500 (cf. figure 11).



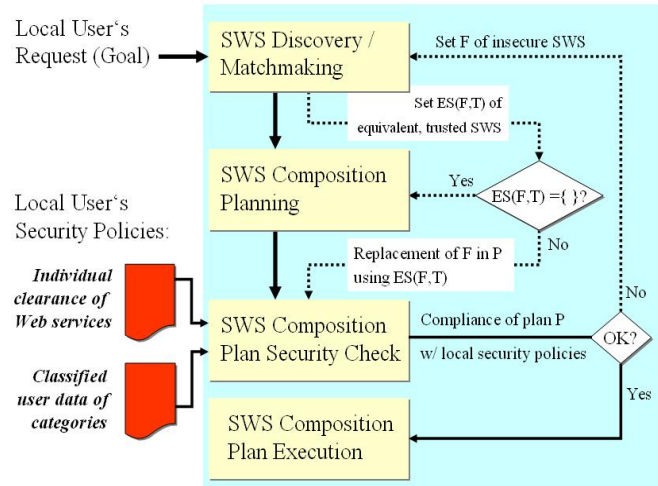
**Figure 11. Mobile Health-SCALLOPS GUI (part of)**

It allows for the initialization (by the patient, the eyewitness of an accident, or hospital doctor), the planning (by EMA centre), and the execution of EMA missions (by mobile emergency physician in ambulance car), and also offers GPS-based route planning.

**Secure and executable composition plans.** In Health-SCALLOPS, we use a variant of OWLS-XPlan that generates privacy preserving and executable medical mission plans. For ensuring the executability of planned sequences of mission related services on the WSDL-grounding level (operation modes, message types) we adopted the approach presented in [16]. In particular, the planning oriented matchmaker module, OWLS-MXP, a variant of OWLS-MX [17], is responsible for continuously checking all pairs of available WSDL service groundings of advertised OWL-S services for their I/O composability at data type level. In particular, it checks whether all pairs of XSLT data types of output-input parameter bindings as stated in the OWL-S process models are coherently matching (either type equivalence or type subsumption). This is to ensure a valid data flow between consecutive services of the semantic service composition plan. OWLS-MXP provides OWLS-XPlan with the set of currently available OWL-S services each

which annotated with a list of grounding composable services. This enables OWLS-XPlan to check at the end of each planning step whether the actual composition plan extended by one helpful action (cf section 2) is also executable.

For verifying whether the generated plan preserves the privacy of patient data before actually executing it, the secure composition planning agent (SCPA) of the EMA centre in the Health-SCALLOPS application scenario performs as follows [18].



**Figure 12: Secure Health-SCALLOPS service composition planning (overview)**

The SCPA gets the request for some desired service in OWL-S from the user, as well as her local security policies in terms of the security classification of personal data and clearance of known OWL-S services as input. It then attempts to discover services that are semantically relevant to the request using the integrated OWLS-MX matchmaker.

In addition, it collects the corresponding security types published by the respective web service provider agents. If the matchmaker finds services detected as being equivalent to the requested one, it directly passes the top ranked one according to its QoS value to the security checking module for letting it verify whether its published security policy complies with both given local security policies and the web service's security type.

If no equivalent service is found, the OWLS-MX module passes the set of services to its composition planner, named OWLS-Xplan. In case a composition plan with more than one service is generated, the compliance of published security types of all web services involved in the plan is checked against the local security policies of the user.

In contrast to usual access control mechanisms, the security checking of the SCPA relies on type-based

information flow analysis. Thereby, the approach also includes dynamically computed data of web services and their security classification, and its proliferation to other services. In any case, the composition plan gets executed only if the security types of all web services meet the local security policies. So, the plan as a whole is formally verified as being secure. Otherwise the SCPA triggers a re-planning activity to be performed as follows. The security checker provides the matchmaker module with a set  $F(P)$  of services of plan  $P$  that caused  $P$  to not comply with the local security policies, in order to select one semantically equivalent service with a different published security policy for each or at least some of them. If successful, the composition planner simply modifies the original plan considered by replacing each service in  $F$  by its substitute, and returns the modified plan to the security checker for verification. If there exists no services in  $F(P)$  for which equivalent service can be found (and which are not yet tried), the composition planner generates a new plan by means of heuristic replanning [9]. In any case, if the modified plan is also provably insecure, the SCPA repeats the same procedure until a secure composition plan is generated, or it returns a failure otherwise. More details on the security checking of OWL-S service plans generated by OWLS-XPlan are provided in [18].

## 6. Related work

There exist quite a few different approaches to service composition planning in the literature. They can roughly be classified into process oriented approaches, and data or signature oriented approaches. Members of the first presume a goal that specifies the global behaviour of the desired service in terms of the set of possible desired conversations, or process flow to be accomplished by synthesizing the process models of available services that can either be modified during composition [2], or not [1]. Specification of the behaviour usually takes the form of FSMs, situation calculus [8], or linear temporal branching logic formulas. Any signature-oriented or data-driven composition approach does not take the process of a service into account but tries to instantiate a goal specification given by the signature of a desired service, i.e. its input/output behaviour together with constraints and user preferences only. Such an instance is a sequence of atomic or other composite services considered as black boxes that accomplishes the goal.

OWLS-XPlan falls into the latter category, and is tightly related to classical planning in AI. An overview of AI planning techniques and their application to the Web service composition planning problem is provided in [9]. An accessible approach to solutions of the problem of cyclic composition planning via model checking is in [14].

The service composition planner that is most relevant to OWLS-XPlan is Shop2 (Simple Hierarchical Ordered Planner 2) developed at the University of Maryland, USA [12]. Shop2 is a hierarchical task network (HTN) planner well-suited for working with the hierarchically structured OWL-S process model. Shop2, like HTN planner in general, replaces those elements of the provided methods (workflows) by special methods or atomic actions until the composition plan contains only atomic actions that correspond to available web services. During planning, web services are not executed, hence do not affect the world state.

Both XPlan and Shop2 base on the closed world assumption, use PDDL, allow external (call-back) functions, and generate totally ordered and instantiated plan sequences for a given initial state, goal and planning domain. However, among others, they differ in their way of planning. In essence, Shop2 plans are generated by use of pre-given decomposition rules (methods), hence a solution to the planning problem is not always guaranteed to be found [7]. In contrast, the hybrid XPlan as part of OWLS-XPlan first tries to plan by means of method decomposition, and if this is not successful, it exploits its relaxed graph plan algorithm to find a solution, if it exists. In addition, for decomposition of given methods it is using only relevant parts by discarding useless actions, thereby reducing the plan size in total.

## 7. Conclusion

We presented the implementation, evaluation, and application of our OWL-S service composition planner OWLS-XPlan. Selected emergency medical assistance related services described in OWL-S 1.1 and corresponding OWL ontologies are converted to initial state and goal descriptions in PDDL 2.1. These are then used by the fast planner XPlan for generating an execution complete composition plan. Results of experimental evaluation of XPlan show its top performance compared with other selected AI planners. OWLS-XPlan is used in an agent based mobile e-Health system for emergency medical assistance planning tasks.

## 8. References

- [1] D. Berardi, D. Calvanese, G. D. Giacomo, M. Lenzerini, and M. Mecella. Automatic service composition based on behavioral descriptions. *Journal of Cooperative Information Systems*, 14(4), 2005.
- [2] T. Bultan, X. Fu, R. Hull, and J. Su. Conversation specification: A new approach to the design and analysis of e-service composition. *Proceedings of World Wide Web Conference WWW, Budapest, Hungary*, 2003.

- [3] I. P. Competition. IPC3. *Homepage: <http://planning.cis.strath.ac.uk/competition/>*, 2002.
- [4] J. Hoffmann. A heuristic for domain independent planning and its use in an enforced hill-climbing algorithm. *Proceedings of 12th Intl Symposium on Methodologies for Intelligent Systems, Springer Verlag*, 2000.
- [5] J. Hoffmann. The Metric-FF planning system: Translating Ignoring Delete Lists to Numeric State Variables. *Artificial Intelligence Research (JAIR)*, vol 20, 2003.
- [6] J. Hoffmann and B. Nebel. The FF Planning System: Fast Plan Generation Through Heuristic Search. *Journal of Artificial Intelligence Research (JAIR)*, (14):253–302, 2001.
- [7] A. Lotem, D. Nau, and J. Hendler. Using planning graphs for solving HTN problems. *Proceedings of AAAI/IAAI conference, USA*, 1999.
- [8] S. McIlraith and T. Son: Adapting Golog for composition of semantic Web services. *Proceedings of International Conference on Knowledge Representation and Reasoning KRR, Toulouse, France*, 2002.
- [9] J. Peer. Web Service Composition as AI Planning: A Survey. *Technical Report, University of St. Gallen, Switzerland Available at <http://elektra.mcm.unisg.ch/pbwsc/docs/pfwsc.pdf>*, 2005.
- [10] M. Schmidt. Ein effizientes Planungsmodul fuer die lokale Planungsebene eines InterRaP Agenten. Master's thesis, Universitaet des Saarlandes, 2005.
- [11] E. Sirin, B. Parsia, D. Wu, J. Hendler, and D. Nau. HTN planning for web service composition using SHOP2. *Journal of Web Semantics*, 1(4), pages 377–396, 2004.
- [12] D. Wu, B. Parsia, E. Sirin, J. Hendler, and D. Nau. Automating DAML-S web services composition using SHOP2. *Proceedings of the 2nd International Semantic Web Conference (ISWC2003), pages 20-23, Sanibel Island, Florida, USA*, 2003.
- [13] M. Klusch, A. Gerber, M. Schmidt: Semantic Web Service Composition Planning with OWLS-XPlan. *Proceedings of the AAAI Fall Symposium on Semantic Web and Agents, Arlington VA, USA*, 2005.
- [14] A. Cimatti, M. Pistore, M. Roveri, P. Traverso: Weak, strong, and strong cyclic planning via symbolic model checking, *Artificial Intelligence*, 147(1/2), pp. 35 - 84, 2003.
- [15] OWLS-XPlan:  
<http://projects.semwebcentral.org/projects/owls-xplan/>
- [16] B. Medjahed, A. Bouguettya, and A.K. Elmagarmid. Composing Web services on the semantic Web. *Very Large Data Bases (VLDB)*, 12(4), 2003.
- [17] M. Klusch, B. Fries, K. Sycara: Automated semantic web service discovery with OWLS-MX. *Proceedings of International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2006), Hakodate, Japan, ACM Press*, 2006
- [18] D. Hutter, M. Klusch, M. Volkamer, A. Gerber: Provably secure execution of composed semantic web services. *Proceedings of the 1st International workshop on Privacy and Security of Agent-Based Collaborative Environments (PSACE), Hakodate, Japan*, 2006

## Advanced Dynamic Service Composition with OWLS-XPlan2

M. Klusch, K-U. Renner: Fast Dynamic Re-Planning of Composite OWL-S Services. Proceedings of the 2nd IEEE Workshop on Semantic Web Service Composition, Hongkong, China, IEEE CS Press, 2006

# Fast Dynamic Re-Planning of Composite OWL-S Services<sup>1</sup>

Matthias Klusch, Kai-Uwe Renner  
German Research Center for Artificial Intelligence  
Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany  
{klusch, Kai-Uwe.Renner}@dfki.de

## Abstract

In this paper, we present an extension of our OWL-S service composition planner OWLS-XPlan that allows for quasi-online re-planning of composite OWL-S services without full restart of the actual planning process, and preliminary experimental evaluation results.

## 1. Introduction

Though the AI based composition planning of complex Web services attracted much interest recently [5, 11], only a few planning tools are actually available for the semantic Web, such as the HTN based composition planner SHOP2 [7, 8, 4], or OWLS-XPlan [9] for OWL-S services. However, none of these planners copes with the open world assumption of OWL, but performs more or less efficient CWA based off line planning.

In open environments, like the semantic Web, non-deterministically occurring events such as broken service links, change of facts, or goal, and availability of new services may affect the actual planning process of a composite service. The actual plan, or parts of it, may become invalid or sub-optimal even before its full generation. Invalid plans may be caused by, for example, services that became unavailable, or facts that satisfied a precondition of some service in the current planning sequence changed such that the semantic compatibility with its preceding service is invalid. Newly introduced services may cause sub-optimality of the current plan in terms of its path length to the given goal state. None of the currently available OWL-S service composition planners does allow for dynamic re-planning, which in turn motivated us to extend our own planner OWLS-XPlan to accomplish this task. Basic idea of OWLS-XPlan+ is to re-use as much as possible of the existing plan such that the minimally modified plan as a whole remains valid in the changed world state. Though the state of the world gets checked for any changes that may affect the current

plan at the end of each plan step, and if so, triggers immediate re-planning off-line, but actions are executed only after a plan has been eventually created that is guaranteed to reach the given goal. This is in contrast to classical on-line planning approaches where typically a planner generates conditional plans that branch over observations, while a controller executes actions in the plan, and monitors observations to decide which branch to execute. Any kind of interleaving framework, in general, cannot guarantee that a goal state will be reached, unless the domain is proven to be safely explorable. Services provided by autonomous providers cannot be assumed to be executable under full control and observation of the planning site, nor to be delivered charge free even in scenarios of tight collaboration with respective service providers. We set the context by briefly introducing our service composition planner OWLS-XPlan in section 2, and then describe the dynamic re-planning by its extended planning module XPlan+ in section 3. We present preliminary experimental evaluation results in section 4, and conclude in section 5.

## 2. OWLS-XPlan Overview

The semantic web service composition planner OWLS-XPlan consists of several modules for pre-processing and planning of composite OWL-S services (cf. figure 1).

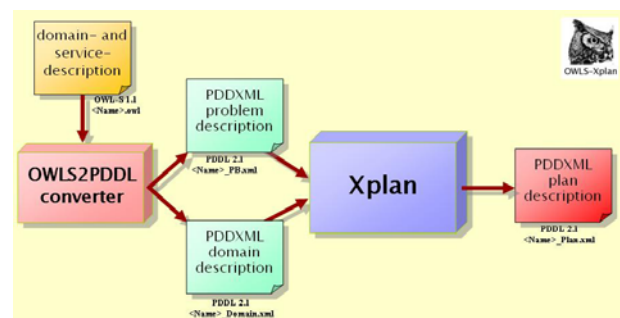


Fig. 1. OWLS-XPlan Architecture

<sup>1</sup> This work has been supported by the German Ministry of Education and Research (BMBF 01-IW-D02-SCALLOPS) and by the European Commission under the project grant FP6-IST-511632-CASCOM.

It takes a set of available OWL-S 1.1 services, related OWL ontologies, and a planning request (goal) as input, and returns a planning sequence of relevant OWL-S services that satisfies the goal. For this purpose, it first converts a given domain ontology and service descriptions in OWL and OWL-S 1.1, respectively, to equivalent PDDL 2.1 problem and domain descriptions using an integrated OWLS2PDDL converter. The domain description contains the definition of all types, predicates and actions, whereas the problem description includes all objects, the initial state, and the goal state. Both descriptions are then used by the AI planner XPlan to create a plan in PDDL that solves the given problem in the actual domain. An operator of the planning domain corresponds to a service profile in OWL-S, while a method is a special type of operator for fixed complex services that OWLS-XPlan may use during its planning process.

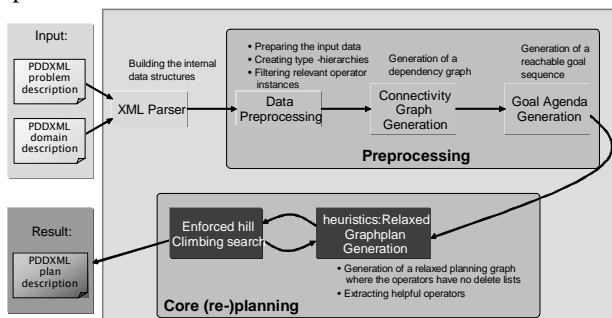


Fig. 2: The planning module XPlan

The planning module XPlan (cf. figure 2) is a heuristic hybrid FF planner based on the FF planner developed by Hoffmann and Nebel [1, 2, 3]. It combines guided local search with relaxed graph planning, and a simple form of hierarchical task networks to produce a plan sequence of actions that solves a given problem. If equipped with methods, XPlan uses only those parts of methods for decomposition that are required to reach the goal state with a sequence of composed services. Due to space restrictions, for more details on OWLS-XPlan in general, and XPlan in particular, we refer the reader to [9]. The sources are available at [12].

### 3. Dynamic Re-Planning by XPlan+

We modified the original XPlan module of OWLS-XPlan to allow for event driven heuristic re-planning of composite services during the actual planning process. The corresponding OWL-S composition planner is called OWLS-XPlan+. The modified planner XPlan+ does perform, in essence, highly frequent event driven off-line re-planning under closed world assumption with heuristic computation of best re-entry points for re-planning at the

end of each planning step if the currently produced plan, or plan fragment gets affected by the observed change. External changes in the world state concern converted OWL ontologies, individuals and the set of available services during the internal planning process each of which potentially affecting the respective operators, actions, predicates, facts and objects in the PDDXML problem and domain descriptions as well as already generated partial plans. For event monitoring, we equipped XPlan+ with an event listener for distinguished classes of events (cf. figure 3).

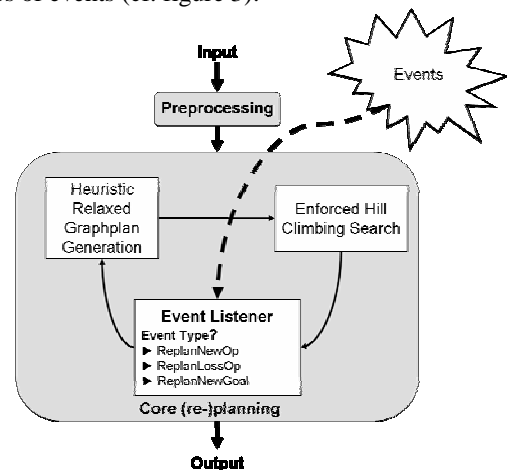


Fig. 3: Modified planning module XPlan+

In each plan step  $i$ , before applying selected helpful action  $A$  to the state  $S_i$ , however, XPlan+ listens for events of state changes. If no events are in its event queue, it applies  $A$  to  $S_i$  and proceeds with plan step  $i+1$ . The plan fragment from initial state  $S_0$  to  $S_i$  is correct and, due to the selection of helpful actions in the minimal relaxed plan, approximated optimal. XPlan+ triggers re-planning in the following cases of observed events of world state changes: (1) An operator (service) instantiation (action) becomes available. This is the case if (a) a new operator has been introduced, or (b) the world state (set of facts) changed such that an operator whose instantiation was impossible before can be instantiated now, or (c) new predicates which are part of the preconditions or effects of an operator are introduced, making it possible to instantiate this operator; (2) An operator (service) of the plan is not possible anymore, if any of the opposites of cases 1.a - 1.c holds; (3) The goal state changed due to a change of the original planning request. Each of these cases is handled separately as described in subsequent sections. If facts or objects change, it searches for the first operator which precondition is satisfied by the new fact, and starts re-planning from there, while the helpful actions get instantiated with the new fact(s). The case in which a predicate  $p()$  changes can be reduced (a) to the latter case of changed facts, if new facts are added; (b) to the case of

change of operator  $o$ , if preconditions or effects of  $o$  include  $p()$ ; or (c) to the case of fact changes, if the deletion of  $p()$  implies the deletion of all instances of  $p()$ . It is assumed that the planning state consistency is checked by means of an appropriate module as integral part of both XPlan and XPlan+.

### 3.1. Case of new operator

If a new operator (service) becomes available, XPlan+ first checks whether re-planning might yield a shorter plan by comparing the old with the newly generated relaxed plan. If positive, it heuristically determines the point in the plan where the new operator might first be helpful to start the re-planning from there.

**Re-planning decision.** XPlan+ first uses the same initial state as for the original (partial) plan  $P$  plus the new operator  $o$  to build the relaxed plan graph RPG, and extract a new relaxed plan  $RP'$ . Second, it estimates the length  $h(RP')$  of  $RP'$  by applying the same relaxed plan length heuristic as done for determining  $h(RP)$ , that is the sum of all actions in all action-layers of the  $RP'$ . It is the number of all (helpful) actions of  $RP'$  as solution paths in the RPG for the initial state. If  $h(RP') < h(RP)$  holds, it continues with re-planning. Otherwise no re-planning is performed.

**Re-planning.** How much of the old plan  $P$  can be reused for the new plan  $P'$ , means what would be the best position to restart planning with the new operator  $o$ ? In order to determine this position, XPlan+ heuristically takes the index of the layer of the new  $RP'$  at which  $o$  occurs first as position  $e$ , else (if  $o$  is not in  $RP'$ ) sets position  $e = -1$  and stops, and retains the old plan (fragment)  $P$  until this position. In other words, the position  $e$  of starting re-planning is the minimal number of actions before first occurrence of  $o$ . Second, it applies all operators from the old plan occurring before  $e$  in the new plan, and then tries to identify the instances of  $o$  which are applicable to the current state. For this purpose it checks whether the precondition of  $o$  is satisfied in the  $RP'$ . If no instance of  $o$  is applicable, it tries to apply more operators from the old plan until an instance of  $o$  eventually becomes applicable. If this fails, a complete re-planning has to be started. Otherwise it applies the new operator  $o$ . That is, XPlan+ identifies a re-entry point in the original plan by searching for already planned operators (actions) which correspond to helpful ones in the current state, and continues with the first step from this position. If no such position can be found, start full re-planning of the plan. If the goal is not yet reached, extend the plan until the goal is reached by continuing with the normal planning process like XPlan.

### 3.2. Case of lost operator

If a planned operator becomes unavailable, the actual plan is invalid. XPlan+ tries to replace the affected

operator(s) by replacing it with alternative ones which achieve the same effect as the lost one. In case of success, the remainder of the plan can be re-used, which reduces re-planning time significantly.

**Re-planning decision.** XPlan+ first marks all actions in the plan which are affected, because of the fact that the respective operator does not exist anymore, or some precondition does not hold anymore. If no actions are marked, it continues with the normal planning process like XPlan.

**Re-planning.** For each affected action, XPlan+ creates a relaxed plan  $RP'$  from  $S_0$ . It then uses (inverse) enforced hill climbing search to circumvent the affected operator by applying alternative operators if possible. Basically, the planner identifies a re-entry point in the old plan  $P$  by searching for already planned actions in  $P$  which correspond to helpful actions in the current state, and continues with re-planning from this position. If no such position can be found, the remainder of the plan has to be re-planned completely. Otherwise, if the goal is not yet reached, extend the plan until the goal is reached by continuing with the normal planning process like XPlan.

### 3.3. Case of new goal

If the given planning goal did change, re-planning is necessary in case the new goal cannot be satisfied by the current plan at all, or could even be achieved by a shorter plan.

**Re-planning decision.** XPlan+ quickly creates a relaxed plan for the new goal from the initial state  $S_0$ , and marks all actions in the already existing plan  $P$  which are also contained in the new relaxed plan.

**Re-planning.** For each non-marked action, XPlan+ uses enforced hill climbing search to circumvent the action by applying alternative operators, identifies a re-entry point in the old plan by searching for planned actions in the old plan  $P$  which correspond to helpful actions in the current state, and continues planning from this position. That is, XPlan+ starts heuristic re-planning with the action in currently valid  $P$  that precedes first occurrence of  $o$ . If no such position can be found, the remainder of the plan has to be re-planned completely. If the goal is not yet reached, XPlan+ extends the plan until it is reached by continuing with the normal planning process like XPlan.

## 4. Preliminary Evaluation

The comparative analysis of the computational complexity of planning with XPlan+ and XPlan, as depicted in figure 5, is concerned with situations where a new operator becomes available, or an operator is deleted just before the initial planning is finished (for plans with at least 20 steps). The denotation "Online( $n$ )", with  $n =$

0,1,2 refers to the case where an observed event does affect the plan at n positions. As a consequence, XPlan+ has to build n relaxed plans during its partial re-planning, whereas the pure off-line planner XPlan denoted in the figure as "Offline" would do a full re-planning.

Planning Time Offset in %

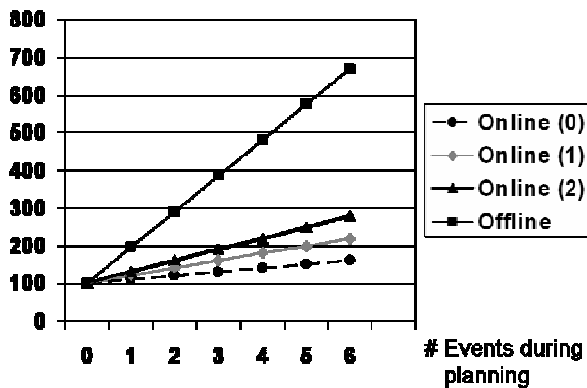


Fig. 4: Computational complexity of planning with XPlan+ (quasi-)online vs. XPlan (full restart/offline)

The resulting planning time offsets for all cases applied to a simple blocks world related plan of 28 steps with initially five operators is shown in figure 4. Only new operators were introduced during the planning process that theoretically would lead to a shorter plan. XPlan+ gained more momentum compared to XPlan with increasing number of such events, and the later in the plan they did occur. This is also experimentally confirmed by the measured run time of XPlan+ (cf. figure 5) which decreased in absolute terms mainly due to its heuristic re-use of plan parts as described above.

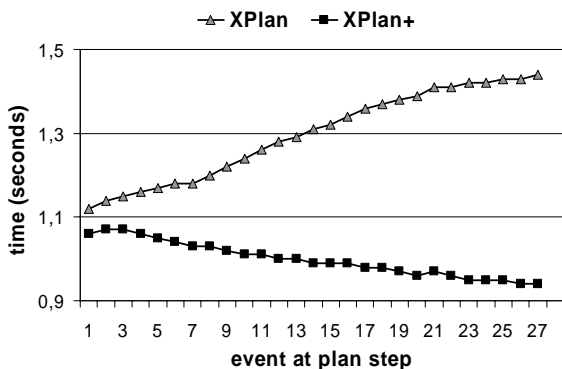


Fig. 5: Measured run time of XPlan+ vs. XPlan

## 5. Conclusion

We presented an extension of the planning module of our OWL-S service composition planner OWLS-XPlan, named XPlan+, that allows for quasi-online re-planning of

composite OWL-S services with reasonable performance according to preliminary evaluation results. We are currently working on the integration of the implemented XPlan+ into OWLS-XPlan, and plan to make the resulting service composition planner OWLS-XPlan+ publicly available at [semwebcentral.org](http://semwebcentral.org)

## 6. References

- [1] J. Hoffmann. A heuristic for domain independent planning and its use in an enforced hill-climbing algorithm. *Proceedings of 12th Intl Symposium on Methodologies for Intelligent Systems, Springe*, 2000.
- [2] J. Hoffmann. The Metric-FF planning system: Translating Ignoring Delete Lists to Numeric State Variables. *Artificial Intelligence Research*, 20, 2003.
- [3] J. Hoffmann and B. Nebel. The FF Planning System: Fast Plan Generation Through Heuristic Search. *Artificial Intelligence Research*, 14, 2001.
- [4] A. Lotem, D. Nau, and J. Hendler. Using planning graphs for solving HTN problems. *Proceedings of AAAI/IAAI conference, USA*, 1999.
- [5] J. Peer. Web Service Composition as AI Planning: A Survey. *Technical Report, U St. Gallen, Switzerland* <http://elektra.mcm.unisg.ch/pbwsc/docs/pfwsc.pdf>, 2005.
- [6] M. Schmidt. Ein effizientes Planungsmodul fuer die lokale Planungsebene eines InteRRaP Agenten. Master Thesis, U Saarland, Germany, 2005.
- [7] E. Sirin, B. Parsia, D. Wu, J. Hendler, D. Nau. HTN planning for web service composition using SHOP2. *Journal of Web Semantics*, 1(4), 2004.
- [8] D. Wu, B. Parsia, E. Sirin, J. Hendler, and D. Nau. Automating DAML-S web services composition using SHOP2. *Proceedings of the 2nd International Semantic Web Conference (ISWC2003), Florida, USA*, 2003.
- [9] M. Klusch, A. Gerber, M. Schmidt: Semantic Web Service Composition Planning with OWLS-XPlan. *Proceedings of the AAAI Fall Symposium on Semantic Web and Agents, Arlington VA, USA, AAAI Press*, 2005.
- [10] L. Pryor, G. Collins. Planning for Contingencies: A Decision-based Approach. *Artificial Intelligence Research*, 4:287-339, 1996.
- [11] B. Medjahed, A. Bouguettya, A.K. Elmagarmid. Composing Web services on the semantic Web. *Very Large Data Bases (VLDB)*, 12(4), 2003
- [12] OWLS-XPlan: <http://projects.semwebcentral.org/projects/owls-xplan/>
- [13] R. Dearden et al.. Incremental Contingency Planning. Proc. Int. Conf. on Automated Planning and Scheduling ICAPS, Workshop on Planning under uncertainty and incomplete information, Trento, Italy, 2003



## Agent-Based Service Negotiation



---

## Introduction

In competitive environments, business application services may not be for free but available via pay-per-call only. For example, service agents could be charged for every single invocation of a Web service according to selected flat-fee or differentiation-based pricing models. As a consequence, service consumer and provider agents have conflicting interests, in principle, that are minimization of service charges, respectively, maximization of profits.

### Service Negotiation in the Web and Semantic Web

Standard solution of this problem in service-oriented computing is to negotiate the terms and conditions of the service usage between service provider and consumer agents in mutually beneficial, so-called extended contractual service level agreements (SLAs). Such agreements specify guarantees for (a) the delivery of certain functionalities of configurable services, and (b) the non-functional service qualities concerning the availability of the service, throughput and latency bounds based on mutually agreed measures, respective pricing, and privacy policy<sup>1</sup>.

In the context of the Semantic Web, semantic service composition planning and discovery as described in the previous parts can be performed by either the service requester agent, or the provider agent, or distinguished middle agents on behalf of either parties. A composed service can be executed as a whole, when the terms and conditions of the execution of its usually configurable subservices are successfully negotiated. In principle, service discovery and composition can be performed before entering the negotiation of relevant

---

<sup>1</sup> Note that in computer networking literature, the traffic-engineering term SLA is restricted to non-functional service quality guarantees for network service consumers that is mostly achieved by prioritizing traffic for considered services such as streaming multimedia applications, video conferencing (VTC), or Internet telephony (VoIP).

services with the respective providers, or in an interleaved fashion during negotiation.

However, in most approaches to agent-based service negotiation, the negotiation phase between the agents starts after the discovery of existing or composed service candidates that semantically match a given user query, and ends with a set of mutually binding, enforceable SLAs, that are signed service contracts (Preist, 2007)[299] (cf. chapter 1, section 1.3.4)<sup>2</sup>. To the best of our knowledge, the problem of dynamically interleaving semantic service negotiation with composition planning at run time to improve the efficiency and quality of the result of both planning and negotiation of relevant services remains open; research on this field has just started.

In general, existing approaches to agent-based service negotiation in the literature can be classified with respect to the following criteria.

- *What is being negotiated?* Agents can negotiate the terms of using single or multiple different concrete services (single-item vs many-item negotiation) in the respective SLA(s) based one single or multiple non-functional and/or functional service parameters (single-issue vs multiple-issue negotiation).
- *How is service negotiation be performed?* What negotiation model and protocol is used by the agents to reach an agreement? Prominent micro-economic negotiation mechanisms for trading goods are bargaining, equilibrium markets, auctions, contracting, and coalition formation. Negotiation items can be any kind of goods, tasks, resources, data sets, time, and concrete services. Second, what kind of interaction between negotiating agents is allowed by the mechanism such as one-to-many, many-to-one or many-to-many negotiation? Third, what are the guaranteed features of possible agreements according to the chosen negotiation model and protocol? Are the negotiation solutions individual rational and Pareto-optimal? Are negotiated coalitions stable with respect to given criteria? Is the negotiation protocol safe against manipulation via deceit and fraud? Other issues are the preservation of data privacy with minimal loss of profit, incentive compatibility, trust, and dynamic reaction to changes of the environment during negotiation. Finally, how does the negotiation mechanisms scale in practice like for large domain-specific virtual markets with potentially thousands of agents and services?

### Negotiation Models for Rational Agents

In general, a negotiation model specifies the problem domain in which rational agents are supposed to reach an agreement on the distribution of goods and payoffs by means of a chosen negotiation protocol. The choice of a negotiation protocol depends on whether it allows the agents to solve the given negotiation problem, and what features the system designer wants the overall agent system to exhibit in the considered environment.

---

<sup>2</sup> In the literature, the notion of a binding SLA and service contract is often used interchangeably.

### *Negotiation problem domain*

The specification of the negotiation problem domain by the system designer concerns assumptions and constraints about the negotiation environment and desired kind of agreements:

- (a) *Negotiation environment*: This concerns, for example, the number and type of trading items and agents (e.g., self-interested, non-cooperative, or collaborative agents; agent capabilities, responsibilities, and knowledge about other agents), the preference modeling and respective utility theory, and the norms, policies and rules of agent interaction (e.g., one-to-one, one-to-many, or many-to-many, or mixed negotiations).
- (b) *Negotiation agreement*: This concerns the desired features of the solution. For example, whether a negotiated agreement between the agents (SLAs and payoff distribution) shall be (non-)binding, exogeneously (not-)enforceable by third party, Pareto-optimal, individual rational, or social-welfare maximizing.

### *Choice of negotiation protocol*

A negotiation protocol (also called mechanism) is used by agents to solve a given negotiation problem in a cooperative or non-cooperative fashion. For example, in cooperative distributed problem solving (DPS) settings, all agents work on a commonly shared goal, and are designed to help each other. That is, in any DPS-based negotiation protocol like the prominent contract net protocol (CNP), the collaborative agents use strategies imposed by the system designer to jointly accomplish the goal, hence maximize the social outcome or welfare. In competitive multiagent negotiation settings, agents are provided with a negotiation protocol which determines the interaction and possible actions or strategies but, in contrast to DPS settings, individually select the best strategy for their own without concern for the social welfare. A negotiation protocol can be evaluated with respect to its (a) communication and computational complexity, and (b) the guaranteed properties of payoff distributions as a solution to the given negotiation problem such as

- Social welfare, that is the sum of all agents' payoffs in a given solution,
- Pareto optimality, means that there is no solution other than the negotiated one such that each agent is better off and no one is worse off;
- Individual rationality, that is an agent's payoff in the negotiated solution is at least as high as it would get when not participating in the negotiation;
- Stability, that is no agent is better off behaving differently than specified by the solution such as breaking a coalition as its member due to its assigned payoff.

A negotiation protocol is said to be incentive-compatible if truth-telling (of the individual valuation of negotiated items) forms a Bayesian-Nash equilibrium,

and strategy-proof if truth-telling is a dominant strategy. Thus, strategy-proofness is a stronger property. However, for many kinds of negotiation, it has been shown that efficient, incentive-compatible, budget-balanced (i.e., all payments between agents sum to zero) and individually rational mechanisms do not exist, or are at least very hard to find (see e.g. [269, 266, 233]). Individual quantitative utility functions are used to determine individually optimal strategies during negotiation based on given user preferences and valuations. Mechanisms for negotiation under uncertainty deal with uncertain, partial, tentative or generic information such as the kind and valuation of traded items, and the payoff distribution to the agents involved. These mechanisms usually make use of the notion of expected value of information (Howard, 1996)[166] and utility [375], as well as possibility or fuzzy theory. Some mechanisms allow the agents to determine which negotiation strategy would be more successful by means of Bayesian learning (Sycara & Zeng, 1997)[354], mixed evolutionary computing and case-based reasoning (Matos & Sierra, 1998)[251], and fuzzy similarity rules (Sierra et al., 1999)[336].

In the following, we briefly introduce the prominent negotiation models of bargaining, auctions, markets and coalition formation together with selected representative examples of their application to rational agent-based service negotiation. Each of these models can be used by self-interested service provider agents in order to maximize their individual profits without concern for the social welfare. Such self-interest naturally prevails in negotiations among agents on behalf of autonomous businesses or individuals. We comment on the principled interrelation between semantic service composition and negotiation, and then present our contributions to the field, that are negotiation protocols for safe, privacy preserving, and dynamic coalition formation among service provider agents.

## Bargaining

Game-theoretic bargaining theory deals with situations in which pairs of competing agents on a market try to make a mutually beneficial agreement about how to distribute a given good such as an abstract objective, a herd of sheeps, Web service values, or monetary amount, but have a conflict of interest about which agreement to make (Sandholm, 1999)[323]. Besides, parts of the considered good may become subject to bargaining during different stages of the negotiation.

The agents have to decide on an object distribution, that is an agreed outcome  $o \in O$  of the bargaining, or a fixed fallback outcome  $f \in O$  occurs when no such agreement is reached. It is assumed that the preferences of each agent  $i$  on the possible outcomes  $o \in O$  can be represented by a von-Neumann-Morgenstern utility function ( $u_i : O \rightarrow R$ ). Both agents try to maximize their utility by means of bilateral (one-to-one) non-cooperative negotiation, that is without binding agreement or forming coalitions with other agents. Unlike non-cooperative games (von Neumann & Morgenstern, 1944)[375], bargaining

games are also cooperative in the sense that the agents can (a) make a binding agreement on the fallback outcome in prior that is exogeneously enforceable, and (b) communicate with each other in several rounds in so-called sequential (strategic) bargaining games to find a solution.<sup>3</sup> There are two major models of bargaining theory: axiomatic bargaining, and strategic bargaining.

#### *Axiomatic bargaining*

In axiomatic bargaining, each agent is supposed to make an individually rational choice between possible agreements such that the negotiation solution satisfies certain axiomatic imposed properties. The behavior of the agents in terms of their strategies in the game is modelled only implicitly by the desired features of the agreement: The axiomatic bargaining game abstracts from the respective bargaining process, that is the exchange of offers and counter-offers, and the making of concessions. The 2-agent Nash bargaining solution is a prominent example.<sup>4</sup>

Other bargaining solutions postulate different desiderata in form of axioms with different utility combination as outcome. However, unlike non-cooperative games, axiomatic bargaining games are often not considered appropriate to explain the behavior of rational utility maximizing agents, since these games are not based on what individual strategies agents could choose to reach some form of equilibrium. This is done in strategic bargaining, a rather more popular variant of bargaining in practice.

#### *Strategic bargaining*

In strategic bargaining (Sutton, 1986)[353], rational agents enter a sequential bargaining game with potentially infinitely rounds of alternating offers and counter-offers in a prespecified order until an agreement is reached, or not. Prominent solutions are the Rubinstein game with a subgame perfect Nash

---

<sup>3</sup> In the literature, bargaining games are termed both a cooperative game in a general sense, and a non-cooperative game - with additional constraints that determine the result of a cooperative game like a binding contract and disagreement payoffs. Every cooperative game can be represented as a sequence of non-cooperative games. Most readings on the subject use the term "non-cooperative bargaining"; a compromise is the term "individualistic-cooperative game" (Holler & Illing, 2000)[157].

<sup>4</sup> Nash bargaining game  $(F, d)$  is defined by a set  $F$  of possible agreements, that are the possible joint utility allocations  $(u_1(o), u_2(o)) \in [d, z]$ ,  $z$  is the total good, to the agents  $i$ , and the disagreement point  $d = (u_i(f))_i$ ,  $i \in \{1, 2\}$  such that if  $u_1(o) + u_2(o) \leq z$ , agents 1, 2 receive  $u_1(o)$ , respectively  $u_2(o)$ , otherwise both get  $d$  (often  $d = 0$ ). The bargaining game can have possibly many Nash equilibria for individually rational agents but has a unique Nash bargaining solution  $o^* = \operatorname{argmax}_o \{(u_1(o) - u_1(f)), (u_2(o) - u_2(f))\}$  that satisfies the axiomatic imposed properties of Pareto efficiency, symmetry, independence from irrelevant alternatives, and invariance to equivalent utility representations.

equilibrium reachable in the first round by use of a time-based discount factor of utilities <sup>5</sup>, and the two-round Zeuthen-Harsanyi game with Nash equilibrium solution.

In (Kraus, 2001)[219], other sequential bargaining games for strategic negotiation of multiagent systems with several applications are presented. These mechanisms allow for bargaining (a) without using discounts but fixed bargaining costs per negotiation round, (b) over time without perfect rationality and information (as assumed in one-shot bargaining) when agents do not fully comprehend the space of deals, in particular do not know each others' types including individual capabilities and preferences, and (c) cases where one agent gains and loses over time. Sandholm (1999)[323] remarks that strategic bargaining models should trade off the bargaining gains with the computational costs of two kinds of searches: The intra-agent deliberative search (e.g., local generation of offers, alternatives, evaluating them, counterspeculating, and doing a lookahead in the negotiation) and the inter-agent committal search (e.g., iterative re-negotiation of agreements) for an agreement in which the agents' strategies are in equilibrium.

For a more comprehensive coverage of strategic bargaining, we refer to the standard literature on game theory like (Osborne & Rubinstein, 1994; Holler & Illing, 2000)[282, 157]. Different strategic negotiation models for multiagent systems to resolve conflicts on the allocation of items such as tasks, time, or data sets in different environments and real world applications are presented in, for example, the excellent volumes (Rosenschein & Zlotkin, 1994)[319], and (Kraus, 2001)[219].

#### *Examples of agent-based service bargaining*

Jonker et al. (2007)[178] propose a bilateral bargaining protocol for multi-attribute negotiation under incomplete preference information of the agents. For each non-functional service attribute except price, given attribute evaluation values, and importance factors are used to compute an overall non-financial utility (so-called ease utility), and a normalized financial utility is employed. These two utilities are then combined into an overall utility with a rationality factor, which allows service agents to negotiate over the price and other service qualities simultaneously. Besides, agents may selectively disclose their preferences over the negotiated attributes to other agents in order to (a) prevent the misuse of known preference information by other agents to get a better deal, and (b) to satisfy externally imposed privacy requirements. Using this protocol agents can heuristically assess the other agents' preferences based on the changes in their offers such that, as shown by experiment, the reached agreements are close to Pareto optimal.

---

<sup>5</sup> The Rubinstein bargaining solution of the 2-agent bargaining game is  $(u_1, u_2)$ ,  $u_1 = (1 - \delta_1)/(1 - \delta_1\delta_2)$ ,  $u_2 = 1 - u_1$ , with time-based discount factor  $\delta_1$  ( $\delta_2$ ) for agent 1 (2).



However, this neither prevents agents from behaving maliciously nor from correctly guessing each others' preferences, thereby violating both data privacy and incentive compatibility. Further, the protocol allows bilateral (one-to-one) negotiation only, hence does not allow agents to coordinate their negotiation of multiple, interdependent services of a compound service with respective providers (one-to-many). Thus, an agent risks to fail bargaining all services involved in its service composition plans.

Hung et al. (2004)[171] propose a declarative XML-based language for specifying one-to-one (bilateral) alternating offers protocols for bargaining of Web services in terms of negotiation message formats, chosen negotiation protocol, and negotiation decision-making by each agent. The specification of decision making consists of two parts: The actual negotiation strategy based on a cost-benefit (utility) model for each agent which is kept private, and an agreement template used to construct the offers and counter-offers exchanged during the bargaining process. This template is then incrementally refined into a binding service level agreement (SLA) at the end of a successful negotiation.

However, the approach covers negotiation of SLAs over multiple issues including price via bilateral bargaining, but leaves the choice of the actual negotiation protocol open. In particular, it does not solve the problem of coordinating the negotiation of multiple interdependent services of a service composition plan.

Dang and Huhns (2006)[82] propose a protocol service bargaining agents that allows them to concurrently perform multiple bilateral negotiations of multiple services, that are so-called many-to-many bilateral service negotiations. This alternating-offers protocol for bargaining services with SLAs over multiple (non-functional) issues including price is defined as a Colored Petri Net for modeling state changes of the negotiation process, and is proven to terminate under certain constraints on used offer generation and evaluation methods (such as minimum increments of an offered price, and automatic rejection if this is not abided by). One important feature of the protocol is that agents can decommit from pre-accepted proposals and agreements without penalty payments.

However, no methods for offer generation and evaluation strategies are presented such that it, in essence, remains unclear how bilateral service bargaining will proceed in practice.

Rahwan et al. (2002)[304] propose a bargaining protocol that allows an agent to coordinate its multiple concurrent single service negotiations other service trading agents. For this purpose, each agent ("master") controls a set of "slave" agents, that are one coordinator agent, and buyer and seller agents, one for each service it wants to purchase from, or sell to other master agents. The respective bargaining between buyer and seller agents on behalf of different master agents is controlled by the respective coordinator agents according to individual negotiation strategies of the master agents. In the proposed alternating-offers protocol, buyer and seller agents generate and evaluate offers on behalf of their master agents by means of multi-attribute utility theory

and constraint-based reasoning. This allows agents negotiating services with SLAs over multiple (non-functional) issues including price and quality. However, from the informal description of the protocol and the 3-agent example the actual decision-making of the "master" agents upon pre-agreements made by their "slave" agents (buyers and sellers), and the proceeding of negotiation in more complex settings remains unclear.

### **General Equilibrium Markets**

An efficient allocation of goods and resources can be performed in a more general and distributed way based on market prices or quantities (of commodities, or resources) by means of so-called general equilibrium market mechanisms. On such markets, prices for commodity goods may change, and the agents providing (producer) and requesting (consumer) these goods may change their behavior but actual production and consumption of goods only occurs once the market has reached a general (so-called Walrasian) equilibrium. The Walrasian equilibrium is reached if the market clears, and, given the prices, the consumers and providers maximized their preferences, respectively, profits.

General equilibrium solutions for markets exist under certain conditions (but may not be unique), are Pareto efficient and coalitional stable (in the sense that no subgroup of consumers can increase their utilities by deviating from the actual equilibrium and forming their own market). Prominent example of a price-based market mechanism is the distributed price tatonnement algorithm for distinguished price adjuster, service (good) consumers, and service (good) providers. For more information on general equilibrium theory for markets from microeconomics, speculative strategies in, and general properties of equilibrium markets, we refer to the vast literature on the subject, in particular the relevant part of the survey (Sandholm, 1999)[323].

However, the computational efforts of each self-interested agent to generate its optimal supply and demand decision given the current price at every iteration of the market protocol (in case of price change) might hinder them to even participate in such markets from the very beginning. On the other hand, the general equilibrium approach allows one to build reasonably non-manipulatable platforms for agent-based service negotiation. There is a small but quite active research community working on market-based multiagent systems for some time (Wellman & Wurman, 1998). However, equilibrium market mechanisms have not been used for agent-based service negotiation like for speculative open Web service markets yet.

### **Contracting**

The notion of contracting originates from economics and game theory, where it models contracts between two agents only. More concrete, an agent may try to contract out some of the tasks that it cannot perform by itself, or that may be performed more efficiently by other agents. For this purpose, the first

agent (or principal) specifies the terms of the contract, i.e., how much it is willing to pay for certain actions performed by the second agent (or actor) on its behalf. The actor then chooses what actions it will perform, while the principal pays a fee based on its observations, and the contract to the actor (cf. chapter 9 in (Kraus, 2001)[219]).

In the multiagent systems literature, the concept of contracting was first used to delegate or contract tasks to other agents for distributed problem solving (DPS). A prominent example is the contract net protocol (CNP)[341]. As mentioned above, in a DPS domain, agents collaborate to achieve a common goal by means of (recursive) task decomposition, distribution, and solution synthesis, thereby maximizing the social outcome or welfare. Issues of self-interest such as individual rationality and strategic negotiation are not relevant for DPS-based contracting approaches such as the CNP, hence are not directly applicable in competitive domains.

#### *Examples of agent-based service contracting*

Only recently, collaborative multi-agent contracting has been extended to competitive business service domains such as logistics. One prominent example is Sandholm's extension of the CNP to a competitive transportation domain (TRACONET) [324]. According to TRACONET, the CNP initiator agents issue a request for proposals with reservation prices, and the participating agents only bid on requests if they can offer a transport (service) cheaper than these prices. Another example is Sandholm and Lesser's leveled commitment contracting, which allow agents to decommit from contracts by paying a pre-determined penalty (Sandholm & Lesser, 2001)[325]. However, the authors do not provide concrete means for negotiating the conditions of such decommitment or contract-termination.

Vokrinek et al. (2007)[376] introduce another extension of the original CNP for competitive environments that mixes strategic bargaining with leveled commitment contracting. On one hand, the initiator of the CNP may not only accept or refuse proposals from participants, but also make counter-proposals. Thus, a bilateral bargaining process is integrated into the protocol, enabling the negotiation of price and other quality levels of a service. On the other hand, the protocol does not conclude with the contracting phase, but follows the leveled commitment contracting approach by means of an optional decommitment and termination phase. In the decommitment phase, the initiator or any participant can propose to decommit from the contract, whereupon a negotiation over the respective decommitment penalty is entered. Unfortunately, no concrete strategies for each of the above mentioned contracting phases is proposed by the authors. Besides, the protocol still requires any service agent to obtain all services of a given individual composition plan but lowers its financial risk in case of failing to do so: It can possibly decommit from already made service contracts at penalty costs that are lower in total than the contracted price of these services. However, the analysis of

concrete strategies for agents using this protocol is missing such that its usefulness in practice remains unclear.

## Auctioning

Auction theory (Wolfstetter, 1996)[383] analyzes protocols and agents' strategies in auctions. An auction is a price-fixing mechanism of an auction house in which negotiation is subject to a very strict coordination process. In this process, an auctioneer wants to mediate the exchange of goods or items between providers and requesters for sale at the highest possible price over a given reservation price, and potential bidders want to buy them at a lowest possible price. Any auction is a sequence of bidding rounds. Asynchronous bidding mechanisms are mostly based on open-outcry with price changes or sealed bids with their periodic partial revelation. The private value of an item depends only on the individual agent's preferences while its common value is the agent's value of the item determined by the values of other agents for it. The correlated value of an item depends partly on the agent's own preferences and partly on others' values for it. Reverse auctions are initiated by requesters themselves to buy relevant items offered by providers who act as bidders to sell the items for the lowest price possible.

Any auction may be classified along three dimensions of (1) the bidding rules including, for example, bid format, and one-to-many or many-to-many participation, (2) the clearing policy that concerns pricing, clearing schedule and closing of the auction, and (3) the revelation policy for information like price quotes and quoting schedule.

### *Prominent auction protocols*

Prominent examples are the following one-to-many auction protocols.

- First-price, open-cry, so-called *English auction*: The bidders successively raise a bid for an item until one bidder remains. The winner is the last bidder remaining at the price of the second-highest bidder. The dominant strategy for consumers here is to bid up to their true (private), maximum value, then drop out.
- Descending price, open-cry, so-called *Dutch auction*: The auctioneer calls out a descending price for an item and the bidders call out their bids in response. The winner, however, is the first bidder to call out at a price bid. Optimal strategy is to bid just below the private value of item. This auction mechanism guarantees the auctioneer the sale of items at highest possible price.
- *First-price, sealed-bid auction*: Each bidder submits one sealed bid in ignorance of all other bids. The highest bidder wins and pays the amount of her bid. This has the potential to force buyers and seller into price wars since the sealed bid of any bidder depends on what she believes of all other opponents bids.

- Second-price, sealed-bid, so-called *Vickrey auction*: The winning bidder pays the price of only the second highest bid.

Auctions in which multiple (identical or different) items are for sale in so-called bundles include the one-to-many combinatorial auction, the many-to-many double, and the matrix auction. In *combinatorial auctions*, bidders can bid for such bundles or combinations of items. This is particularly useful in situations in which the value of some item to a bidder depends on which other items the bidder can get in the auction. However, main problem of combinatorial auctions is the NP-complete computation of the revenue-maximizing set of nonconflicting bids by the auctioneer.

#### *Winner's curse and security issues*

Assuming the private value of auctioned items, any of the above listed types of auctions yields the same expected price and revenue for the seller when the participating bidders are not risk-averse but risk neutral and symmetric (means they use the same measurements to estimate their valuations). However, revenue equivalence does not hold true under the common value assumption when bidders have similar valuations. Furthermore, bidders suffer, in principle, from the so-called winner's curse, that is, the winner of any auction always offers (and has to pay) the winning bid for an item that is higher than its (actual) value such that any auction is basically a win-lose game.

Main security issues of auctioning for bidders are so-called shills, lying auctioneers, and the revelation of private values of items. On the other hand, collusions of bidders and shills are illegal but hard to detect by any auction house in practice. In particular shills violate the trust in auctioneers and the integrity of offers for English auctions like in eBay, and all-pay auctions. However, Vickrey, first-price sealed-bid, and Dutch auctions are not vulnerable to shills. Further, a coalition of bidders can legally participate in an auction by means of one distinguished bidder representing the coalition.

For a more in-depth discussion of the pros and cons of different types of auctions, we refer to, for example, (Sandholm, 1999)[323] and (Fischer et al., 1998)[120].

#### *Examples of agent-based service auctioning*

One particular problem of service auctioning, from the perspective of bidders, is the efficient auctioning of all services of a given service composition plan. This can be achieved by participating in appropriate combinatorial auctions, many-to-many double auctions, or multiple auctions with respective services for sale at the same time. There are quite a few approaches to agent-based service auctioning in the literature of which we only present representative examples.

(Preist et al., 2003)[300, 301] present an approach to agent-based service composition through simultaneous negotiation between service consumers, dedicated service composition agents, and service providers in forward and reverse

auctions. Each service consumer agent initiates a reverse English auction for service composition agents to satisfy a complex service request at minimum costs. In turn, the composition agents try to generate service composition plans that satisfy these requests, and to obtain the services of these plans from the respective providers for a fixed price, or through bidding on possibly multiple English auctions that are initiated by service providers to maximize their profits.

Since all auctions are executed simultaneously, the composition agents face two kinds of risk: (a) The risk of winning reverse auctions for which it has not yet obtained all services of its plans that can satisfy the respective requests, and (b) the risk of winning English auctions by service providers before having won any reverse auction for service requests it can satisfy by the respectively generated composition plans. In order to minimize these risks, each composition agent continuously monitors auctions and computes the expected utility for its bidding on a set of auctions (which is a function on the agent's current bids in these auctions, and the expected cost of winning each of these auctions). Simplifying heuristic strategies for deciding on which set of auctions to actually bid on for relevant services reduce the otherwise combinatorial number of options at the cost of optimality.

However, the proposed approach has not been experimentally evaluated. The interleaving of negotiation with composition planning and discovery is static in the sense that the service composition plans are generated by the composition agents from the set of available services before any negotiation with the relevant service providers takes place.

Sandholm (2002)[326] advocates the use of combinatorial auctions for agent-based service negotiation. In a combinatorial auction, providers offer individual services for sale, but requester agents are allowed to bid for individual and combinations of services (so-called service bundles). The auctioneer decides which one of these bids maximizes the revenue for which provider. As a consequence, service agents can reduce their risk to obtain only some but not all services of their composition plans by bidding for the whole set in form of service bundles at once.<sup>6</sup> The author presents a bidding language for combinatorial auctions, and, in particular, a search algorithm that copes with the NP-complete winner determination problem of combinatorial auctions.

The search algorithm performs efficiently in cases where the bid space is sparsely populated, as it is argued to be common in practice. It is also shown that it is impossible to approximate a solution within a finite bound from the optimum in polynomial time in the general case. Further, two expressive bidding languages are introduced, allowing bidders to express both complimentary and substitutable preferences. Finally, it is shown that the so-called

---

<sup>6</sup> Alternatively, reverse combinatorial auctions can be initiated by a service composition agent to find providers that are offering service bundles (as bidders) which would cover either individually or in combination the set of services required to execute its service composition plan at the lowest possible price, if at all.

VickreyClarkeGroves mechanism can be employed such that each bidder's dominant strategy is to bid truthfully.

However, combinatorial auctions are designed for only one auctioneer, or a set of collaborating auctioneers. Thus, in situations where a service composition consists of services which are offered by competing providers, a bidder still has to participate in and win more than one auction at respective costs and risk of succeeding. This weakens the original advantage of combinatorial auctions over single-item auctions.

In the European research project AgentCities, my research team at DFKI developed and actually deployed an agent-based online auction house (AD-MIT) in which registered agents can concurrently participate in multiple one-to-many auctions (but not combinatorial) auctions. The integrated payment services were developed and hosted at EPFL in Lausanne, Switzerland.

## Coalition Formation

Self-interested service provider and consumer agents may form rational coalitions to maximize their individual payoffs by coordinating their activities with other agents. Cooperative game theory offers solution concepts, so-called coalition theories, to the problem of what coalition to form among individually rational agents with what stable joint payoff distribution. It does not provide any mechanism for agents to actually negotiate these coalitions; coalition formation protocols are developed in multi-agent systems research.<sup>7</sup>

In the following, we briefly introduce to cooperative game theory, comment on possible types of coalitions between service provider and consumer agents, and provide examples of agent-based coalition negotiation protocols. For further readings on the subject, we refer to the standard literature on cooperative game theory, and the excellent readings of Kahan and Rapoport (1984)[180] in particular.

### *Classic coalition games*

In game theory, a cooperative (or coalition) game  $(A, v)$  in normal form is defined by a set  $A$  of agents, and a characteristic function  $v$  that assigns each subset  $C$  (coalition) of agents in  $A$  its maximum profit, the so-called coalition value  $v(C)$ <sup>8</sup>. A coalition value shall not depend on the actions of agents

---

<sup>7</sup> In the literature, the interpretation of the term coalition often differs from the utility driven principle of "bellum omnium contra omnes" favored in game theory. Alternative approaches to cooperation often rely instead on the collaborative use of complementary individual skills to enhance the power of each agent to accomplish its goals such as in team formation.

<sup>8</sup> In other words, the value of a coalition  $C$  is the maximum amount of monetary utilities (payoffs) its members can jointly obtain in a given application environment. Any coalition game in normal form can be equivalently described in an extensive form (sequential coalition game).

outside the considered coalition. Any coalition  $C$  forms through a binding agreement on the distribution of its joint payoff  $v(C)$  among its members, the so-called payoff distribution. A coalition formation environment is superadditive or subadditive, depending on the type of all cooperative games it allows. In subadditive games, at least one pair of potential coalitions is not better off by merging into one, while in superadditive games coalition merging is always beneficial.

The solution of a cooperative game with side-payments is a so-called coalition configuration  $(S, u)$ . It consists of a partition  $S$  of  $A$ , the so-called coalition structure, and an  $n$ -dimensional payoff distribution vector which components are computed by a utility function  $u$ . In other words, each agent  $a \in A$  gets assigned an utility  $u(a)$  (or payoff) out of the value  $v(C)$  of the coalition  $C$  it is member of in a given coalition structure  $S$ . A configuration  $(S, u)$  is stable if no agent has an incentive to leave its coalition in  $S$  due to its assigned individually rational payoff  $u(a) \geq v(a)$  according to an agreed-upon stability concept (also called coalition theory). The negotiation of stable coalitions includes two main activities which are not independent from each other: The building of a coalition structure  $S$ , and the distribution of joint payoffs  $v(C)$  among the members of each coalition  $C$  formed in  $S$ .

### *Classic coalition theories*

Prominent coalition theories (solution spaces for coalition games) are the Shapley-value, the Core, the Bargaining Set, the Nucleolus, and the Kernel. Our own contributions to the field rely on the Shapley-value and the Kernel.

**Core-stable coalitions.** One approach to form stable coalition configurations consists of the following two steps: searching for a coalition structure in a corresponding coalition structure graph for the given game  $(A, v)$  and then computing its payoff according to the stability concept of the Core (Sandholm, 1999). The Core of a coalition game for coalition structure  $S$  is the set of individually rational payoff distributions, so-called not dominated coalitional rational imputations, that maximizes the social-welfare, i.e., the sum of all coalition values of coalitions in  $S$ . Unfortunately, the Core is often empty for coalition games. Besides, searching for an optimal coalition structure  $S$  among the exponential number of  $|A|^{|A|/2}$  possible coalition structures is computationally difficult, because we have to try  $2^{|A|}$  coalition structures.

**Shapley-value-stable coalitions.** A payoff division according to the popular Shapley-value provides an agent with the added value or so-called marginal contribution it brings to the given coalition structure, averaged over all of its possible joining orders - which makes the Shapley-value individual rational and fair for agents to use in superadditive games, but exponentially hard to compute. Algorithms for negotiating coalitions that rely on this stability concept, and a variation of it, the bilateral Shapley-value (Ketchpel,



1994), have been developed and successfully used in cooperative information systems (Klusch, 1998; Klusch & Shehory, 1996)[211, 191], and decentralized power transmission planning (Contreras et al., 2004)[77]. In chapters 10 and 11, we present coalition algorithms based on the bilateral Shapley-value, and its fuzzy variant.

**Kernel-stable coalitions.** The Kernel of a coalition game for a coalition structure  $S$  is the set of payoff distributions  $u$  of so-called Kernel-stable configurations  $(S, u)$  in which all coalitions in  $S$  are in equilibrium. Coalition  $C$  is in such an equilibrium if each pair of agents in  $C$  is in equilibrium, that is, if any pair of agents in  $C$  is balanced so that none of both agents can outweigh the other in  $(S, u)$  by having the option to get a better payoff in an alternative hypothetical coalition. In other words, in Kernel-stable configurations, no agent  $a$  in some coalition  $C \in S$  can object to its assigned payoff by making the claim against any other agent  $a^*$  in  $C$  that  $a$  could obtain more payoff in alternative coalitions (not in  $S$ ) without  $a^*$ , than  $a^*$  without  $a$ . That requires each agent  $a \in C$  to compare its surplus  $s(a, a^*)$  in all alternative coalitions over all agents  $a^* \in C$ , which makes the Kernel exponentially hard to compute unless a constant limits the size of coalitions.

However, the Kernel appears to be attractive for many applications because it is unique for any three-agent game, it assigns symmetric agents of some coalition in a given coalition structure for equal payoff, and it is locally Pareto-optimal. Polynomial-time coalition algorithms for negotiating polynomial Kernel-stable coalition configurations have been developed and applied to the domain of cooperative information systems in (Klusch, 1998; Kraus & Shehory, 1999). In chapters 11 and 12, we present coalition algorithms that base on the Kernel and its fuzzy variant.

**Fuzzy-valued and fuzzy coalition games.** In so-called fuzzy-valued coalition games, agents negotiate fuzzy payoff distributions in order to deal with uncertainties about joint payoffs in coalitions. Respective negotiation protocols additionally provide a defuzzification method for fuzzy payoffs that guarantees stability of the defuzzified payoff assignment. In non-classical so-called fuzzy coalition games, each agent can vary its degree of membership in one or multiple overlapping coalitions.

A fuzzy-valued coalition game consists of a set of agents, a fuzzy characteristic function  $v$ , and the membership function  $m$  of the fuzzy quantities  $v(C)$  that can be interpreted as expectation of the common coalitional profit that is to be distributed among its members (Mares, 2001)[249]. That is, the worth  $v(C)$  of a fuzzy-valued coalition  $C$  is a fuzzy set of its possible real-valued coalitional profits. This set of fuzzy quantity  $v(C)$  has at least one modal value determined by the membership function  $m$ . If, for a given fuzzy cooperative game, the coalition value  $v(C)$  is equal to one modal value of  $C$  for all possible coalitions  $C$ , it is equivalent to a (deterministic) cooperative game.

In chapter 11, we present negotiation protocols that allow agents to solve fuzzy-valued coalition games by configurations that are stable according to the fuzzy bilateral Shapley-value, or the fuzzy Kernel. Chapter 12 presents a coalition algorithm for negotiating risk-bounded, Kernel-stable fuzzy coalitions of service providers.

**Stochastic coalition games.** A cooperative game with stochastic (probabilistic) payoffs is defined by a set of agents, a set of possible actions coalitions might take, and a function that assigns each action of a coalition a real-valued stochastic variable with finite expectation, representing the payoff to a coalition when this particular action would be taken (Suijs, 1999)[350]. In contrast to classical, deterministic coalition games, the payoffs can be random variables, and the actions a coalition can choose from are explicitly modeled, because the payoffs are not uniquely determined.

#### *Dynamic coalition formation*

Frequent changes of tasks, resources, user preferences, or service availability, as well as the set of trading partners affect the respective coalition games. However, the majority of coalition formation protocols for multi-agent systems are static in the sense that the agents cannot react on such changes during the negotiation but have to perform a complete restart of the negotiation. This problem of dynamic coalition forming (DCF) is more general (Klusch & Gerber, 2002) than the equally named problem considered in cooperative game theory.

In fact, the (classical) latter variant of dynamic coalition formation only refers to possible changes of coalition memberships by the agents during negotiation such that the underlying coalition game is not affected. Since the agent society and the coalition values are assumed to be not affected by environmental changes that occur during the actual negotiation, we categorize this classical "dynamic coalition theory" and respective solutions that have been proposed by, for example, Arnold and Schwalbe (2002)[12], and Konishi and Ray (2003)[216] as static.

#### *Types of coalitions among service provider and consumer agents*

We distinguish between three types of coalitions between rational service provider and consumer agents.

- **Service provider coalitions.** In this case, only service providers form coalitions to maximize their individual profits obtained from the joint satisfaction of service requests issued by consumer agents which are exogenous to the respective coalition game. In particular, each provider charges its own local customers (consumer agents) and those of other providers for the

satisfaction of their service requests with a fixed price, and then tries to maximize its individual profits by coalition forming with other providers. The joint value of each provider coalition is the maximum amount of service charges its members can obtain from their consumers for individually and jointly satisfying service requests. In case of fuzzy or probabilistic charges, hence uncertain coalition values such as in reverse auctions<sup>9</sup>, stochastic or fuzzy-valued coalition mechanisms can be used. In non-classical coalition forming, providers can reduce their individual risk of monetary losses by participating in (or distributing its resources over) multiple (fuzzy) coalitions with varying degree of involvement.

- **Service consumer coalitions.** Service consumers may form coalitions to exploit synergies none of them could accomplish alone. The joint value of a consumer coalition is the sum of maximal individual service valuations its members can obtain from their users reduced by the joint payment of fixed service charges by the voted coalition leader to exogeneous service providers. For example, if one concrete service would benefit multiple consumer agents, these agents could coalesce to request and pay for this service only once. The same holds in case the services are offered as a bundle only and each consumer agent is only interested in paying for parts of it.<sup>10</sup>
- **Mixed service provider and consumer coalitions.** In general, cooperative game theory does not distinguish between different types of rational agents such as service consumers or service providers but focuses on how joint values are generated and distributed in stable coalitions. In contrast to pure service provider or consumer coalitions, it is assumed that providers advertise their service execution costs only without pre-defined

<sup>9</sup> For example, in a consumer-initiated reverse auction providers can form coalitions based on their acceptable minimum service charges in order to bid for satisfying requests (and consequently make profits), and the consumer selects the winning coalition as the one with the lowest sum of service charges. The value of a provider coalition is uncertain since it is not known for which charges the coalition can actually win the auction.

<sup>10</sup> For example, a provider offers concrete services A and B as a bundle [A, B] for a fixed price of 10 Euros covering both the cost of executing the bundle and additional charge for profit. Consider two service consumer agents  $a_1, a_2$  each interested in only part of the advertised bundle. Consumer  $a_1$  values service A with 8 Euros but service B with 0 Euros, while consumer  $a_2$  values A with 0 Euros and service B with 7 Euros. None of them is able to pay for the bundle as a whole such that both individually fail in providing their users with the desired service. However, by forming a coalition  $C$  (without the exogeneous provider) with voted coalition leader  $a_1$  responsible to purchase the bundle from the provider (on behalf of  $C$ ), both consumers could obtain the service they are interested with joint profit ( $v(C) = (8 - 10) + (7 - 0)$ ) of 5 Euros. Any agreed-upon solution of this 2-agent coalition game ( $\{a_1, a_2\}, v$ ) with side-payments distributes this joint payoff between the consumers which minimizes their individual service charges respectively.

(fixed) profits; these profits are determined by the negotiated payoff distribution with other provider and consumer agents in respectively mixed coalitions based on an agreed-upon coalition theory.

For example, a mixed coalition of one pure service consumer and multiple pure service providers that can only jointly satisfy the consumer's service requests draws its complete joint value from the utility of the consumer. That is, the consumer agent contributes the maximum valuation of desired services by its user to this coalition, while the providers contribute their service execution costs. If the utility of the consumer is higher than these costs, a stable payoff distribution will specify side-payments from the consumer to the providers such that each coalition member benefits.<sup>11</sup> The same holds in cases where agents are acting both as provider and consumer at the same time.

In the following, we discuss selected negotiation protocols for static and dynamic coalition formation among rational agents. These protocols are, in principle, applicable to each of the above types of service agent coalitions. The majority of approaches in the literature focuses on the negotiation of payoff distributions in service provider coalitions with exogeneous consumer agents and non-negotiable (fixed) service charges.

#### *Examples of static coalition formation protocols*

Kraus et al. (2003)[220] propose and analyse negotiation protocols for heuristic coalition formation among service provider agents that try to satisfy issued service requests. These requests have deadlines and offer a fixed price for their successful satisfaction with time discount (price declines over time). For each request, the first provider coalition that forms for its satisfaction is chosen, such that optimal quality of service is not an issue. Coalitions are formed using a synchronized, turn-based negotiation protocol which is coordinated by a central manager. Two heuristics are proposed for the providers during coalition forming. The marginal heuristic ranks coalitions according to their overall utility, whereas the expert heuristic favors coalitions for an agent if it is an expert in a required task. It was experimentally shown that the marginal heuristic outperforms in settings with complete information, while the

---

<sup>11</sup> In the above example, the provider only states the bundle execution costs (without additional fixed profit), and enters a coalition negotiation with both consumer agents  $a_1, a_2$ . This may eventually yield a grand coalition with individual rational, stable payoff distribution and respective side-payments from the consumer to the provider such that each coalition member benefits: While the provider can cover its bundle execution costs plus some possible extra charge, each consumer is able to obtain its desired service for a possibly lower price than its original maximal service valuation - depending on the negotiated payoff distribution. In other words, the provider's profit is maximal and consumers' charges are minimal in the sense of the agreed-upon coalition theory.

expert heuristic is better for negotiation under incomplete information. Under complete information, near optimal results can be reached.

However, in contrast to traditional game-theoretic coalition formation, the joint profit of each coalition is distributed equally among members, hence raises the question of fairness and individual rationality of the solution. Besides, agents are assumed to report their costs for offered service values (service execution costs) truthfully, although they could (unfairly) increase their individual payoffs by lying.

In a follow-up work, Kraus et al. (2004)[221] investigate different payoff distributions and show that under given time constraints, the willing of agents to compromise up to 20% of their payoff is beneficial, because coalitions can be formed faster in these cases. It is also shown that using the game-theoretic stability concept of the Kernel does not lead to more stable coalitions.

Müller et al. (2006)[268] propose a negotiation protocol for the static formation of complementary-based coalitions (teams) among Web service providers. This is not a game-theoretic coalition formation protocol, since no coalition theory is applied to form stable coalitions. According to the negotiation protocol, service requests are satisfied by respectively formed coalitions of providers. The coalition (team) building process starts with an initially contacted provider that invites other providers that are relevant to realize some part of the given composition plan to contribute to the solution. Coalition leaders are responsible for such invitations, while coalition members perform majority voting on their acceptance. However, the decision making of the agents except the coalition leader, as well as the stability concept, and the service composition remains unclear from the description.

#### *Examples of dynamic coalition formation protocols*

To the best of our knowledge, there are only two solutions to the hard DCF problem, that are the general DCF-S scheme of Klusch and Gerber (2002) (see chapter 13), and the DCF model for resource allocation presented by Soh and Tsatsoulis (2002)[342].

In the DCF-S scheme, each agent simulates, selects, and negotiates coalitions each of which is able to accomplish one of its goals with an acceptable ratio between estimated risk of failure and individual profit. In other words, the agents strive to solve a set of single goal-oriented coalition games  $(A, v)|G$  by forming potentially overlapping coalitions with stable payoff distributions. For this purpose, the game-theoretic stability concepts for strict coalitions either have to be properly adapted such as proposed for Kernel stable fuzzy coalitions (cf. chapter 12), or to be replaced by other notions of stability such as the asymptotic, SIBO or BIBO stability of a distributed system of complementary-based coalitions, that are teams of agents.

Soh and Tsatsoulis (2002)[342] propose a dynamic coalition forming scheme for the task allocation domain[342]. They model coalition formation as an ongoing, continuous process where new tasks can appear, agents can join

coalitions, and existing coalitions can finish their tasks and disband at any time. Agents initiate the coalition formation process for given tasks by looking for promising partners, regarding their potential utility for the coalition. The potential utility is based on their previous behaviour in negotiations as well as their performance of tasks executions, using reinforcement learning. However, the approach does not cope with the problem of payoff distribution at all.

### **Interrelations**

We can distinguish between static and dynamic interleaving of service composition planning and negotiation.

#### *Static interleaving of service composition with negotiation*

Statically interleaved composition and negotiation is sequential, that is, negotiation can take place either after or before planning. A service composition agent can negotiate every single service of its composition plan after its generation, or the set of services available for composition planning in advance. In the first case, negotiation fails if at least one service (or its functional replacement by additionally interleaved service discovery) of the given composition plan cannot be negotiated. Negotiation failures may trigger service composition replanning and require a complete restart of negotiation. In principle, the choice of a static negotiation protocol is independent from the choice of a Semantic Web service composition planner that can be used by trading agents. Approaches for such sequential interleaving of composition and negotiation those in Preist et al. (2003)[300] for service composition planning agents on multiple auctions, and the negotiation protocols in Kraus et al. (2003) and Müller et al. (2006) for static coalition, respectively, team formation among service provider agents for given service composition plans. Our static coalition formation protocols are presented in chapters 10 to 12.

As mentioned above, negotiation could also take place even before composition planning starts. For example, a coalition of agents representing a joint venture of enterprises can negotiate a common set of services as a basis for composition and joint trading. Thus, the search space for any composition planning within the coalition is restricted to these pre-selected services. In such settings, the determination of the joint payoff of the grand coalition, and its stable distribution to all members is of interest only. Negotiation protocols restricted to the grand-coalition are presented, for example, by Goradia and Vidal (2007)[135, 134] and Rahwan et al. (2007)[306, 305].

#### *Dynamic interleaving of service composition with negotiation*

Dynamic interleaving of negotiation with service composition allows agents to negotiate services even during their composition planning such that both activities become mutually dependent. For example, the planning agent may check at each plan step whether services required to reach the given goal

state<sup>12</sup> can be successfully negotiated under given budgetary constraints. If some intermediate negotiation result exceeds the given limit before the goal is actually reached the planning has to be either fully or heuristically restarted. In turn, the selected negotiation protocol should allow for leveled or defeatable precommitments until the planning process is actually completed.

An example of a negotiation protocol supporting two phases of accept and reject is the alternating-offers bargaining protocol by Dang and Huhns (2006) which bases on the two-phase commit protocol from database transaction theory. However, to the best of our knowledge, there is no approach to agent-based service negotiation in the literature that dynamically interleaves with Semantic Web service composition planning.

## The Contributions

The following four chapters present solutions to various open problems of game-theoretic coalition negotiation between rational service agents. We start with an analysis of manipulation-safe and privacy-preserving coalition forming based on selected coalition theories (chapter 10). This is followed by negotiation protocols for non-classical coalition games with fuzzy coalition values (chapter 11). In addition, we propose non-classical coalition negotiation protocols that allow the agents to bound their individual risk of coalition failure by negotiating their involvement in multiple overlapping coalitions to a certain degree (chapter 12). Finally, we present solutions to non-classical dynamic coalition forming in open environments (chapter 13). In such environments, non-deterministically occurring changes to the set of tasks and agent society may affect the running coalition negotiation. These contributions are joint work with colleagues at the University of Southampton, the PhD students Bastian Blankenburg and Andreas Gerber, and master student Wosseok Park of my research team at DFKI.

*Chapter 10: Secure Negotiation of Coalitions.* In this chapter, we first propose a coalition formation protocol, called BSCA-P, that allows rational service agents to keep certain kinds of financial data private during the negotiation of stable coalitions with minimum loss of assigned payoff. In particular, we show that by using this protocol no agent has to reveal its local service sales value and final payoff, and can achieve a certain degree of both agent and service anonymity while still successfully participating in coalition negotiations. Further, we analyse to what extent a special game-theoretic coalition algorithm, called KCA (Klusch, 1998)[191], for forming Kernel stable coalitions among rational agents is manipulation-safe against frauds in face of imperfect information on actual coalition values, and safe against changes of the agent society. We prove a negative result for the latter case: If agents are

---

<sup>12</sup> These services correspond to, for example, helpful actions with minimal goal distance selected by XPlan2 at each plan step (cf. chapter 9).

leaving the actual coalition game  $(A, v)$  then a K-stable solution of the respectively changed game  $(A', v')$  cannot be guaranteed without a total restart of the coalition negotiation using the KCA protocol. Besides, it is also not manipulation-safe against frauds: The agents can neither prevent nor detect a certain type of fraud which would lead to an unjustified increase of individual profits during the negotiation - though this comes at the very expense of exponentially high computation costs for the deceiving agent.

As one positive result for the KCA protocol, we showed that coalition negotiations with the KCA are privacy-preserving in the sense that they allow for non-disclosing self-values without any loss of profits. The proof relies on the fact that the computation of Kernel stable payoff distributions is not dependent on such self-values such that individual agents can hide the respective local financial data from other agents without causing a loss of benefit for them and anyone else involved in the negotiation.

Related work on trusted coalition forming in the national BMB+F project SEMAS at DFKI includes the development and experimental evaluation of the coalition protocol BSCA-TR\*. Using this protocol, agents can negotiate bilateral Shapley-value-stable coalition configurations with trust and reputation. This work is not included in this chapter; for more details on the BSCA-TR\* (and its variants), I refer to the diploma thesis of my diploma student Wooseok Park (Park, 2004)[285].

*Chapter 11: Negotiation of Fuzzy-Valued Coalitions.* One major underlying assumption of classical game-theoretic coalition algorithms is that the given coalition values of the considered game are crisp. This does not necessarily hold in service negotiation settings with potentially uncertain coalition outcomes. In this chapter, we first propose a novel solution to the problem of negotiating stable coalitions with fuzzy-valued coalitions. For this purpose, we combine concepts from fuzzy set theory with the game-theoretic stability concept of the Kernel to deduce the new concept of a so-called fuzzy Kernel, and provide a low-complexity algorithm, called KCA-F, for forming fuzzy Kernel-stable coalitions among rational agents.

We complement this non-classical game-theoretic negotiation protocol with another one, called BSCA-F. This protocol enables agents to negotiate a solution of fuzzy-valued coalition games that is stable according to the so-called fuzzy bilateral Shapley-value based on the fuzzy Shapley-value introduced by Mares (2001)[249]. In particular, we show that utilizing the proposed possibilistic mean value for eventually defuzzifying the negotiated fuzzy agent payoffs is reasonable, and that fuzzy ranking methods can be utilized to implement optimistic, or pessimistic strategies of individually rational agents.

*Chapter 12: Negotiation of Fuzzy Coalitions.* Classical cooperative game theory restricts each rational agent to be member of one coalition only. In this chapter, we show that non-classical game theoretic negotiation of overlapping coalitions with our negotiation protocol, called RFCF, allows agents to make



better use of their resources, hence gain more profits. In particular, agents can control and bound the risk caused by possible failure or default of some potential coalition partner by spreading their involvement in multiple coalitions to a certain degree. In other words, agents can lower their individual risk of monetary losses by participating in a number of appropriate coalitions, if coherent risk measures are considered.

Service provider agents can split their computational resources for service execution to concurrently participate in different coalitions each of which trying to realize one service composition plan. Hence, the degree of membership of providers in a coalition is fuzzy, which results in overlapping (fuzzy) coalitions. If one group of providers decides to execute an additional plan, it simply forms an additional fuzzy coalition. Due to strict deadlines given by consumer agents for the delivery of a complex service in their request, the joint payoff of each coalition depends on the individually estimated runtime of services in a plan that satisfies the request, hence relies on the probability of failure and success of timely plan execution by a potential coalition. We assume each provider agent to assess the other agent's risk of failure in a coalition and consider membership in a coalition as an investment with shared rewards for timely joint service delivery. Consequently, each provider can specify bounds of its individual financial risk based on the financial risk measure TCE (tail conditional expectation) that is coherent for continuous probability (of failure and success) distributions. The RFCF protocol guarantees providers to respect these risk bounds while negotiating Kernel-stable fuzzy coalitions.

*Chapter 13: Dynamic Coalition Forming.* The coalition negotiation protocols presented in previous chapters are not safe against dynamic changes of the environment (e.g., goals, tasks, agent society) that affect the actual coalition game to solve, and require a total restart of negotiations in such a case. This chapter provides an adaptive solution to this problem of dynamic coalition forming by means of an opportunistic and high-risk coalition formation scheme, called DCF-S. It allows agents to respond to changes in their set of goals and the agent society even during their coalition negotiation.

Basic idea of the DCF-S scheme is that each agent concurrently simulates, selects, and negotiates coalitions each of which able to accomplish one of its goals with an acceptable ratio between estimated risk of failure and individual profit. In case of changes that affect the bilateral negotiations for one of its coalitions to realize, it restarts the simulation of potential alternatives for this particular coalition keeping those agents with which it has already successfully reached an agreement (opportunistic) and updates local knowledge about the environment. This way, the agent gradually learns how to select best coalitions avoiding a full restart with implied penalty payments for removing agents from other valid coalitions and respective decrease of its reputation for other agents. Using the DCF-S scheme agents continuously approximate the best solutions to their coalition games based on local knowledge about the dynamic environment. Instances of this scheme, so-called DCF-S coalition

algorithms have been successfully demonstrated for dynamic joint resource re-planning among farmers for cereal harvesting. For more details on these algorithms and their experimental evaluation, I refer to the PhD thesis of Andreas Gerber (Gerber, 2004)[126].

### **Open Problems**

Some major research challenges of agent-based semantic service negotiation are the following.

- Theory and practice of dynamic and efficient interleaving of game-theoretic negotiation models with semantic service composition (re-)planning and discovery.
- In-depth analysis of the impact of logic based, or hybrid forms of semantic annotation of negotiated services on both the efficiency of the negotiation process, and the quality of possible service level agreements for given negotiation mechanisms.
- Development of negotiation protocols for dynamic, privacy preserving and trusted coalition forming in open environments with uncertainties on the coalitional outcomes. Each of the contributions presented in the following chapters provide a separate solution to each part of this challenge. Though, research in this direction has started, and can build on the vast literature on trust, and trusted negotiation; for accessible reviews of the field, we refer to, for example, (Artz & Gil, 2007)[15], (Ramchurn & Jennings, 2004), [308], and (Sabater & Sierra, 2005)[321].

---

## Secure Negotiation of Coalitions

B. Blankenburg and M. Klusch: BSCA-P: Privacy Preserving Coalition Forming Among Rational Web Service Agents. *Kuenstliche Intelligenz*, 1/06, pages 19 - 25, BoettcherIT Verlag, February 2006.

B. Blankenburg and M. Klusch: On Safe Kernel Stable Coalition Forming Among Agents. *Proceedings of the 3rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, New York, USA, pages 580 - 587, ACM Press, 2004.

# BSCA-P: Privacy Preserving Coalition Forming Among Rational Web Service Agents

Bastian Blankenburg, Matthias Klusch

In this paper, we propose a coalition formation protocol, called BSCA-P, that allows intelligent agents to negotiate game-theoretically stable coalitions for Web service trading with a maximum of individual monetary profit, while keeping certain kinds of financial data private. We show that no agent has to reveal its local service sales value and final payoff, and can achieve a certain degree of both agent and service anonymity while still successfully participating in rational coalitions.

## 1 Introduction

In today's increasingly networked and competitive world, the appropriate utilization of pay per use Web services are considered as one major key to the success of commercial service oriented business applications in domains such as e-logistics, tourism, and entertainment. In the near future, intelligent service agents are not only supposed to search for, interact with, and compose, but also negotiate access to, and execute such Web services on behalf of its user, or other agents. In fact, they may exhibit some form of economically rational cooperation by forming coalitions to share the created joint monetary value while at the same time maximizing their own individual payoff. According to classical microeconomics, means and concepts of cooperative game theory are inherently well suited to this purpose.

However, the public revelation of quantity and value of local service sales, and individual requests for particular services required to play cooperative games with complete knowledge could lead to an unsolicited competitive advantage in web service oriented business. The problem is, how can certain kinds of local financial data be kept private while still successfully participating in coalition negotiations to maximize individual profits? Research on privacy preserving coalition formation is in its infancies; first solutions to this problem have been presented in [1, 2]<sup>1</sup>.

The remainder of this paper is organized as follows. In section 2, we introduce basic notions of service agents, coalition theory, and negotiation used throughout this paper. Section 3 provides an analysis and examples of how certain types of financial information can be kept private during negotiation of coalitions, whereas in section 4, we prove that at the communication level, service requests can only be anonymized by means of an anonymous routing protocol. Finally, the overall coalition formation protocol BSCA-P is then presented with its computational and communication complexity in section 5. We conclude in section 6.

## 2 Coalitions of Service Agents

In this section, we introduce the basic notions of Web service agents and cooperative game theory that are required to understand the approach proposed in subsequent sections.

<sup>1</sup>This paper is an extended version of [2].

### 2.1 Service Agents

We consider a Web service to be any kind of task-oriented, XML-based business application software that is location transparent, i.e., network accessible from anywhere with one or multiple protocols of the IP suite, possibly enlarged with additional descriptive metadata to describe its semantics for service consumers, programmable via an API, and loosely coupled with other software applications to implement processes within, or across enterprises. It is supposed to be registered and located by means of web service registries, or intelligent middle agents [5]. Examples of ontology languages for describing Web services range from WSDL for the contemporary Web, to WSDL-S, OWL-S, and WSMO for the future semantic Web.

Unfortunately, in recent literature, the terms agent and Web service are often used interchangeably. An autonomous service agent is a special kind of intelligent information agent [4] that is supposed to pro-actively search for, interact with, and compose, but also negotiate access to, and execute atomic, or composed Web services on behalf of its user, or other agents. In contrast, Web services are considered passive in that they are not expected to be able to, for example, autonomously decide upon its invocation, or intelligently (re-)plan the composition of its own or other services either individually, or in joint cooperation with other services.

There are, in principle, three different ways of how an individual service consumer or provider agent can interact with a network accessible Web service, that is via (1) the service interface, or communication with another service agent that either (2) provides this and possibly multiple other services, or even (3) temporarily integrates parts of the service code into its own on demand, thereby changing the individual reactive agent behaviour accordingly. In this paper, we adopt the second perspective of interaction, and do not differentiate between the offering of atomic, or composite web services  $WS$  by service agents.

However, we do assume that each agent  $a$  is equipped with an individual model of monetary valuation  $w_a(WS)$  of each local, or remote service  $WS$  it can deliver to its users. Besides, the local execution of its own services does produce a certain amount of costs  $c_a(WS)$  per invocation. Any pair of service agents  $a, a'$  is interested to access or execute, respectively, a particular service  $WS$  provided by  $a'$  only if it is possibly profitable to do so, i.e.,  $w_a(WS) > c_{a'}(WS)$ . Since we further assume an individual service agent to act economically rational, it will try to negotiate a profitable joint agreement for cooper-

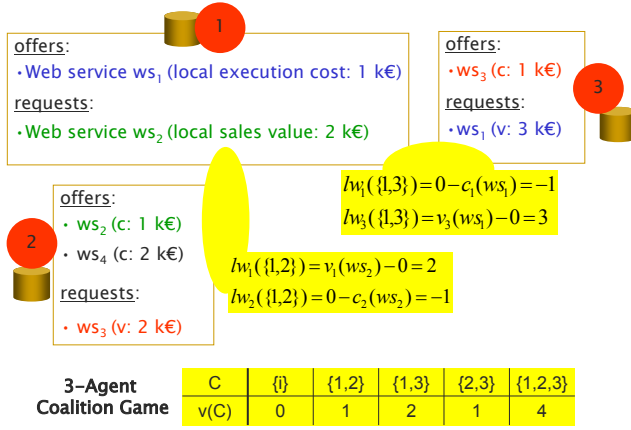


Figure 1: Example coalition game for three web service agents.

ation with other service agents in a coalition to maximize its individual payoff. Such an agreement includes the commitment of each coalition member to deliver both relevant local services and those it is planning to compose jointly with other members, as well as the implementation of the negotiated payoff distribution among them. Such kind of rational cooperation between Web service agents can be described in terms of cooperative, or coalition games.

## 2.2 Coalition Games

According to microeconomics, a *coalition game*  $(\mathcal{A}, v)$  is constituted by a given set  $\mathcal{A}$  of service agents, and the value  $v$  of every possible joint coalition  $C \subseteq \mathcal{A}$  among them. Each coalition value  $v(C)$

$$v(C) := \sum_{a \in C} lw_a(C) \quad (1)$$

is the maximum monetary gain that can be achieved by cooperation between the members of coalition  $C$ . This gain is defined by the sum of the so called *local worth*  $lw_a(C)$  of each agent  $a \in \mathcal{A}$  in  $C$  as its member.

Let  $E_a(C)$  denote the set of services that are executed by  $a \in \mathcal{A}$ , and  $R_a(C)$  the set of services of members of  $C$  which are accessed by  $a$ . The local worth of  $a$  in  $C$

$$lw_a(C) := \sum_{WS \in R_a(C)} w_a(WS) - \sum_{WS \in E_a(C)} c_a(WS) \quad (2)$$

is its total monetary contribution to  $C$  (without sidepayments), that is the difference between the local income of the service agent by charging its users for relevant data produced by local, or remote services offered by another coalition member, and the cost of executing its local services as requested.

**Example 1** Consider a 3-agent coalition game as shown in figure 1. Service agent  $a_1$ , for example, offers its own web service  $ws_1$  to any other known agent of the game, that are service agents  $a_2$  and  $a_3$ . Each local execution of its service would cost  $a_1$  an amount of 1ke, but produces no monetary income as it is of no relevance for its own users. Hence, its self value is zero.

Agent  $a_3$  is requesting access to service  $ws_1$  from  $a_1$ , as it can charge its local users with an total amount of 3ke per

use, but does not offer any service of interest for users of  $a_1$  in turn. As a consequence, the local worth of  $a_1$  in a joint coalition with  $a_3$  is  $lw_{a_1}(C_1) = -c_{a_1}(ws_1) = -1$  whereas that of  $a_3$  is  $lw_{a_3}(C_1) = -c_{a_3}(ws_1) = 3$ . Summing up the local worths of all agents in every possible coalition yields the set of coalition values which is the cooperative game to solve by negotiation: What coalitions shall the agents form, and how then to distribute the coalition values to their members?

## 2.3 Stable coalitions

A solution  $(S, u)$  of a cooperative game  $(\mathcal{A}, v)$  is a partition  $S$  of the set of agents, means a set of disjunct coalitions that have been formed together with a distribution  $u$  of their coalition values to each agent as member of respective coalitions. This payoff distribution is assumed to be efficient, that is the joint benefit is distributed completely without any loss, and individual rational, such that no agent gets less than it could obtain by staying alone.

As soon as the coalitions have been formed, the computed payoff distribution will be implemented by means of certain sidepayments that are to be exchanged among the agents. In our case, each service agent  $a$  as a member of a certain coalition  $C$  may only claim for some sidepayment  $sp_u(a, C)$  by other agents, if the difference

$$sp_u(a, C) := u(a) - lw_a(C); \quad (3)$$

$$sp_u(C^*, C) := \sum_{a \in C^*} sp_u(a, C), C^* \subseteq C \quad (4)$$

between its assigned payoff  $u(a)$ , that is the money it shall get, and its local worth  $lw_a(C)$  in  $C$ , that is its local income based on charging its own users, is positive. Otherwise, it has to make a sidepayment of an amount of  $|sp_u(a, C)|$  to other agents in  $C$ . If the payoffs  $u$  are distributed without loss, the same holds for its implementation by exchange of sidepayments between members of a coalition.

**Corollary 1** Let  $C \in S$ ,  $(S, u)$  be a solution of a game  $(\mathcal{A}, v)$ . If  $C^* = C$ , we write  $sp_u(C)$ . Then  $sp_u(C) = 0$ , if and only if  $u$  is efficient wrt.  $S$ .

A solution is called *stable*, in case no agent could have an incentive to leave its coalition due to its assigned payoff. There exist different stability concepts in game theory from which we adopted, for the work reported in this paper, an efficient variant of the Shapley value [7], the so called *bilateral Shapley value*.

**Definition 1** The union  $C$  of two disjoint coalitions  $C_1, C_2 \subset \mathcal{A} \setminus \emptyset$  is called a bilateral coalition, with  $C_1$  and  $C_2$  called founders of  $C$ . A bilateral coalition  $C$  is called recursively bilateral iff it is the root node of a binary tree denoted by  $T_C$  for which (a) every non-leaf node is a bilateral coalition, and its founders and sub-coalitions are its children, and (b) every leaf is a single agent coalition. For the depth  $d(C^*, T_C)$  of a node  $C^*$  in  $T_C$  with either  $C^* = C$ , or  $C^* \subset C^{**}$ ,  $C^{**} \in T_C$  it holds that

$$d(C^*, T_C) = \begin{cases} d(C^*, T_C) = 0 & \text{if } C^* = C \\ d(C^*, T_C) = d(C^{**}, T_C) + 1 & \text{otherwise} \end{cases}$$

A coalition structure  $S$  for  $(\mathcal{A}, v)$  is called (recursively) bilateral if  $\forall C \in S : C$  is (recursively) bilateral, or  $C = a$ ,  $a \in \mathcal{A}$ .

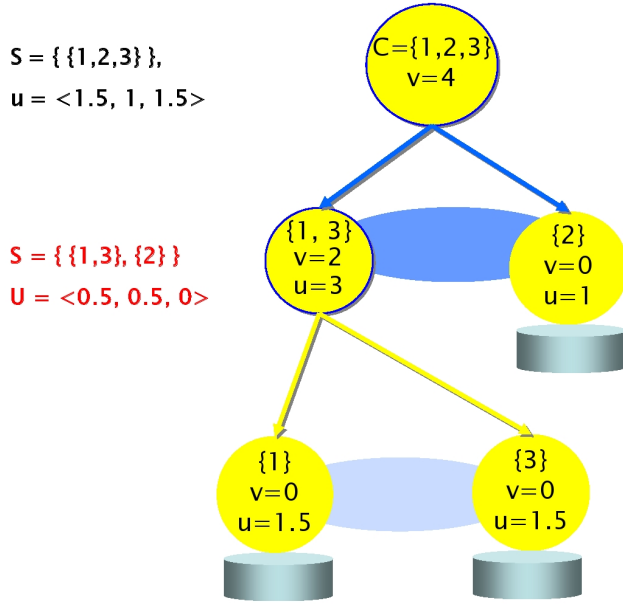


Figure 2: Binary tree of bilateral coalitions for the example game.

The bilateral Shapley value  $\sigma_b(C, C_i, v), C_i, i \in \{1, 2\}$  of the bilateral coalition  $C$  is defined as the Shapley value of  $C_i$  in the game  $(\{C_1, C_2\}, v)$ :

$$\sigma_b(C_i, C, v) = \frac{1}{2}v(C_i) + \frac{1}{2}(v(C) - v(C_k)) \quad (5)$$

with  $k \in \{1, 2\}, k \neq i$ .

Given a recursively bilateral coalition structure  $S$  for a game  $(\mathcal{A}, v)$ , a payoff distribution  $u$  is called recursively bilateral Shapley value stable iff for each  $C \in S$ , every non-leaf node  $C^*$  in  $T_C : u(C_i^*) = \sigma_b(C_i^*, C^*, v_{C^*}), i \in 1, 2$  with  $\forall C^{**} \subseteq \mathcal{A}$ :

$$v_{C^*}(C^{**}) = \begin{cases} \sigma_b(C_k^p, C^p, v_{C^p}) & \text{if } C^p \in T_C, \\ & C^* = C^{**} = C_k^p, k \in 1, 2 \\ v(C^{**}) & \text{otherwise} \end{cases} \quad (6)$$

In other words, when merging two recursively bilateral coalitions into one its value will be distributed down the corresponding coalition tree to its members by means of recursively replacing the coalition value of the respective parent coalition with its payoff, that is the bilateral Shapley value.

**Example 2** Consider our example game, and the bilateral coalition  $C_1 = \{a_1\} \cup \{a_3\}$ . Since  $v(\{a_1\}) = v(\{a_3\}) = 0$ , it holds that  $\sigma_b(\{a_1\}, \{a_1\} \cup \{a_3\}, v) = \sigma_b(\{a_1\}, \{a_1\} \cup \{a_3\}, v) = 0 + \frac{1}{2}(2 - 0) = 1$ . Merging of  $C_1$  with  $C_2 = \{a_2\}$  ( $C = C_1 \cup C_2$ ) yields  $v(C) = 4$  and  $v(C_2) = 0$ , thus  $\sigma_b(C_1, C, v) = 2 + \frac{1}{2}(4 - 2) = 3$  and  $\sigma_b(C_2, C, v) = 0 + \frac{1}{2}(4 - 2) = 1$ . Recursively replacing the coalition value  $v(C_i)$  in (5) with the bilateral Shapley value of  $C_i$  then leads to the following payoff distribution (cf. figure 2):  $u(a_1) = \sigma_b(\{a_1\}, \{a_1\} \cup \{a_3\}, v^*) = 0 + \frac{1}{2}(3 - 0) = 1.5$  and  $u(a_3) = \sigma_b(\{a_3\}, \{a_1\} \cup \{a_3\}, v^*) = 0 + \frac{1}{2}(3 - 0) = 1.5$ .

## 2.4 Negotiation of stable coalitions

The BSCA protocol for negotiating such stable coalitions does restrict negotiation to pairs of voted leaders of coalitions of given

maximum size, thereby reducing the communication complexity. Each coalition leader recursively distributes the potential joint coalition value to those agents that are members of its current coalition according to the bilateral Shapley values (cf. figure 2). Coalitions are formed bilaterally per round based on coalition proposals that are mutually accepted based on the expected maximum of individually rational payoffs for the agents involved. However, to determine these potential payoffs, the BSCA protocol requires each agent to reveal its local worth to every potential coalition partner per round.

From the knowledge about the local worth of an agent in some coalition, one could easily deduce, for example, its monetary self value, that is the local income of the agent from selling its own services exclusively to its own users. Further, from the distribution of service requests, and the known set of local worth values, any third party could easily deduce that some agent does apparently have a stronger interest in certain services offered by some agents than by others. These kinds of revelation could lead to an unsolicited competitive advantage of these parties in web service oriented business after, or in parallel to playing this particular coalition game.

In general, that is the problem of how to preserve data privacy in cooperative games playing: To what extent an individual service agent could keep its self values, and expected final payoffs private to other agents such that all agents are negotiating a solution of still the same game that is stable according to the bilateral Shapley value? More general, what is the trade off for any service agent between hiding certain kinds of private financial data from potential collaboration partners, and collectively rational profit making?

## 3 Non-Disclosure of Financial Data

The basic idea to solve this problem is that each agent should not disclose its total local worth in a potential joint coalition to any other agents but the amount resulting from collaboration only. This so called *additional local worth* is the difference between its local worth in the merger  $C$  and its current coalition. In fact, any coalition (leader)  $C_1$  can locally compute its bilateral Shapley value  $u_C(C_1) = v(C_1) + \frac{1}{2}av(C_1, C_2)$  in a joint coalition  $C$  with some other coalition  $C_2$  simply by means of its self value, and an equal distribution of the *additional joint coalition value*  $av(C_1, C_2)$ . The latter value is computed by summing up the additional local worths of the agents in each of the bilateral coalition founders. As a consequence, coalition  $C_1$  could compute its expected payoff without knowing anything about the total local worth of its potential coalition partner  $C_2$ .

In more detail, (5) can be rewritten as

$$\sigma_b(C_i, C, v) = v(C_i) + \frac{1}{2} \cdot (v(C) - v(C_1) - v(C_2)) \quad (7)$$

with  $i \in \{1, 2\}$ . Thus, the *additional coalition value*

$$av(C_1, C_2) := v(C_1 \cup C_2) - v(C_1) - v(C_2) \quad (8)$$

produced by forming coalition  $C_1 \cup C_2$  is evenly distributed among  $C_1$  and  $C_2$ . For recursively bilateral Shapley value stable payoff distributions, this means that each child node in the coalition tree gets half of the additional payoff of its parent node.

The share of the total payoff that a node gets is thus directly dependent on its depth in the tree, which is shown by the following lemma.

**Lemma 1** Let  $(S_1, u_1)$  and  $(S_2, u_2)$  configurations for a game  $(A, v)$ , with  $u_1$  and  $u_2$  being recursively bilateral Shapley value stable, and  $\exists C_1, C_2 \in S_1 : C = C_1 \cup C_2 \in S_2$ . Then

$$\forall C^* \in T_C : u_2(C^*) = u_1(C^*) + \frac{av(C_1, C_2)}{2^{d(C^*, T_C)}}$$

*Proof:* Induction over  $d(C^*, T_C)$ . The case  $d(C^*, T_C) = 0$  is obvious because of the efficiency of  $\sigma_b$  and definition of  $av$ . For  $d(C^*, T_C) = 1$ , we have  $C^* = C_i$ ,  $i \in \{1, 2\}$  and  $u_2(C_i) = \sigma_b(C_i, C, v) = v(C_i) + \frac{1}{2}av(C)$ . Again because of the efficiency of  $\sigma_b$ ,  $v(C_i) = u_1(C_i)$ , and thus  $v(C_i) + \frac{1}{2}av(C) = u_1(C_i) + \frac{av(C)}{2^{d(C^*, T_C)}}$ . In case  $d(C^*, T_C) = k > 1$  and lemma 1 holds for all  $C^{**}$  with  $d(C^{**}, T_C) < k$ , we have  $C^* = C_i^p$ ,  $i \in \{1, 2\}$ ,  $C^p \in T_C$ ,  $d(C_i^p, T_C) = d(C^p, T_C) + 1$  and  $u_2(C_i^p) = \sigma_b(C_i^p, C^p, v_{C_i^p})$  with  $v_{C_i^p}(C^p) = u_2(C^p) = u_1(C^p) + \frac{av(C)}{2^{d(C^p, T_C)}}$ . Applying 6 and 7, we get

$$\begin{aligned} u_2(C_i^p) &= v(C_i^p) + \frac{1}{2}(u_2(C^p) - v(C_i^p) - v(C_k^p)) \\ &= v(C_i^p) + \frac{1}{2}(u_1(C^p) + \frac{av(C)}{2^{d(C^p, T_C)}} - v(C_i^p) - v(C_k^p)) \\ &= v(C_i^p) + \frac{1}{2}(u_1(C^p) - v(C_i^p) - v(C_k^p)) \\ &\quad + \frac{av(C)}{2^{d(C^p, T_C)+1}} \\ &= u_1(C_i^p) + \frac{av(C)}{2^{d(C_i^p, T_C)}} \end{aligned}$$

For the merge of  $C_1$  and  $C_2$  to form  $C = C_1 \cup C_2$ , we further define the *additional local worth* of agent  $a \in C_i$ ,  $i \in \{1, 2\}$ :

$$alw_a(C_i, C) := lw_a(C) - lw_a(C_i), \quad (9)$$

and the summarized additional local worth for a subcoalition  $C^* \in T_{C_i}$

$$alw(C^*, C_i, C) := \sum_{a \in C^*} alw_a(C_i, C) \quad (10)$$

Also, note that

$$\begin{aligned} av(C_1, C_2) &= \sum_{a \in C} lw_a(C) - \sum_{a \in C_1} lw_a(C_1) - \sum_{a \in C_2} lw_a(C_2) \\ &= alw(C_1, C_1, C) + alw(C_2, C_2, C) \end{aligned} \quad (11)$$

The following theorem shows that in order to compute its sidepayment when merging coalitions  $C_1$  and  $C_2$ , each subcoalition  $C^* \in T_{C_i}$  only needs to consider its sidepayment for the case without the merge and the additional local worths of  $C_1$ ,  $C_2$  and  $C^*$ :

**Theorem 1** Let  $(S_1, u_1)$  and  $(S_2, u_2)$  configurations for a game  $(A, v)$ , with  $u_1$  and  $u_2$  being recursively bilateral Shapley value stable, and  $\exists C_1, C_2 \in S_1 : C = C_1 \cup C_2 \in S_2$ . Then  $\forall C^* \in T_{C_i}$ ,  $i \in \{1, 2\}$ :

$$\begin{aligned} sp_{u_2}(C^*, C) &= sp_{u_1}(C^*, C_i) - alw(C^*, C_i, C) \\ &\quad + \frac{alw(C_1, C_1, C) + alw(C_2, C_2, C)}{2^{d(C^*, T_C)}} \end{aligned}$$

Negotiation round 1:

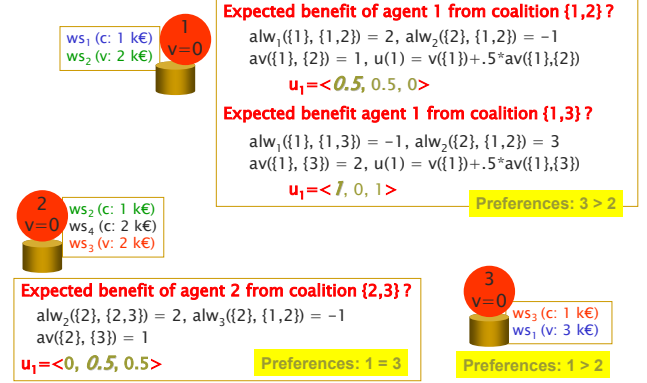


Figure 3: Privacy preserving negotiation of coalitions (round 1).

*Proof:* Remember that for any  $u$ ,  $sp_u(C^*, C) = \sum_{a \in C^*} u(a) - lw_a(C) = u(C^*) - \sum_{a \in C^*} lw_a(C)$  (see 4). Because of lemma 1, 9, 10 and 11, we can rewrite

$$\begin{aligned} sp_{u_2}(C^*, C) &= u_1(C^*) - \sum_{a \in C^*} lw_a(C) + \frac{av(C_1, C_2)}{2^{d(C^*, T_C)}} \\ &= u_1(C^*) - \sum_{a \in C^*} (lw_a(C_i) + alw_a(C_i, C)) \\ &\quad + \frac{av(C_1, C_2)}{2^{d(C^*, T_C)}} \\ &= sp_{u_1}(C^*, C_i) - alw(C^*, C_i, C) + \frac{av(C_1, C_2)}{2^{d(C^*, T_C)}} \\ &= sp_{u_1}(C^*, C_i) - alw(C^*, C_i, C) \\ &\quad + \frac{alw(C_1, C_1, C) + alw(C_2, C_2, C)}{2^{d(C^*, T_C)}} \end{aligned}$$

Please note that in case of  $C^* = C_i$ , it holds that  $sp_{u_1}(C^*, C_i) = 0$ , because of  $C_i \in S_1$  and corollary 1. Hence, in order to obtain recursively bilateral Shapley value stable payoff distributions by repeatedly merging coalitions, all subcoalitions have to inform each other only about their additional local worths. Absolute local worths as well as coalition values do not have to be revealed at all. This is in contrast to the traditional way of negotiating stable coalitions with complete prior knowledge about local worth and coalition values that constitute the game to be solved. We acknowledge that this does hold in particular for the bilateral Shapley value but not necessarily for other game-theoretic stability concepts.

**Example 3** Consider, again, our example coalition game (cf. fig. 1). During the first negotiation round, it turns out that agents  $a_1$  and  $a_3$  would prefer each other as a coalition partner, since both of them could obtain a higher individually rational payoff in a joint coalition than each could get in a separate coalition with agent  $a_2$  (cf. figure 3). Agent  $a_2$  is even indifferent in respect to the coalition it would prefer.

More concrete,  $\{a_1\}$  and  $\{a_3\}$  form a coalition  $C_1$ , with  $alw_{a_1}(\{a_1\}, C_1) = -1 - 0 = -1$  and  $alw_{a_3}(\{a_3\}, C_1) = 3 - 0 = 3$ . According to theorem 1 we get

$$sp_u(\{a_1\}) = 0 + \frac{(-1) + 3}{2^1} - (-1) = 2$$





Checking of desired service and agent anonymity in  $C = C_1 \cup \{2\}$  before proposal submission:

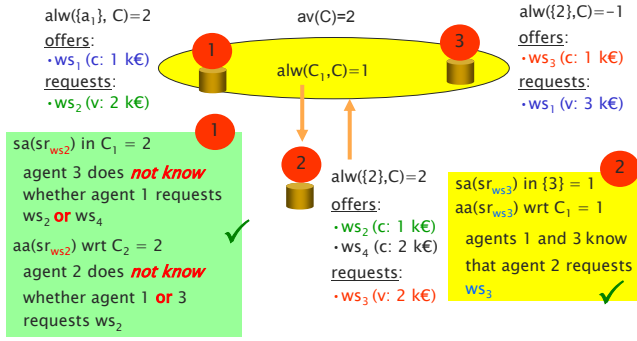


Figure 5: Individual service request anonymities.

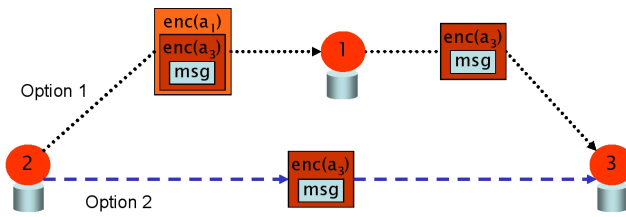


Figure 6: Options of encrypted service request message "onion" routing from agent  $a_2$  to agent  $a_3$ .

provider agent in  $C_2$ , with  $|C_2| \geq 2$  knows that its service has been requested by an agent in  $C_1$  but not which one (cf. figure 5).

We can quantify these kinds of possibilistic anonymity for each service  $WS$  requested by an individual agent  $a \in C_1$  in terms of

- *service anonymity*  $sa(WS, C_1) = |\bigcup_{a \in C_2} OS_a|$  within  $C_1$  in terms of the number of services offered by members of  $C_2$ , such that, in the extreme, no agent knows which of its coalition partners does access what specific service, and
- *agent anonymity*  $aa(WS, C_2) = |C_1|$  with respect to  $C_2$  in terms of the size of its actual coalition  $C_1$ , since from the perspective of agents in  $C_2$ , any agent in  $C_1$  might be the originator of the service request.

Assuming that each agent specifies its desired (default) minimum degrees of service and agent anonymity for each web service  $WS$  it is interested in, any request and coalition proposal to potential cooperation partners will be submitted, i.e.,  $WS \in R_a(C)$ , if and only if these requirements are met.

To maintain the above mentioned types of anonymity also at the communication level, we adopt the simple onion routing protocol [8] to anonymize the exchange of service request messages between the service agents. In essence, each service request message gets routed between sender and receiver via randomly selected intermediate agents each of which encrypting the message with its individual public key (cf. figure 6). This way, for communication paths consisting of at least three agents, no intermediate agent is able to determine both the origin and the receiver of a service request message nor to decrypt its content to some extent as guaranteed by the underlying encryption

protocol.

## 5 Coalition Formation Protocol BSCA-P

In this section, we finally propose the coalition formation protocol BSCA-P that makes use of all concepts and means that have been introduced in the previous sections. We assume that service offers along with service execution costs are known in prior.

**Algorithm 1** For a game  $(A, v)$ ,  $S_0 := \{\{a\} | a \in \mathcal{A}\}$ ,  $r := 0$  and  $\forall C \in S_0 : sp_0(C) := 0$ . In every coalition  $C \in S_r$ , every agent  $a \in C$  performs:

1. Let  $C \in S_r$ ,  $a \in C$  and  $S^* := S \setminus C$ .
2. Communication:
  - (a) For all  $C^* \in S^*$  do:
    - i. Determine set  $R_a(C^*)$  of requests, subject to the sets  $OS_{a^*}$  of offers for each  $a^* \in C^*$ , costs and minimum anonymity degrees.
    - ii. For each service request which is in  $R_a(C) \cap R_a(C^*)$  keep the one with minimum costs.
    - iii. Set  $alws_a(C^*) := alw_a(C, C^*)$ .
    - iv. For each bilateral coalition  $C^a$ ,  $C^a \in T_C$ ,  $a \in C^a$ ,  $a = Rep(C^a)$ , wait for a message from  $Rep(C_i^a)$ ,  $i \in 1, 2, a \notin C_i^a$  containing  $alws_{Rep(C)}(C^*)$  and set  $alws_a(C^*) := alws_a(C^*) + alws_{Rep(C)}(C^*)$ .
    - v. If  $a = Rep(C)$  then send  $alws_a(C^*)$  to  $Rep(C^*)$ ; else send  $alws_a(C^*)$  to  $Rep(C^+)$  with  $C^+ \in T_C$ ,  $a = Rep(C_i^+)$ ,  $i \in 1, 2, a \neq Rep(C^+)$ .
  - (b) If  $a = Rep(C)$  then receive  $alws_{Rep(C^*)}(C)$  and set  $alws(C^*) := alws_{Rep(C^*)}(C) + alws_a(C^*)$  for all  $C^* \in S^*$ ; else go to step 3i.
3. Coalition Proposals:
  - (a) Set  $Candidates := S^*$ ,  $New := \emptyset$  and  $Obs := \emptyset$
  - (b) Determine a coalition  $C^+ \in Candidates$  with  $\forall C^* \in Candidates : alws_a(C^+) \geq alws_a(C^*)$ .
  - (c) Send a proposal to  $Rep(C^+)$  to form coalition  $C \cup C^+$ .
  - (d) Receive all coalition proposals from other agents.
  - (e) If no proposal from  $Rep(C^+)$  is received and  $Candidates \neq \emptyset$ , set  $Candidates := Candidates \setminus \{C^+\}$  and go to step 3b.
  - (f) If a proposal from  $Rep(C^+)$  is received, then form the coalition  $C \cup C^+$ :
    - i. If  $o(Rep(C)) < o(Rep(C^+))$  then set  $Rep(C \cup C^+) := Rep(C)$ ; else set  $Rep(C \cup C^+) := Rep(C^+)$ .
    - ii. Inform all other  $Rep(C^*)$ ,  $C^* \in S^* \setminus C^+$  and all  $a^* \in C$ ,  $a^* \neq a$  about the new coalition and  $Rep(C \cup C^+)$
    - iii.  $New := \{C \cup C^+\}$ ,  $Obs := \{C, C^+\}$
  - (g) Receive all messages about new coalitions. For each new coalition  $C_1 \cup C_2$  and  $Rep_{C_1 \cup C_2}$ , set  $Candidates := Candidates \setminus \{C_1, C_2\}$ ,  $New := New \cup \{C_1 \cup C_2\}$  and  $Obs := Obs \cup \{C_1, C_2\}$ .

- (h) Send the sets *New* and *Obs* to all other coalition members  $a^* \in C$ ,  $a^* \neq a$
- (i) If  $a \neq \text{Rep}(C)$  then receive the sets *New* and *Obs* from  $\text{Rep}(C)$ .
- (j) Set  $r := r + 1$ ,  $S_r := (S_{r-1} \setminus \text{Obs}) \cup \text{New}$ .
- (k) For each (sub-)coalition  $C^* \in T_C$  with  $\text{Rep}(C^*) = a$ , determine  $sp_r(C^*)$  according to theorem 1 (using  $sp_{r-1}(C^*)$  instead of  $sp_u(C^*)$ ).
- (l) If  $C_r = C_{r-1}$  then stop; else go to step 2

**Theorem 2** Let  $n = |A|$  and  $m := \max_{a \in A} \{|R_a|\}$ . The computational complexity of the protocol BSCA-P is in  $O(n^3 m^2)$ . The communication complexity in terms of the number of exchanged messages per agent is in  $O(n^2)$ .

*Proof:* cf. [2]

After stable coalition configurations have been negotiated among service agents following the BSCA-P protocol, it will be implemented by exchange of actual sidepayments. For this purpose, each leader of a (sub-)coalition  $C$  makes or receives payments  $sp$  to, or from other leaders of immediate parent and child coalitions in the binary coalition tree. This way, only leaders of 2-agent coalitions get informed about individual sidepayments, that are its own, and that of the other agent. As a consequence, only the very first coalition partner of an individual agent  $a$ , that is its direct neighbour leaf in the coalition tree, might ever know  $a$ 's exact sidepayment, though its individual utility value still remains private. To ensure anonymous service requests and access, we require each agent to follow the simple onion routing protocol.

## 6 Conclusions

We proposed a protocol for privacy preserving and stable coalition formation among rational web service agents. In particular, the payoffs and utilities of these agents can almost or even completely be kept private, respectively, during bilateral negotiations of recursively bilateral Shapley value stable coalitions. Further, following the BSCA-P protocol, there is no need to reveal absolute coalition values to successfully participate in coalition negotiations at all. However, we showed that the existence of service requests might not be hidden in general but anonymized to a specified degree. In summary, the negotiation protocol BSCA-P allows service agents in the Internet to keep personal financial data private, while increasing their individual profits by means of rational cooperation with others in coalitions.

**Acknowledgement:** This work has been supported by the German ministry for education and research (BMBF) under project grant SCALLOPS-01-IW-D02.

## References

- [1] B. Blankenburg and M. Klusch. On safe kernel stable coalition forming among agents. In *Proc. 3<sup>rd</sup> Int. Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004)*, New York, USA, 2004.

- [2] B. Blankenburg and M. Klusch. BSCA-p: Privacy-preserving coalition formation. In F. Kluegl et al., editor, *Proc. 3<sup>rd</sup> German Conference on Multi-Agent System Technologies (MATES)*, Koblenz, Germany. Springer, LNAI, 3550, 2005.
- [3] Joseph Halpern and Kevin O'Neill. Anonymity and information hiding in multiagent systems. *Journal of Computer Security*, Special Edition on CSFW 16:75–88, 2003.
- [4] M. Klusch. Information agent technology for the internet: A survey. *Data and Knowledge Engineering*, 36(3), 1980.
- [5] M. Klusch and K. Sycara. Brokering and matchmaking for coordination of agent societies: A survey. In A. Omicini et al., editor, *Coordination of Internet Agents*. Springer, 2001.
- [6] A. Pfitzmann and M. Köhntopp. Anonymity, unobservability and pseudonymity: a proposal for terminology. In *International Workshop on Designing Privacy Enhancing Technologies*, pages 1–9, New York, 2001. Springer-Verlag.
- [7] L. S. Shapley. A value for n-person games. In H. W. Kuhn and A. W. Tucker, editors, *Contributions to the Theory of Games II*, volume 28 of *Annals of Mathematics Studies*, pages 307–317. Princeton University Press, Princeton, 1953.
- [8] P F Syverson, D M Goldschlag, and M G Reed. Anonymous connections and onion routing. In *IEEE Symposium on Security and Privacy*, pages 44–54, Oakland, California, 4–7 1997.

## Kontakt

Bastian Blankenburg  
DFKI GmbH  
Stuhlsatzenhausweg 3, 66123 Saarbrücken  
Tel.: +49 (0)681 302-64823 Email: blankenb@dfki.de  
Dr. Matthias Klusch  
DFKI GmbH  
Stuhlsatzenhausweg 3, 66123 Saarbrücken  
Tel.: +49 (0)681 302-5297 Email: klusch@dfki.de

Bild

**Bastian Blankenburg** is a PhD student of the university of the Saarland and researcher of the multi-agent systems group of the German research center for Artificial Intelligence (DFKI). His research interests include cooperation, risk, trust and privacy in multi-agent systems. He received a diploma from the University of the Saarland, Germany.

Bild

**Matthias Klusch** is a research fellow and senior researcher of the German research center for Artificial Intelligence (DFKI) where he is leading a research team on intelligent information agents and systems. His research interests include agent-based and service-oriented computing, intelligent information search and management, innovative applications of AI, rational cooperation and decision-making, and the semantic Web. He received a PhD in computer science from the University of Kiel, Germany.

# On Safe Kernel Stable Coalition Forming among Agents

Bastian Blankenburg, Matthias Klusch  
German Research Center for Artificial Intelligence,  
Deduction and Multiagent Systems,  
Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany  
e-mail: {blankenb, klusch}@dfki.de

## Abstract

*We investigate and discuss safety and privacy preserving properties of a game-theoretic based coalition algorithm KCA for forming kernel stable coalitions among information agents in face of imperfect information on actual coalition values, and changing agent society. In addition, we analyze the chances of deceiving information agents to succeed in coalition negotiations using the KCA protocol. We show that a certain type of fraud which leads to an increase of individual profit can neither be prevented nor detected, but this comes at the expense of exponentially high computation costs for the deceiving agent.*

## 1. Introduction

Game-theoretic coalition algorithms can be used by intelligent agents as coordination means in a variety of applications in different environments. Applications in the health care and m-commerce domain are required to preserve the privacy of user information to a large extent. In this respect, one interesting question concerns the relation between the safety properties of a given coalition negotiation protocol, and the privacy of information required to specify the underlying coalition game to be solved by the agents according to the protocol. In particular, what is the minimum amount of information required for a given coalition algorithm to output stable solutions? Will it be individually beneficial to deceive negotiation partners on local information that is used to determine the value of joint coalitions? What kinds of information can be hidden by an agent from selected agents at what costs in terms of its bargaining position in the coalition negotiation?

As to our knowledge, there is no work on this topic available yet. Hence, in this paper, we provide some first thoughts on, and preliminary answers to these questions, taking a special coalition algorithm KCA [4] for kernel stable coalition forming as an example.

In section 2, we introduce the reader to the basics of co-operative game theory, with focus on Kernel stable coalition forming, to an extent that is necessary to understand the results presented. Readers who are familiar with the field can skip this section. In section 3, we analyze and discuss some safety properties of the KCA protocol for negotiation settings with imperfect information on coalition values, and changes in the set of negotiating agents. Finally, the chances of deceiving agents to succeed in coalition negotiations according to the KCA protocol are discussed in section 4.

## 2. Kernel stable coalition forming

In this section, we briefly introduce the reader to the basic concepts of coalition games, kernel stable coalitions, and a specific negotiation protocol KCA [4] that can be used by agents to form such coalitions. For a more comprehensive introduction to co-operative game theory we refer the reader to, for example, [2, 5].

### 2.1. Basics

A co-operative or *coalition* game  $(A, v)$  is defined by a set  $A$  of agents wherein each subset of  $A$  is called a *coalition*, and a real-valued characteristic or *coalition value* function  $v : A \rightarrow \mathbb{R}$  that assigns each coalition  $C \subseteq A$  its maximum monetary gain. Any set of coalition values for all possible coalitions defines a coalition game. The *self-value*  $v(\{a_k\}) \equiv v(a_k)$  of an agent  $a_k$  denotes the maximum profit it may gain without any cooperation with other agents. It is assumed that each value  $v(C)$  does not depend on the actions of agents outside  $C$ , any coalition  $C$  forms by a binding agreement on the distribution of  $v(C)$  among its members, and no side-payments are allowed from  $C$  to any agents outside  $C$  within the given game.

The sum of both self-value and marginal contribution to a coalition  $C$  is called the *local value* or *worth*  $lworth_a(C)$  of agent  $a$  for  $C$ . An individual *agent production utility* function  $U_k$  determines the worth of task-related produc-

tions of agent  $a_k$ . Agents coalesce to increase their individual profits that may result from jointly accomplishing their tasks. The value  $lworth_a(C)$  of agent  $a$  for coalition  $C$  is the total revenue  $a$  may obtain for accomplishing its tasks in  $C$  on behalf of its user or other agents in  $C$ . Each coalition value is the sum of the local values of its members ( $v(C) = \sum_{a \in C} lworth_a(C)$ ).

*Stable Solutions of Coalition Games.* The solution of a cooperative game with side payments is a *coalition configuration*  $(S, u)$  which consists of a partition  $S$  of  $A$ , the *coalition structure*, and a  $n$ -dimensional, real-valued *payoff distribution* vector which components are computed by a real-valued *payoff or utility function*  $u$ . The payoff distribution assigns each agent in  $A$  its utility  $u(a)$  out of the value  $v(C)$  of coalition  $C$  it is member of in a given coalition structure  $S$ .

In *individually rational* payoff distributions each agent gets at least its self-value  $u(a) \geq v(\{a\})$ . For *group rational* distributions, it holds that the group of all agents maximises its joint payoff. In *Pareto-optimal* payoff distributions no agent is better off in any other valid payoff distribution for the given game and coalition structure.

A configuration  $(S, u)$  is called *stable* if no agent has an incentive to leave its coalition in  $S$  due to its assigned payoff  $u(a)$ . Different characteristics and criterions of stability define different solution spaces for a co-operative game.

In general, non-super-additive games at least one pair of potential coalitions is not better off by merging into one. The meaning of stability of coalitions depends on the considered discipline and application domain. Many if not most of the coalition formation algorithms today rely on chosen game-theoretic concepts for stable pay-off division within coalitions according to, for example, the Shapley-value, the Core, the Bargaining Set, or the Kernel [2]. In this paper, we focus on the latter concept of coalition stability.

*Kernel Stable Configurations.* The *kernel* of a co-operative game  $(A, v)$  with respect to a given coalition structure  $S$  is a set of *K-stable configurations*  $(S, u)$  wherein each coalition in  $S$  is in equilibrium. Each pair of agents  $a_k, a_l$  in  $C$  is in equilibrium, if they cannot outweigh each other in  $(S, u)$  by having the option to get a better payoff in coalition(s) *not* in  $S$  excluding the opponent agent. The *surplus* of agent  $a_k \in A$  with respect to the opponent  $a_l$  in a given configuration  $(S, u)$  is  $s_{kl} = \max_{R \notin S, a_k \in R, a_l \notin R} \{e(R, u)\}$ , where  $e(R, u) = v(R) - u(R)$  denotes the *excess* of alternative coalitions  $R$ . Agent  $a_k$  outweighs  $a_l$ , if  $s_{kl} > s_{lk}$  and  $u(a_l) > v(a_l)$ . Any pair of agents  $a_k, a_l$  is in equilibrium with respect to  $(S, u)$ , if one of the following constraints is satisfied: ( $s_{kl} = s_{lk}$ ), or ( $s_{kl} = s_{lk}$  and  $u_l = v(a_l)$ ), or  $s_{kl} < s_{lk}$  and  $u_k = v(a_k)$ .

To compute a K-stable payoff distribution, agents transfer side payments among each other; the demand of agent  $a_k$  from  $a_l$  is defined as  $d_{kl} = \min\{\frac{s_{kl}-s_{lk}}{2}, u(a_l) -$

$v(a_l)\} \geq \alpha$ , and zero else, as an upper limit of any side-payment  $\alpha$  to be added (subtracted) from the payoff  $u(a_k)$  ( $u(a_l)$ ). The transfer scheme converges against a K-stable  $(S, u)$  after  $O(n \log(re/\epsilon))$  iterations with  $O(n2^n)$  steps each, where  $re(u)$  denotes the relative error.

The kernel of a game is exponentially hard to compute unless, for example, the size of the coalition is limited by a constant. The kernel appears to be attractive, since it is unique for any 3-agent game  $(A, v)$ , assigns symmetric agents of some coalition in a given coalition structure for  $(A, v)$  equal payoff, and is locally Pareto-optimal in the set  $K$ . Polynomial coalition algorithms for polynomial K-stable coalition configurations have been developed for cooperative information agents with perfect [4] or imperfect knowledge [1].

## 2.2. KCA coalition algorithm

Any set of rational agents can negotiate kernel stable solutions  $(S, u)$  of co-operative games  $(A, v)$  by using the so called KCA coalition algorithm [4] which proceeds as follows.

Each agent  $a$  performs

### 1. Communication

- (a) Set  $(\{a_1\}, \dots, \{a_n\}, (v(\{a_1\}), \dots, v(\{a_n\})))$  to be the current configuration.
- (b) Set each agent to be the coalition leader of its coalition.
- (c) Generate a totally ordered list of all agents sorted by their overall computational power. *The sorting of this list is the same for all agents. It is not important here how this is exactly done.*
- (d) Send the set of tasks  $T_a$  to all other agents.
- (e) Receive the set of tasks of each other agent.
- (f) Evaluate the set of tasks accomplishable by  $a$  and send it to all other agents.
- (g) For each other agent, receive the set of accomplishable tasks.
- (h) For every coalition  $C \subseteq A$ , evaluate  $lworth_a(C)$  and send it to all other agents.
- (i) Receive all local values from each other agent.

### 2. Generating Proposals

- (a) If  $a$  is not coalition leader of  $C$ ,  $a \in C$ , go to 4e.
- (b) For each other coalition  $C^* \in S$ ,  $a \notin C^*$  compute a Kernel-stable configuration  $(S^*, u^*)$  with  $C \cup C^* \in S^*$  and all other coalitions unchanged. If  $u^*$  strictly dominates  $u$ , send  $(S^*, u^*)$  as a configuration proposal to the leader of coalition  $C^*$ .

### 3. Evaluating Proposals

- (a) Receive configuration proposals from the other coalition leaders.
- (b) Evaluate the received proposals. Choose one proposal  $(S^+, u^+)$  that is most beneficial to accept, i.e. for which  $u^+$  strictly dominates  $u$  and is not strictly dominated by  $u^*$  of any other received proposal  $(S^*, u^*)$ .
- (c) Inform all other coalition leaders about the accepted proposal.

#### 4. Deciding Upon Coalition Configuration

- (a) Receive all accepted proposals from the other coalition leaders.
- (b) If no proposal was accepted, stop.
- (c) Choose one proposal to become the new configuration. To do this, determine an order of preference of the proposals according to the following keys, priority in descending order:
  - i. Bilaterally accepted proposals are preferred to unilaterally accepted ones. An accepted proposal  $(S^*, u^*)$  of coalition  $C^*$  for coalition  $C^+$  is bilaterally accepted iff  $C^+$  accepted a proposal of  $C^*$ .
  - ii. If any two proposals  $(S^*, u^*)$  and  $(S^+, u^+)$ ,  $\sum_{a^* \in A} u^*(a^*) > \sum_{a^+ \in A} u^+(a^+)$  holds,  $(S^*, u^*)$  is preferred to  $(S^+, u^+)$ .
  - iii. If any two proposals are equally preferable according to the above properties, the one which was made by the agent with the greater computational power is preferred.
- (d) Inform all other coalition members in  $C$  about the new configuration.
- (e) If  $a$  is not coalition leader, receive the new configuration.
- (f) If  $a$  is in the coalition, determine the new coalition leader as the agent in the new coalition with the greatest computational power.
- (g) If  $a$  is coalition leader, do: if  $a$  is in the new coalition, inform all other coalition leaders about  $a$  being the new coalition leader. If  $a$  is not in the new coalition, receive the new coalition leader of the new coalition.
- (h) If the grand coalition was formed or a previously defined time for the coalition formation process is exceeded, stop.
  - (i) Go to 2.

It is assumed that the time for message exchange is limited, and inter-agent communication is correct. Please note that according to the KCA protocol, in each round at most

one new coalition is formed (as a merger of two coalition of the previous configuration), and the computation of a kernel stable payoff distribution with respect to a proposed coalition may affect also the payoffs of those agents that are not involved in that proposal.

## 3. Safety properties of the KCA

### 3.1. Safe KCA with incomplete information

*Unknown Tasks.* Suppose agent  $a_1$  does not receive the complete set of tasks from agent  $a_2$ . The local value  $lworth_{a_2}(C)$  of  $a_2$  for any coalition  $C$  in which both agents are involved depends on the extent  $a_1$  is able to help  $a_2$  in accomplishing its tasks. Since  $v(C) = \sum_{a \in C} lworth_a(C)$  holds, and the task sets of agents are mutually exchanged before the coalition negotiation starts, any partial knowledge on tasks to accomplish only changes the original game, thus does not affect the correctness of the output of the KCA protocol.

*Unknown Local and Coalition Values.* In cases where local values are not known to some agent, it can estimate, or set the corresponding coalition by default. However, this leads to situations in which different agents try to solve different games at the same time with respectively different outcomes of the coalition negotiations according to the KCA protocol.

#### Example 3.1: KCA negotiation with estimated coalition value

Consider a non-superadditive 3-agent game for a set of agents  $\{a_1, a_2, a_3\}$  and coalition values  $v(\{a_i\}) = 0, i \in \{1, 2, 3\}$ ,  $v(\{a_1, a_2\}) = 3$ ,  $v(\{a_1, a_3\}) = 1$ ,  $v(\{a_2, a_3\}) = 3$ , and  $v(\{a_1, a_2, a_3\}) = 0$ . The agents may solve this game using the KCA. For example, in the first round, the mutually exchanged kernel stable proposals are  $Cfg_1 = (\{\{a_1, a_2\}, \{a_3\}\}, (0.5, 2.5, 0))$ ,  $Cfg_2 = (\{\{a_1, a_3\}, \{a_2\}\}, (0.5, 0, 0.5))$  and  $Cfg_3 = (\{\{a_1\}, \{a_2, a_3\}\}, (0, 2.5, 0.5))$ . Suppose that no proposal is bilaterally but unilaterally accepted. As a result, both most beneficial coalitions  $\{a_1, a_2\}$  or  $\{a_2, a_3\}$  could form. In order to obtain a unique configuration, the proposal of the most powerful agent, here  $a_2$ , is selected, that is  $Cfg_1$ .

Suppose that  $a_1$  did not receive  $lworth_{a_3}(\{a_2, a_3\})$  and *under-estimates* the related coalition value  $v^*(\{a_2, a_3\}) = 2$ .  $a_1$  proposes  $(\{\{a_1, a_2\}, \{a_3\}\}, (1, 2, 0))$  to  $a_2$  instead of  $Cfg_1$ . Unfortunately, this proposal is not as good as the one in the original game for  $a_2$  such that it rather accepts the alternative proposal of  $a_3$ . Thus, the kernel stable configuration  $Cfg_3$  is formed. Thus, in contrast to the above solu-

tion of the original game,  $a_1$  stays alone with no profit from joint collaboration.

In case  $a_1$  *over-estimates* the coalition value  $v^*({a_2, a_3}) = 3.5$ , it proposes  $Cfg_{1c} = (\{\{a_1, a_2\}, \{a_3\}\}, (0.25, 2.75, 0))$  instead of  $Cfg_1$  to  $a_2$ , and  $Cfg_{2c} = (\{\{a_1, a_3\}, \{a_2\}\}, (0.25, 0, 0.75))$  instead of  $Cfg_2$  to  $a_3$ . As a result,  $a_2$  and  $a_3$  accept  $Cfg_{1c}$ , respectively,  $Cfg_{2c}$ , and  $a_1$  accepts either  $Cfg_1$  from  $a_2$  or  $Cfg_2$  from  $a_3$ . Given that  $a_1$  and  $a_2$  accepts  $Cfg_1$ , respectively,  $Cfg_{1c}$ , all agents uniquely decide on the configuration  $Cfg_{1c}$  proposed by  $a_1$  based on given criteria for drawing a tie. However, this solution is not kernel stable with respect to the original game but the different game considered by  $a_1$ . Even worse,  $a_1$  obtains less profit than it would be possible in most kernel stable solution of the original game.

◦

If an agent decides to compute kernel stable configurations in case of complete information on coalition values only, the interesting question is how its proposals and behavior in negotiations are affected. In Ex. 3.1,  $a_1$  would not make any proposal to any agent, thus non-kernel stable proposals do not appear in the negotiation.

In general, if  $a$  does not know the coalition value  $v(C^*)$  of a certain coalition  $C^*$ , let  $s_{ik}^*$  denote the estimated surplus of  $a_i$  over  $a_k$ , that is the maximum of the set of excesses of coalitions  $C \neq C^*$  with  $a_i \in C, a_k \notin C$ . This surplus estimation is computable by  $a$  without knowing  $v(C^*)$ . Hence, for each kernel stable configuration  $(S, u)$  which can exactly be computed by  $a$  by using  $s_{ik}^*$  instead of  $s_{ik}$ , it holds that (a)  $C^* \notin S$ , and (b)  $\forall C \in S, a_i, a_k \in C, a_i \in C^*, a_k \notin C^* : u(a_k) = v(a_k)$ . The second constraint means that  $a$  does not need to compute any surplus with  $v(C^*)$ . Agent  $a$  can determine whether a configuration it computes with incomplete information is kernel stable by checking if both constraints are satisfied.

**Example 3.2:** *Computable configurations without certain local values*

Consider the coalition game in example 3.1. Suppose that  $a_1$  does not know  $v(\{a_2, a_3\})$  and decides to neither estimate, nor use any default value for it. As a result,  $a_1$  is not able to compute a kernel stable configuration for the coalition structure  $\{\{a_1, a_2\}, \{a_3\}\}$ , because the excess of coalition  $C^* = \{a_2, a_3\}$  is not computable by  $a_1$ . The reason is that if  $a_1$  (wrongly) assumes  $s_{21} = v(\{a_2\}) - u(a_2)$ , then  $u = (2, 1, 0)$  is kernel stable with respect to the game it considers. However, the second constraint is not satisfied, since  $u(a_1) > v(\{a_1\})$  holds. Thus  $a$  cannot be sure that the assumption is right and the resulting configuration is kernel stable. In fact,  $s_{21}$  is given

by  $v(\{a_2, a_3\}) - u(a_2) - u(a_3) > v(\{a_2\}) - u(a_2)$  since  $v(\{a_2, a_3\}) > v(\{a_2\})$  and  $u(a_3) = 0$ . However, if the game includes  $v(\{a_2\}) = 3$ , then  $u = (0, 3, 0)$  is determinable by  $a_1$  to be Kernel-stable because it is sufficient to know that  $s_{21} \geq v(\{a_2\}) - u(a_2)$  holds. Because  $u(a_1) = v(\{a_1\})$  in this case, it does not matter if  $s_{21} > v(\{a_2\}) - u(a_2)$  or  $s_{21} = v(\{a_2\}) - u(a_2)$  is true and  $v(\{a_2\}) - u(a_2) \geq s_{12}$  holds.

◦

To summarize, using the KCA, coalition negotiations are safe with respect to unknown coalition values  $v(C)$ . Any under- or over-estimation of  $v(C)$  by agent  $a$  yields a changed game  $(A, v)$  for which K-stable solutions can be computed by  $a$  following the KCA protocol but with possibly lower payoff  $u(a)$  compared to solutions that are computed at the same time by other agents for the original game  $(A, v)$ .

### 3.2. Safe KCA with changing agent sets

New agents trying to enter negotiations after these actually started can easily be avoided by verifying the sender of each message to be an expected one. But it may happen that an agent for whatever reason, intended or unintended, becomes unavailable before negotiations end and/or the actions as determined by the coalitional contract are carried out. That is, it does not send any more messages required by the protocol. For example, its network connection may break down. The consequences are different for coalitions leaders and members.

*Coalition leaders dropping out of negotiation.* If the coalitions leader  $a$  of a coalition  $C$  becomes unavailable, consequently no proposals are made or accepted for  $C$ . Other members of  $C$  receive no more configuration updates and eventually time out, finishing the negotiation process. After this, the members of  $C$  have an outdated configuration if the other coalitions made further merges, which means that their payoffs might no longer be Kernel-stable. This is true even if  $a$  becomes available again to accomplish his part of the coalitional contract. If  $a$  stays unavailable, the remaining members of  $C$  are in fact forming the coalition  $C \setminus \{a\}$ . If there exist agents in  $C$  which rely on payments by  $a$ , individual rationality might not be guaranteed any more. Alternatively,  $a$  might become available again before negotiations are finished. This might, depending on the state of  $a$  then, lead to different configurations considered true by different agents.

**Example 3.3:** *Coalition leader leaving and re-entering negotiation*

Consider a 3-agent game with  $A = \{a_1, a_2, a_3\}$  in

which usually  $a_1$  and  $a_2$  bilaterally accept proposals in the first round because  $\{a_1, a_2\}$  is much more beneficial than any other coalition. But  $a_1$  becomes unavailable right after the initial exchange of local values, and thus does not make a proposal for  $a_2$ , nor accepts  $a_2$ 's proposal. Thus,  $\{a_2, a_3\}$  is formed. Suppose  $a_2$  becomes coalition leader. In the second round,  $a_1$  becomes available again just when proposals are to be made.  $a_2$  proposes the formation of the grand coalition while  $a_1$ , not having received a configuration update, proposes the formation of  $\{a_1, a_2\}$ . Because this is the most beneficial proposal, it is chosen as the new configuration, thus splitting up the coalition  $\{a_2, a_3\}$  again (which is not allowed by the KCA). Suppose  $a_1$  becomes coalition leader of  $\{a_1, a_2\}$ .  $a_1$  thus sends a configuration update to  $a_3$ , but  $a_3$  is waiting for a configuration update by its (now former) leader  $a_2$ .  $a_3$  will eventually time out and finish negotiations still 'believing' it would coalesce with  $a_2$ .

◦

*Other coalition members dropping out of negotiation.* The dropping out of coalition members does not influence the coalition formation process unless this happens during task execution which leads to the same problem as for coalition leaders. That is, the configuration may no longer be kernel stable, since the original game no longer models the situation appropriately. As in the case of unavailable coalition leaders, individual rationality of payoffs cannot be guaranteed for agents that rely on payments from unavailable coalition members.

To summarize, using the KCA, coalition negotiations are not safe in case of changing agent society: If agents are leaving the actual coalition game  $(A, v)$  a K-stable solution of the changed game  $(A, v)$  cannot be guaranteed without total restart of the KCA.

## 4. Secure and safe KCA

In this section, we show that, using the KCA, agents can safely negotiate K-stable coalitions and preserve individual data privacy (security) at the same time. In particular, any agent that is involved in the negotiations can completely hide its local data and information used to compute its self-value from other agents. Surprisingly, it can do so without even risking any loss of profit in the final coalition configuration.

### 4.1. Privacy preserving K-stable coalition negotiations

This property of local data privacy preservation in coalition negotiations using the KCA is an inherent property of

the definition of kernel stability, which is stated in the following lemma.

**Lemma 1.** *Let  $(A, v)$  and  $(A, v^*)$  with*

$$\exists a^* \in A, r \in \mathbb{R} : v^*(C) := \begin{cases} v(C) + r & \text{for } a^* \in C \\ v(C) & \text{otherwise} \end{cases}$$

*Then it holds that the configuration  $(S, u^*)$  with  $u^*(a^*) = u(a^*) + r$  and  $\forall a \in A, a \neq a^* : u^*(a) = u(a)$  is K-stable with respect to the game  $(A, v^*)$  iff  $(S, u)$  is K-stable with respect to the game  $(A, v)$ .*

*Proof.* Let  $s_{a^*, a^\circ}^*(C)$  the surplus of agent  $a^*$  over agent  $a^\circ$ ,  $a^*, a^\circ \in C \in S$  in configuration  $(S, u^*)$ . Then it holds

$$\begin{aligned} s_{a^*, a^\circ}^*(C) &= \max_{C^+ : a^* \in C^+, a^\circ \notin C^+} \{v^*(C^+) - \sum_{a \in C^+} u^*(a)\} \\ &= \max_{C^+ : a^* \in C^+, a^\circ \notin C^+} \{v(C^+) + r - \sum_{a \in C^+} u(a) + r\} \\ &= \max_{C^+ : a^* \in C^+, a^\circ \notin C^+} \{v(C^+) + r - \sum_{a \in C^+} u(a) - r\} \\ &= s_{a^*, a^\circ}^*(C) \text{ (in configuration } (S, u). \end{aligned}$$

□

According to lemma 1, it holds for any K-stable solution  $(S, u)$  of coalition game  $(A, v)$  that the configuration  $(S, u^*)$  for the changed game  $(A, v^*)$  with  $v^*(C) = v(C) - \text{lworth}(a, \{a\})$ ,  $v^*(\{a\}) = 0$ ,  $a \in C \in S$ , and  $u^*(a) = u(a) - \text{lworth}(a, \{a\})$  is K-stable. Since  $u(a) - u^*(a)$  is constant for all pairs of K-stable proposals  $(S, u)$  and  $(S, u^*)$ , the KCA negotiation protocol is safe against non-disclosure of self-values.

Since  $v(C) = \sum_{a \in C} \text{lworth}_a(C)$  holds, any agent  $a$  may add  $r \in \mathbb{R}$  to each of its local values, thereby constantly changing its actual contribution to each  $C$  by  $r$ , such that  $a$ 's net result remains the same as in the original game with  $v(C)$ . In particular, if for each agent  $a_i$  the factor  $r_i = -\text{lworth}_{a_i}(\{a_i\})$  is added to every coalition value  $v(C)$  with  $a_i \in C$ , then the resulting game contains only zero-valued self-values and is equivalent to the original game  $(A, v)$ .

To understand why that is the case, consider an agent which communicates its worth  $\text{lworth}^*(a, C) = \text{lworth}(a, C) - \text{lworth}(a, \{a\})$  to every coalition  $C$  in the coalition structure  $S$  of configuration  $(S, u)$  for given game  $(A, v)$ . That action changes the original coalition game to a new game  $(A, v^*)$  with  $v^*(C) = v(C) - \text{lworth}(a, \{a\})$  and  $v^*(\{a\}) = 0$ . However, this change does not affect the value of the excess of any coalition  $C$ , since it holds that  $e^*(C) := v^*(C) - u^*(C) = v(C) - u(C) = e(C)$ . This, in turn, implies that the surplus values of agents in a solution  $(S, u)$  of  $(A, v)$ , and  $(S, u^*)$  of  $(A, v^*)$  remain the same. By induction over all agents  $a \in A$  and  $S$ , it can easily be shown that this finally yields equivalent games for any set of agents modulo their self-values.

As a result, self-values are not required to be communicated among the agents at all in order to solve any given coalition game with a K-stable configuration. That means that, using the KCA, any agent  $a_i$  can hide any set of local information from other agents in K-stable coalition negotiations, without loss of benefit for anyone, iff this local information is exclusively used to compute its self-value  $v(\{a_i\})$ . However, the extent to which local information can be hidden depends on the structure of the coalition game, as we will show by means of a simple application example in the following section.

## 4.2. Application to retailer coalition games

In general, rational agents in e-markets are envisioned to be capable of forming different kinds of coalitions for different purposes. For example, retailer agent coalitions are commonly formed to maximize individual benefits of joint sales to customers. On the other side, customer agent coalitions can be built to maximize individual benefits of joint purchases at retailer site, or to maximize individual brokerage/commission from the customer agents' users.

In the following, we consider a given set  $A$  of retailer agents that form coalitions to improve and share their joint benefits of selling requested items to customer agents. In terms of coalition game theory, the coalition value  $v(C)$  of a retailer agent coalition  $C \subseteq A$  is the maximum joint benefit of retailer agents in  $C$  for selling relevant items to their customer agents. Individual item utility  $U(a, p)$  for retailer agent  $a$  of selling item  $p$  to its customers. The self-value  $v(\{a\})$  of retailer agent  $a$  is the maximum gain of sales without any cooperation. Finally, the retailer agent coalition game  $(A, v)$  is the set of all coalition values.

### Example 4.1: Car Retailer Coalition Game

Consider the following (superadditive) coalition game  $(A, v)$  of three car retailer agents  $\{a_1, a_2, a_3\}$  and following coalition values defined by maximum car sales in each coalition:  $v(\{a_1\}) = 2$ ,  $v(\{a_2\}) = 1.5$ ,  $v(\{a_3\}) = 1$ ,  $v(\{a_1, a_2\}) = 6$ ,  $v(\{a_1, a_3\}) = 8$ ,  $v(\{a_2, a_3\}) = 7$ ,  $v(\{a_1, a_2, a_3\}) = 15$ .

Using the KCA, after first negotiation round, the agents reach the following K-stable solution of the game:  $(S, u) = (\{\{a_1, a_2\}, \{a_3\}\}, (3.5, 2.5, 1))$  which balances the agents' surpluses  $s(a_1, a_2) = v(\{a_1, a_3\}) - (u(a_1) + u(a_3)) = 8 - 3.5 - 1 = 3.5$ , and  $s(a_2, a_1) = 7 - 2.5 - 1 = 3.5$ . After a second (final) negotiation round, the grand coalition is formed with the following K-stable payoff distribution among the retailer agents:  $(S, u) = (\{\{a_1, a_2, a_3\}\}, (5, 4.25, 5.75))$  which balances the agents' surpluses  $s(a_1, a_2) = e(\{a_1, a_3\}) = -2.75$ ,  $s(a_2, a_1) = e(\{a_2\}) = -2.75$ ,  $s(a_1, a_3) = e(\{a_1, a_2\})$

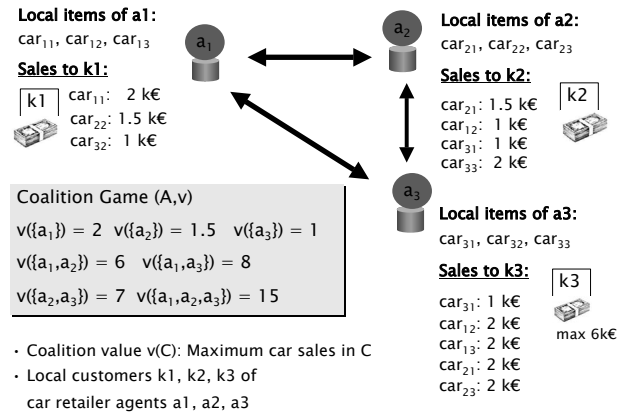


Figure 1. Example of a car retailer agent coalition game

$$= -3, s(a_3, a_1) = e(\{a_2, a_3\}) = -3, s(a_2, a_3) = e(\{a_2\}) = -2.75, s(a_3, a_2) = e(\{a_1, a_3\}) = -2.75.$$

o

### Example 4.2: Privacy Preservation in K-Stable Coalition Negotiations

The K-stable solution  $(\{\{a_1, a_2\}, \{a_3\}\}, (3.5, 2.5, 1))$  of game  $(A, v)$  after first negotiation round bases on the agents' surpluses  $s(a_1, a_2) = v(\{a_1, a_3\}) - (u(a_1) + u(a_3)) = 8 - 3.5 - 1 = 3.5$ , and  $s(a_2, a_1) = 7 - 2.5 - 1 = 3.5$ . Hiding of the self-value by each agent induces a new 3-agent game  $(A, v^*)$  with a new, unique and K-stable solution  $(\{\{a_1, a_2\}, \{a_3\}\}, (1.5, 1, 0))$  which balances the newly formed but equally valued surpluses  $s^*(a_1, a_2) = (v(\{a_1, a_3\}) - v(\{a_1\}) - v(\{a_3\})) - ((u(a_1) - v(\{a_1\})) + (u(a_3) - v(\{a_3\}))) = 5 - 1.5 = 3.5$  and  $s^*(a_2, a_1) = 4.5 - 1 = 3.5$ . This game is equivalent to the original one modulo the agents' self values.

Hiding of self-values implies that, for example, car retailer agent  $a_1$  can prevent the other car retailer agents  $a_2, a_3$  from knowing how many and what kind of own cars it can sell to its local customer for what price. In this example,  $a_1$  can only sell one car of type car11 to its local customer for a price of 2k euros. It can protect this local information without loss of benefit in implied coalition negotiation by simply communicating a zero-valued self-value  $v^*(\{a_1\}) = 0$ . In this example, agents  $a_2$  and  $a_3$  by no means are able to infer the true self-value of  $a_1$  from their local knowledge, since its car11 cannot be alternatively sold to them to maximize any of their joint coalition values. Both



agents do not even know about the existence of car11.

That is not true in cases where local items can be both locally and remotely sold. In such cases the local sales value can be partially inferred by other agents. For example, consider the situation in which car retailer agent  $a_2$  communicates a zero-valued self-value  $v^*({a_2})$  to protect its local information that it can sell car21 to its customer for 1.5k euros. Further, it does not tell  $a_3$  that car21 exists. However, in order to determine the maximum joint coalition value with  $a_3$  it has to communicate to  $a_3$  that  $a_3$ 's car33 can be sold to its local customer  $k_2$  for 2k euros, that is more than  $a_3$  could obtain locally. As a consequence,  $a_2$  communicates its worth  $lworth^*(a_2, \{a_2, a_3\}) = 0.5 (= 2 - 1.5)$  to  $a_3$  such that both agents are now able to compute the maximum sales value of a joint coalition. But now  $a_3$  can easily infer that the self-value  $v({a_2})$  of  $a_2$  must at least be of value 1.5k  $(= 2 - lworth^*(a_2, \{a_2, a_3\}))$  euros, which does not match with the zero value that has been communicated by  $a_2$  to  $a_3$  before.

◦

## 5. Fraud in Kernel stable coalition forming

Serious threats to safe kernel stable coalition forming based on the KCA protocol are caused by, for example, fraudulent agents that intend to (a) unreasonably strengthen their bargaining position in the coalition negotiation, or (b) influence the building of coalitions for any other strategic reason. We show that this is in principle possible, but at high computational costs in practice.

Suppose that agent  $a$  wants to demand more payoff in a given coalition game  $(A, v)$  than originally possible. For this purpose, in the first negotiation round, it delays the communication of its worth  $lworth_a(C)$  for any coalition  $C \subseteq A$  to all other agents. Since  $v(C) = \sum_{a \in C} lworth_a(C)$  holds, at this moment, it is the only agent that can compute all coalition values, hence the complete game  $(A, v)$ . Since  $a$  can influence the coalition formation process according to the KCA protocol only at the beginning of the first negotiation round, it must decide on whether to perform a fraud to unfoundedly increase its profit, or not, right before it communicates its local values to the other agents. For this purpose, it locally simulates the KCA protocol to predict the final, kernel stable solution  $(S, u)$  for  $(A, v)$ . In case of 3-agent coalition games, this predicted configuration is even unique.

How can agent  $a$  increase its predicted utility  $u(a)$  without getting harmed by unmasking claims of other agents within a joint coalition in the predicted final structure  $S$ ? One option is to choose one coalition  $C' \notin S$ , and deceive

all agents on its individual worth  $lworth_a(C')$  for  $C'$  such that the unfoundedly increase of its utility  $u'(a)$  in the corresponding final configuration  $(S', u')$  is maximum. Such a fraud is in principle impossible to detect, since the worth of each agent depends on its individual production utility functions  $U_a$  which can be different for all agents.

With increasing  $v(C')$ , by definition of the KCA the probability that  $C'$  is actually formed also increases. How to determine the highest possible value  $r \in \mathbb{R}, r \neq 0$  by which agent  $a$  can increase its original  $lworth_a(C')$ ? Each change of worth in some round  $t \in \mathbb{N}$ , that is  $lworth_a^{(t)}(C') = lworth_a^{(t-1)}(C') + r^{(t)}$  where  $lworth_a^{(0)}(C') = lworth_a(C')$ , implies a change of the respective coalition value  $v^{(t)}(C')$ . That, in turn, changes the current game  $(A, v^{(t)})$  to  $(A, v^{(t+1)})$  by replacing  $v^{(t)}(C')$  with  $v^{(t+1)}(C') = v^{(t)}(C') + r^{(t)}$ . For each new game  $(A, v^{(t)})$ ,  $t = 1..m$ , agent  $a$  simulates a full run of the KCA protocol until it finds, in round  $m$ , a configuration  $(S_m, u_m)$  such that  $C \in S_m$ . In this case, it holds that  $u^{(m)}(a) \leq u^{(0)}(a) + r^{(m)}$  (cf. lemma 2). But that is adverse to  $a$ , since it would be enforced to bring the additional amount  $r^{(m)}$  into the coalition once it is formed.

**Lemma 2.** *Let  $(A, v)$ ,  $(A, v^*)$  games,  $r \in \mathbb{R}$ , and  $C^* \subseteq A$  with*

$$\forall C \subseteq A : v^*(C) := \begin{cases} v(C) + r & \text{for } C = C^* \\ v(C) & \text{otherwise} \end{cases}$$

*Further, let  $a^* \in C^*$ ,  $(S, u)$  a kernel stable configuration for  $(A, v)$ ,  $u^*$  a payoff distribution for which holds that  $u^*(a) = u(a) + r$ ,  $a \in A$ , if  $a = a^*$ , and  $u^*(a) = u(a)$  otherwise.*

*Then  $(S, u^*)$  is not kernel stable for  $(A, v^*)$ , if there exists an agent  $a^+ \in C^*, a^+ \neq a^*$  such that  $s_{a^+, a^*} - s_{a^+, a^*} < r$  holds in  $(A, v)$ .*

*Proof.* Let  $s_{a_1, a_2}^*$  denote the surplus of agent  $a_1$  over agent  $a_2$  in the game  $(A, v^*)$  and  $Z := \{C | C \subseteq 2^A, a^* \in C, a^+ \notin C\}$ . Further, let  $e^*(C)$  the excess of coalition  $C$  in the game  $(A, v^*)$ . Then

$$\begin{aligned} s_{a^*, a^+}^* &= \max_{C \in Z} \{e^*(C)\} \\ &= \max_{C \in Z} \{v(C) - \sum_{a' \in C, a' \neq a^*} u(a') - u^*(a^*)\} \\ &= \max_{C \in Z} \{v(C) - \sum_{a' \in C, a' \neq a^*} u(a') - (u(a^*) + r)\} \\ &= \max_{C \in Z} \{v(C) - \sum_{a' \in C} u(a')\} - r \\ &= s_{a^*, a^+} - r < s_{a^+, a^*} \end{aligned}$$

But since  $u^*(a^*) > v^*({a^*}) = v({a^*})$ , configuration  $(S, u^*)$  is not kernel stable for  $(A, v^*)$ . ◻

As a consequence, agent  $a$  selects  $r^{(m-1)}$  as the maximum raise of its original worth for  $C'$  to the fraudulent value  $lworth_a^*(C') = lworth_a(C') + r^{(m)}$ . The value  $r^{(m)}$  is the amount of additional profit  $a$  expects to gain by communicating its worth  $lworth_a(C)$  for all possible coalitions  $C$ , including the incorrect value  $lworth_a^*(C')$  for coalition  $C'$ , to all other agents in the first round of the KCA based negotiations.

**Example 5.1:** *Fraudulent manipulation of coalition value*

Consider a game  $(A = \{a_1, a_2, a_3\}, v)$  with  $v(\{a_i\}) = 0, i \in \{1, 2, 3\}$ ,  $v(\{a_1, a_2\}) = 5$ ,  $v(\{a_1, a_3\}) = 1$ ,  $v(\{a_2, a_3\}) = 2$ ,  $v(\{a_1, a_2, a_3\}) = 0$ . Applying the KCA to this game clearly results in the coalition structure  $\{\{a_1, a_2\}, \{a_3\}\}$  and the kernel stable payoff distribution  $u = (2, 3, 0)$ . Now, the question is what happens if agent  $a_1$  by intention deceives the other agents on the true coalition value  $v(\{a_1, a_3\})$  by communicating an artificially increased value of its worth  $lworth_{a_1}^*(\{a_1, a_3\}) = lworth_{a_1}(\{a_1, a_3\}) + r, r \in \mathbb{R}^+$ ?

Consider the respectively changed game  $(A, v^*)$  with  $r = 3$ . Then it holds that  $v^*(\{a_1, a_3\}) = 4$ , and all other coalition values  $v^*$  remain the same as in the original game  $v$ . Again, using the KCA, the coalition structure  $\{\{a_1, a_2\}, \{a_3\}\}$  is formed, but now with different payoff distribution  $u^* = (3.5, 1.5, 0)$ . Thus, the fraud of  $a_1$  has been successful, since it resulted in an increase  $(3.5 - 2 = 1.5)$  of its profit compared with payoff distribution for the original game.

Now consider the case in which agent  $a_1$  decides to increase its worth even more ( $r > 4$ ), say  $r = 5$ . For the resulting game  $(A, v^+)$  it holds that  $v^+(\{a_1, a_3\}) = 6$  and all other coalition values remain the same. This time, using the KCA, a different coalition structure  $\{\{a_1, a_3\}, \{a_2\}\}$  forms with payoff distribution  $u^+ = (4.5, 0, 1.5)$ . Again, it seems that agent  $a_1$  increased its payoff compared with that in the original game by  $4.5 - 2 = 2.5$ . However, agent  $a_1$  now has the severe problem to actually bring in the amount of  $r = 5$  into the formed coalition  $\{a_1, a_3\}$ , since its real increase in profit is given by  $2.5 - 5 = -2.5$ , hence actually a loss! How shall  $a_1$  explain that to its committed coalition partner  $a_3$ ?

o

The computational efforts that are required to decide whether there exists an individually beneficial option to deceive other agents in KCA based coalition negotiation for

a given game are exponentially expensive. Once the predicted configuration  $(S, u)$  is locally computed, agent  $a$  still has to check each of the  $O(2^n)$  alternative coalitions  $C' \notin S$  with polynomial computational complexity of simulated KCA based negotiations for each respective game. Besides, the possibly large delay in communicating its values to the other agents after having received theirs may already draw some initial suspicion of fraud on agent  $a$ .

## 6. Conclusion

We showed that the KCA coalition protocol exhibits both desirable and non-desirable properties with respect to safety and privacy preservation. The possibility for any agent to hide its self-values without risking any decrease of its payoff is clearly an advantageous result for kernel based coalition formation procedures such as the KCA. On the other hand, tasks and information may be hidden from other agents only at the cost of giving up some cooperation opportunities. However, if agents hide certain local values from other agents the course of negotiations according to the KCA can seriously be affected such that non-kernel stable solutions of the original game will form. Another threat to KCA based negotiations is caused by agents that drop out of running negotiations. As a consequence, the remaining agents and those that reenter the negotiation process after a while, may end up with different ideas about the final configuration. This makes it impossible for most agents to abide by their coalition contracts. Finally, it has been shown that individual fraud in kernel stable coalition forming using the KCA is in principle possible but appears impractical in terms of computational complexity.

## References

- [1] B. Blankenburg, M. Klusch, O. Shehory: Fuzzy Kernel-Stable Coalitions Between Rational Agents. Proc. 2nd Intl. Conference on Autonomous Agents and Multiagent Systems (AAMAS 2003), Melbourne, Australia, ACM Press, 2003.
- [2] J. P. Kahan, A. Rapoport: Theories of coalition formation. Lawrence Erlbaum Associates, Hillsdale, NJ, 1984.
- [3] M. Klusch, A. Gerber: Dynamic Coalition Formation among Rational Agents. IEEE Intelligent Systems, 17(3), May/June 2002.
- [4] M. Klusch, O. Shehory: A Polynomial Kernel-Oriented Coalition Algorithm for Rational Information Agents. Proc. 2nd Intl. Conference on Multi-Agent Systems (ICMAS '96), Kyoto (Japan), AAAI Press, 1996.
- [5] G. Owen. Game Theory. Academic Press, NY, 1995.

---

## Negotiation of Fuzzy-Valued Coalitions

B. Blankenburg, M. Klusch, O. Shehory: Fuzzy Kernel-Stable Coalitions Between Rational Agents. Proceedings of the 2nd International Conference on Autonomous Agents and Multiagent Systems (AA-MAS), Melbourne, Australia, pages 9 - 16, ACM Press, 2003.

B. Blankenburg and M. Klusch: BSCA-F: Efficient Fuzzy Valued Stable Coalition Forming Among Agents. Proceedings of the 4th IEEE Conference on Intelligent Agent Technology (IAT), Compiègne, France, IEEE Computer Society Press, 2005.

# Fuzzy Kernel-Stable Coalitions Between Rational Agents

Bastian Blankenburg\*  
DFKI - German Research  
Center for Artificial Intelligence  
Stuhlsatzenhausweg 3  
66123 Saarbrücken, Germany  
blankenb@dfki.de

Matthias Klusch  
DFKI - German Research  
Center for Artificial Intelligence  
Stuhlsatzenhausweg 3  
66123 Saarbrücken, Germany  
klusch@dfki.de

Onn Shehory  
IBM - Haifa Research Lab  
Tel Aviv Site, Haifa University  
Mount Carmel, Haifa  
31905 Israel  
onn@il.ibm.com

## ABSTRACT

A large variety of solutions exists for the problem of coalition formation among autonomous agents, at the theoretical level within game theory, and at the practical, algorithmic level, within multi-agent systems. However, one major underlying assumption of algorithmic solutions suggested to date is that the values of the coalitions are known and are certain at the time of coalition formation negotiation. In many practical cases such as in open, dynamically changing environments this assumption does not hold. In this paper we propose an algorithmic solution to the coalition formation problem that overcomes this limitation of previous solutions. Our solution supports fuzzy coalition values and allows agents to form stable coalition configurations. For this, we combine concepts from the theory of fuzzy sets with the game-theoretic stability concept of the Kernel to deduce the new concept of a fuzzy Kernel. We further provide a low-complexity algorithm for forming fuzzy Kernel stable coalitions among agents.

## Keywords

Fuzzy cooperative games, coalition formation, fuzzy kernel

## 1. INTRODUCTION

Cooperation within coalitions allows agents to perform tasks that they may otherwise be unable to perform. Cooperative game theory provides a well developed and mathematically founded theory according to which one can determine which coalitions are beneficial and what coalition configurations are stable and (Pareto) optimal. Game theory itself, however, does not provide algorithms to be used as a coalition formation process. Such algorithms are investigated in the field of multi-agent systems. In recent years, several coalition formation algorithms were proposed, some concentrating on agents that attempt to maximize group

\*Student author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS'03, July 14–18, 2003, Melbourne, Australia.  
Copyright 2003 ACM 1-58113-683-8/03/0007 ...\$5.00.

utility (e.g., [9]), others addressing self-interested agent that attempt to maximize individual utilities (e.g., [10]). Some solutions tried to reduce complexity [6], compromising optimality, whereas other, exponential solutions sought optimality, proving that without an exponential complexity the solution may be far from optimal [8]. All of these solutions assumed that the values of the coalitions are known for certain before the coalitions are formed and during the formation process. However, in many real-world environments, this assumption does not hold, and values may be known only to a limited degree of certainty. Thus, existing coalition formation methods are inapplicable. In this paper, we address exactly this problem. In particular, we propose a model and an algorithm that allow self-interested agents to form stable coalitions in the face of uncertain values. We do so by fuzzifying concepts of cooperative game theory to allow specification of uncertain coalition values. In our solution, payoff calculation incorporates fuzzy quantities instead of real numbers. Similar work [7] fuzzified the core and the Shapely value. Here, we present a fuzzified version of the Kernel [3], a set-based stability concept which yields stable solutions for every game. The fuzzified Kernel has an intuitively similar interpretation as the crisp Kernel has, however it extends it to contain information about the degree of certainty to which a fuzzy configuration is Kernel-stable. We show that the computational complexity of the fuzzy Kernel is similar to the complexity of the crisp Kernel. We further show that the originally exponential complexity can be reduced to polynomial complexity by placing a cap on the size of coalitions. Finally, we exploit this property to present a polynomial coalition formation algorithm based on the fuzzy kernel.

The paper begins with an introduction to cooperative games and the Kernel (section 2), followed by an introduction to fuzzy quantities in section 3. It proceeds with fuzzifications of the Kernel and related concepts in section 4, and then discusses computational complexity issues in section 5. Sections 6 and 7 present a corresponding fuzzified transfer scheme for side-payments and the algorithm for forming fuzzy Kernel stable coalitions among agents, respectively. We conclude the paper with an outline of future work in section 8.

## 2. CRISP KERNEL-STABLE GAMES

In order introduce fuzziness to the cooperative game-theoretic concepts needed for a fuzzified kernel, we briefly remember their original, crisp definitions here.

*Definition 1.* A cooperative game in characteristic function form is a pair  $(A, v)$  with the set of agents  $A$  and the characteristic function  $v : 2^A \mapsto \mathbb{R}$ .  $v(C)$  is called the value of the coalition  $C \subset A$ .  $v(\emptyset) := 0$ .

This value of a coalition  $C$  can be viewed as a measure of the payoff achievable by  $C$  by cooperating behaviour of its members.

*Definition 2.* A configuration  $(\mathcal{C}, u)$  for a game  $(A, v)$  specifies a payoff distribution  $u : A \mapsto \mathbb{R}$  for a coalition structure  $\mathcal{C}$ , a partition of  $A$ .  $u(a)$ ,  $a \in A$  denotes the payoff for agent  $a$ .  $u$  is called *individually rational* iff  $\forall a \in A : u(a) \geq v(a)$  and *efficient* iff  $\forall C \in \mathcal{C} : \sum_{a \in C} u(a) = v(C)$

A solution to a game is given by an individually rational and efficient configuration which satisfies a chosen *stability concept*. In this paper we focus on the stability concept of the *kernel*. It is based on the idea that the members of a coalition should be in an equilibrium regarding their "power to object" each others payoff. This power is called *surplus* of an agent over another. It is measured by regarding coalitions not in the coalition structure of the configuration  $(\mathcal{C}, u)$  in question. The *excess* of such a coalition is the additional payoff its members can gain by actually forming this coalition with respect to their original payoffs according to  $u$ .

*Definition 3.* The *excess*  $e(C^*, u)$  of a coalition  $C^* \notin \mathcal{C}$  in the configuration  $(\mathcal{C}, u)$  is given by

$$e(C^*, u) := v(C^*) - \sum_{a \in C^*} u(a)$$

The surplus of an agent  $a_i$  over another agent  $a_k$  is then defined as the maximum excess of all coalitions including agent  $a_i$  but without agent  $a_k$ .

*Definition 4.* The *surplus*  $s_{ik}$  of an agent  $a_i$  over agent  $a_k$  with  $a_i, a_k \in C \in \mathcal{C}, a_i \neq a_k$ , is defined as

$$s_{ik} := \max\{e(C^*, u) \mid C^* \notin \mathcal{C}, a_i \in C^*, a_k \notin C^*\}$$

Please note that this implies the assumption that  $a_i$  would be able to gain all of the excess of each considered coalition if it was realized. This might be considered as not very realistic, because the other agents in the coalition would be likely to claim a part of the additional profit for themselves. However, because in the definition of the kernel the surplus is used more like an index but an accurate measurement, this seems to be acceptable.

An agent  $a_i$  is said to dominate agent  $a_k$  in the configuration  $(\mathcal{C}, u)$  iff  $a_i, a_k \in C \in \mathcal{C}$ ,  $s_{ik} > s_{ki}$  and  $u(a_k) > v(a_k)$  hold. The last condition means that agent  $a_k$  really has to lose something, i.e. he is better off staying in his current coalition than acting alone. The kernel is then defined as the set of configurations in which no agent dominates another.

*Definition 5.* The *kernel*  $K$  of a game  $(A, v)$  is defined as

$$K := \left\{ (\mathcal{C}, u) \mid \forall a_i, a_k \in C \in \mathcal{C} : \left[ (s_{ik} = s_{ki}) \right. \right. \\ \left. \left. \vee (s_{ik} > s_{ki} \wedge u(a_k) = v(\{a_k\})) \right. \right. \\ \left. \left. \vee (s_{ki} > s_{ik} \wedge u(a_i) = v(\{a_i\})) \right] \right\}$$

Thus, a configuration  $(\mathcal{C}, u)$  is called a kernel-stable solution of a game  $(A, v)$  iff  $u$  is individually rational and efficient and  $(\mathcal{C}, u)$  is an element of the kernel  $K$  of  $(A, v)$ .

### 3. FUZZY QUANTITIES

The concept of fuzzy subsets was first introduced by Lotfi A. Zadeh in [13]. It emerged from his idea that it should be possible for elements of a set to belong to a fuzzy subset only to a certain degree. The actual meaning of this degree is application-dependant. For instance, it might be a degree of truth ("truth value") or a possibility in the sense of possibility theory. In the following, we assume some possibilistic interpretation of the fuzziness.

*Definition 6.* A *fuzzy subset*  $F$  of a set  $M$  is defined by its *membership function*  $\mu_F : M \mapsto [0, 1]$  where  $\mu_F(x)$ ,  $x \in M$  is called the *degree of membership* or *membership value* of  $x$  in  $M$ .

We define some further fuzzy concepts to be used later.

*Definition 7.*

1. Let  $F$  be a fuzzy subset of some set  $M$ , and  $x \in M$ . We write  $x \in F$  iff  $\mu_F(x) > 0$  and define  $support(F) := \{x \mid x \in F\}$
2. Any fuzzy subset of  $\mathbb{R}$  is called a *fuzzy quantity*.
3. Let  $F$  be a fuzzy quantity.  $size(F) := \sup\{support(F)\} - \inf\{support(F)\}$
4. A fuzzy quantity  $F$  is called *normalized* iff  $\sup_{x \in \mathbb{R}}\{\mu_F(x)\} = 1$
5.  $x \in \mathbb{R}$  with  $\mu_F(x) = \max_{y \in \mathbb{R}}(\mu_F(y))$  is called a *modal value* of  $F$ .
6.  $\mathbb{R}^F$  denotes the set of all fuzzy quantities.
7.  $r^F$  denotes a fuzzy quantity with

$$\mu_{r^F}(x) = \begin{cases} 1 & \text{if } x = r \\ 0 & \text{otherwise} \end{cases}, r, x \in \mathbb{R}$$

8. A *fuzzy interval*  $I$  is a fuzzy quantity with  $\forall x_1, x_2, x_3 \in \mathbb{R}, x_1 < x_2 < x_3 : \mu_I(x_2) \geq \min(\mu_I(x_1), \mu_I(x_3))$
9. Any fuzzy interval  $N$  satisfying (a) there exists exactly one  $x_m \in \mathbb{R} : \mu_N(x_m) = 1$  and (b)  $\exists x_1, x_2 \in \mathbb{R}, x_1 < x_m < x_2 : \mu_N(x) = 0$  for all  $x \in \mathbb{R} \setminus [x_1, x_2]$  is called a *fuzzy number*.
10. A *triangular shaped fuzzy number*  $(x, y, z)^F$ ,  $x, y, z \in \mathbb{R}$  is a fuzzy number with

$$\mu_{(x,y,z)^F}(r) = \begin{cases} \frac{r-x}{y-x} & \text{if } x < r \leq y \\ \frac{z-r}{z-y} & \text{if } y < r < z \\ 0 & \text{otherwise} \end{cases}, r \in \mathbb{R}$$

For fuzzy quantities and numbers, arithmetic operations can be defined following Zadeh's *extension principle*. For the fuzzy kernel, we need the operations of addition, negation, subtraction and multiplication with a crisp number.

*Definition 8.* Let  $F_1, F_2 \in \mathbb{R}^F$ ,  $x, y, z, a \in \mathbb{R}$ .

$$\begin{aligned} \mu_{F_1 \oplus F_2}(x) &:= \sup\{\min(\mu_{F_1}(y), \mu_{F_2}(z)) \mid y + z = x\} \\ \mu_{-F_1}(x) &:= \mu_{F_1}(-x) \\ \mu_{F_1 \ominus F_2}(x) &:= \mu_{F_1 \oplus (-F_2)}(x) \\ \mu_{a \cdot F_1}(x) &:= \begin{cases} \mu_{F_1}(x/a) & \text{if } a \neq 0 \\ \mu_0 & \text{if } a = 0 \end{cases} \end{aligned}$$

Note that for additions and subtractions on  $F_1$  and  $F_2$  with a result  $F_3$ , we have  $size(F_3) = size(F_1) + size(F_2)$ .

The extension principle is also used to define the fuzzy extension of the max function.

*Definition 9.* Let  $F_1, F_2 \in \mathbb{R}^F$ ,  $x, y, z \in \mathbb{R}$ .

$$\mu_{\widetilde{\max}(F_1, F_2)}(x) := \sup\{\min(\mu_{F_1}(y), \mu_{F_2}(z)) \mid \max(y, z) = x\}$$

For convenience, let

$$\begin{aligned} \widetilde{\max}(\{x_1, \dots, x_n\} \subset \mathbb{R}^F) := \\ \widetilde{\max}(x_1, \widetilde{\max}(\dots, \widetilde{\max}(x_{n-1}, x_n) \dots)) \end{aligned}$$

and  $\widetilde{\max}(\{F_1\}) := F_1$

From the definition of the crisp kernel, it is clear that we will need to compare fuzzy quantities to each other in order to determine whether a fuzzy quantity  $A$  is greater than a fuzzy quantity  $B$ . We cannot use  $\widetilde{\max}$  for this comparison because it constructs a new fuzzy quantity out of the membership functions of its operands rather than choosing one of the two to be maximal. Thus, we will need a so-called *ranking method* for fuzzy quantities. Several of such methods have been proposed in the literature (see for example [1]). In section 4, the fuzzy kernel is defined such that configurations are to some degree fuzzy kernel-stable. So we need a ranking operator  $R$  that yields a certain degree of a possibilistic measure to a comparison between two fuzzy quantities and thus is a fuzzy subset of  $\mathbb{R}^F \times \mathbb{R}^F$ .

*Definition 10.* Let  $F_1, F_2 \in \mathbb{R}^F$  and  $R$  be a fuzzy subset of  $\mathbb{R}^F \times \mathbb{R}^F$ .  $R$  is called a *fuzzy ranking operator* if  $\mu_R(F_1, F_2)$  denotes the degree to which  $F_1$  can be considered "greater" compared to  $F_2$ .  $R$  is called a *fuzzy similarity relation* if  $\mu_R(F_1, F_2)$  denotes the degree to which  $F_1$  can be considered "similar" to  $F_2$ . Further let  $G$  be a fuzzy ranking operator and  $S$  a fuzzy similarity relation. We define

$$\begin{aligned} (F_1 \succ_G F_2) &:= \mu_G(F_1, F_2) \\ (F_1 \approx_S F_2) &:= \mu_S(F_1, F_2) \end{aligned}$$

In the following, we use the "possibility of dominance"  $PD$ , which was introduced in [4], as an instance of  $G$ . It is defined as

$$F_1, F_2 \in \mathbb{R}^F : (F_1 \succ_{PD} F_2) := \sup_{x, y \in \mathbb{R}, x \geq y} \{\min(\mu_{F_1}(x), \mu_{F_2}(y))\}$$

As an instance of  $S$ , we use the analogously defined similarity relation  $PS$  with

$$(F_1 \approx_{PS} F_2) := \sup_{x \in \mathbb{R}} \{\min(\mu_{F_1}(x), \mu_{F_2}(x))\}$$

Further we define a set of maximal elements of a set of fuzzy quantities in the way Mareš did it in [7].

*Definition 11.* Let  $X$  be a set of fuzzy quantities and  $G$  a fuzzy ranking operator. The fuzzy subset  $X_{\max G}$  of  $X$  is given by

$$\forall F_1 \in X : \mu_{X_{\max G}}(F_1) := \min_{F_2 \in X} (F_1 \succ_G F_2)$$

Thus,  $\mu_{X_{\max G}}(F_1)$  denotes the degree to which  $F_1$  can be considered a maximal element of  $X$ . For convenience,

$$\max^G X := X_{\max G}.$$

Finally, we will need the logical operations "AND" and "OR" with operands  $\in [0, 1]$ .

*Definition 12.* Let  $x, y \in [0, 1] : x \wedge y := \min(x, y)$  and  $x \vee y := \max(x, y)$

## 4. FUZZY KERNEL-STABLE GAMES

As we indicated in section 1, negotiation during the coalition forming process might face uncertainties. Such uncertainties could be caused by the possibility of nondeterministic events that hamper the negotiation process and produce incomplete information regarding the values of coalitions. This leads us to the formation of fuzzy-valued coalitions (see also [7]). Other approaches to introduce fuzziness into game theory include the formation of fuzzy coalitions, where agents are members of (multiple) coalitions to a certain degree (see [2]). For fuzzy-valued coalitions, the characteristic function of the game maps to fuzzy values. Thus, the definitions of the basic game-theoretic concepts as given in section 2 have to be revised for the fuzzy-valued case.

*Definition 13.* A *fuzzy cooperative game* in characteristic function form is a pair  $(A, v^F)$  with the set of agents  $A$  and the fuzzy characteristic function  $v^F : 2^A \mapsto \mathbb{R}^F$ .  $v^F(C)$  is called the fuzzy value of the coalition  $C$ .

In the following we say just "fuzzy game" instead of "fuzzy cooperative game".

*Example 1.* We give a simple example of a fuzzy game definition with four information agents:

$$A := \{a_1, a_2, a_3, a_4\}$$

Let us suppose these agents have accepted a task to deliver some information to their respective users. The agents' evaluation of the relevance of the available information items to the search tasks is uncertain. The agents now collect as much possibly relevant information as they are able to obtain. An agent can obtain such information either by looking it up in its local database, or by cooperation with another agent. It finally offers the collected information items to its user, who pays some price to get access to those items which are relevant to him/her.

The uncertainty of the relevance of information items to the search task is expressed by the use of fuzzy numbers to summarize the value of a bundle of information items. Thus, let  $I_d^s$  denote the fuzzy value of the information items which are locally stored by agent  $s$  with respect to the search task of agent  $d$ . For example, let's assume agent  $a_1$  has locally stored some information items which he can sell to its own user for "about" 3.5 (monetary units). Further, it is assumed to be certain that  $a_1$  will get no less than 3.0 and no more than 4.0 for them. Thus, we summarize his local information's value for his search task with the triangular fuzzy number  $I_{a_1}^{a_1} := (3, 3.5, 4)^F$ . The other values of game-relevant bundles of information items are assumed to be obtained in a similar way and given in table 1. The last line of this table states the sum of each agent's game-relevant information which gives an easy but very rough indication of an agent's "power" in the upcoming negotiations. For simplicity, we assume that every information is locally available to at most one agent in the game. We restrict size of profitable coalitions to max. two agents by defining  $v^F(C_{3,4}) := 0^F$  for all three- and four-agent coalitions  $C_{3,4}$ . The fuzzy coalition value of a one- or two-agent coalition  $C$  is defined as the sum of the fuzzy values of all game-relevant information items available in the coalition:  $v^F(C) := \sum_{a_1 \in C}^{\oplus} \sum_{a_2 \in C}^{\oplus} I_{a_1}^{a_1}, 1 \leq |C| \leq 2$ . The game is

$I_d^s$	$s = a_1$	$s = a_2$
$d = a_1$	$(3, 3.5, 4)^F$	$(3, 3.25, 3.5)^F$
$d = a_2$	$(2, 2.25, 2.5)^F$	$(4, 5, 6)^F$
$d = a_3$	$(0.5, 0.65, 2)^F$	$(1.6, 1.75, 1.9)^F$
$d = a_4$	$(5, 5.25, 6)^F$	$(10, 10.5, 11)^F$
$\sum^\oplus$	$(10.5, 11.65, 14.5)^F$	$(18.6, 20.5, 22.4)^F$

$I_d^s$	$s = a_3$	$s = a_4$
$d = a_1$	$(0, 0.1, 1)^F$	$(7, 7.25, 8)^F$
$d = a_2$	$(0.4, 0.5, 0.6)^F$	$(5, 6, 7)^F$
$d = a_3$	$(0.5, 0.75, 1)^F$	$(2.5, 2.9, 3.25)^F$
$d = a_4$	$(1, 1.1, 1.25)^F$	$(2, 2.5, 4)^F$
$\sum^\oplus$	$(1.9, 2.45, 3.85)^F$	$(16.5, 18.65, 22.25)^F$

Table 1: Fuzzy values of information items

summarized as follows:

$$\begin{aligned}
v^F(a_1) &= (3, 3.5, 4)^F & v^F(a_2) &= (4, 5, 6)^F \\
v^F(a_3) &= (0.5, 0.75, 1)^F & v^F(a_4) &= (2, 2.5, 4)^F \\
v^F(a_1, a_2) &= (12, 14, 16)^F & v^F(a_1, a_3) &= (4, 5, 8)^F \\
v^F(a_1, a_4) &= (17, 18.5, 22)^F & v^F(a_2, a_3) &= (6.5, 8, 9.5)^F \\
v^F(a_2, a_4) &= (21, 24, 28)^F & v^F(a_3, a_4) &= (6, 7.25, 9.5)^F \\
v^F(a_1, a_2, a_3) &= 0^F & v^F(a_1, a_2, a_4) &= 0^F \\
v^F(a_1, a_3, a_4) &= 0^F & v^F(a_2, a_3, a_4) &= 0^F \\
v^F(a_1, a_2, a_3, a_4) &= 0^F & &
\end{aligned}$$

With the coalition values being fuzzy, it seems plausible to also fuzzify the payoff distribution: no crisp payoff can be guaranteed as long as it is not known which coalition values will be realized.

*Definition 14.* A fuzzy configuration is a pair  $(\mathcal{C}, u^F)$  with the (crisp) coalition structure  $\mathcal{C}$  and the fuzzy payoff distribution  $u^F : A \mapsto \mathbb{R}^F$ . With a fuzzy coalition values and payoff distributions, the concepts of individual rationality and efficiency also become fuzzy: let  $G$  be a fuzzy ranking operator and  $S$  a fuzzy similarity relation.

$a \in A : \mu_{\text{indrat}^G}(a) := u^F(a) \succ_G v^F(a)$  denotes the degree of fuzzy individual  $G$ -rationality of agent  $a$ .

$\mu_{\text{indrat}^G}(u^F) := \bigwedge_{a \in A} \{\mu_{\text{indrat}^G}(a)\}$  denotes the degree of fuzzy individual  $G$ -rationality of  $u^F$ .

$\mu_{\text{eff}^S}(u^F) := \bigwedge_{C \in \mathcal{C}} \left\{ \sum_{a_i \in C}^\oplus u^F(a_i) \approx_S v^F(C) \right\}$  denotes the degree of fuzzy  $S$ -efficiency of  $u^F$ .

*Example 2.* With  $\mathcal{C} := \{\{a_1\}, \{a_3\}, \{a_2, a_4\}\}$ ,  $u^F(a_1) = v^F(\{a_1\})$ ,  $u^F(a_2) = (8.9, 10.4, 12.4)^F$ ,  $u^F(a_3) = v^F(\{a_3\})$  and  $u^F(a_4) = (12.1, 13.6, 15.6)^F$ , let  $(\mathcal{C}, u^F)$  be a fuzzy configuration for the fuzzy game defined in example 1. Here  $a_2$ 's and  $a_4$ 's payoffs are certainly greater than their respective single-agent coalition values, thus the degree of fuzzy individual  $PD$ -rationality is 1.0. The degree of fuzzy  $PS$ -efficiency is also 1.0 because  $u^F(a_2) \oplus u^F(a_4) = v^F(\{a_2, a_4\})$ .

Fuzzy coalition stability concepts define a degree to which given fuzzy configurations are stable. Thus, we define solutions of a fuzzy game as set of fuzzy configurations that satisfy given minimal requirements on the degrees of stability, individual rationality and efficiency.

*Definition 15.* Let  $\mu_{\text{stable}^{SC}}(\mathcal{C}, u^F)$  denote the degree to which a fuzzy configuration  $(\mathcal{C}, u^F)$  of a fuzzy game  $(A, v^F)$  is stable according to some fuzzy stability concept  $SC$ . Let  $ir_{\min}, ef_{\min}, st_{\min} \in [0, 1]$ ,  $G$  a fuzzy ranking operator and  $S$  a fuzzy similarity relation. The configuration  $(\mathcal{C}, u^F)$  is called a  $(ir_{\min}, ef_{\min}, st_{\min}, G, S, SC)$ -solution of the fuzzy game  $(A, v^F)$  iff  $\mu_{\text{indrat}^G}(u^F) \geq ir_{\min}$ ,  $\mu_{\text{eff}^S}(u^F) \geq ef_{\min}$  and  $\mu_{\text{stable}^{SC}}(\mathcal{C}, u^F) \geq st_{\min}$

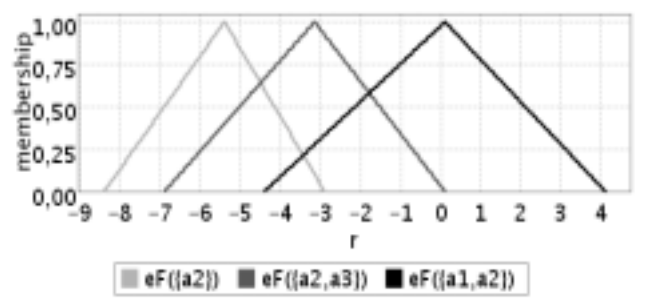


Figure 1: Membership functions of the maximum fuzzy excesses of agent  $a_2$  excluding  $a_4$

As we have seen in section 2, the definition of the kernel is based on the concepts of excess and surplus. For the definition of a fuzzy kernel, these concepts also need to be fuzzified. In the case of the excess, this is straight-forward.

*Definition 16.* The fuzzy excess  $e^F(C^*, u^F)$  of a coalition  $C^* \notin \mathcal{C}$  in the fuzzy configuration  $(\mathcal{C}, u^F)$  is defined as

$$e^F(C^*, u^F) := v^F(C^*) \ominus \sum_{a \in C^*}^\oplus u^F(a)$$

To define the fuzzy surplus, however, we have to keep in mind that the maximum of a set of fuzzy quantities is a fuzzy subset. Thus, the fuzzy surplus should be an element of the (fuzzy) set of maximal fuzzy excesses. However, there could be more than one excess with maximal membership value in this set. Therefore, we will define the surplus as the  $\widetilde{\max}$  of the maximal excesses.

*Definition 17.* Let  $E_{ik}^F$  be the set of fuzzy excesses of an agent  $a_i$  excluding agent  $a_k$  in the fuzzy configuration  $(\mathcal{C}, u^F)$ :

$$E_{ik}^F := \{e^F(C^*, u^F) \mid C^* \notin \mathcal{C}, a_i \in C^*, a_k \notin C^*\}$$

Let  $G$  be a fuzzy ranking operator. The fuzzy  $G$ -surplus  $s_{ik}^{FG}$  in  $(\mathcal{C}, u^F)$  of agent  $a_i$  over agent  $a_k$  is then defined as

$$s_{ik}^{FG} := \widetilde{\max}^G(\max^G(E_{ik}^F))$$

*Example 3.* For the fuzzy configuration given in example 2, we consider  $\max^{PD}(E_{24}^F)$ , the set of maximal fuzzy excesses of agent  $a_2$  excluding agent  $a_4$ , which is illustrated in figure 1.  $e^F(\{a_1, a_2, a_3\})$  is not included because its support lies completely to the left of the other three excesses which overlap pairwise. Then,  $s_{24}^{FG} = \widetilde{\max}^G(\max^G(E_{24}^F)) = e^F(\{a_1, a_2\})$

Now, we are able to give a definition of the fuzzy kernel by just substituting the crisp terms and operators of the definition of the crisp kernel to their fuzzy counterparts.

*Definition 18.* Let  $G$  be a fuzzy ranking operator and  $S$  a fuzzy similarity relation. The fuzzy  $(G, S)$ -kernel  $K^{FG, S}$  of a fuzzy game  $(A, v^F)$  is defined by its membership function  $\mu_K^{FG, S} : (\mathcal{C}, u^F) \mapsto [0, 1]$  with

$$\begin{aligned}
\mu_{K^{FG, S}}(\mathcal{C}, u^F) &:= \bigwedge_{a_i, a_k \in \mathcal{C} \in \mathcal{C}} \left\{ (s_{ik}^{FG} \approx_S s_{ki}^{FG}) \right. \\
&\quad \vee (s_{ik}^{FG} \succ_G s_{ki}^{FG} \wedge u^F(k) \approx_S v^F(\{a_k\})) \\
&\quad \left. \vee (s_{ki}^{FG} \succ_G s_{ik}^{FG} \wedge u^F(i) \approx_S v^F(\{a_i\})) \right\}
\end{aligned}$$

This definition is perfectly in accordance with the definitions of the crisp kernel; there, the surplus is defined by means of the maximum excess of *possible* coalitions. Agents are further assumed to gain all of the excess, as it was mentioned above. So the crisp excess could be seen as an amount that an agent could *possibly* (but maybe not likely) gain if the corresponding coalition was realized. Thus, already the crisp kernel has some kind of possibilistic interpretation. This is just extended here to the values of the excesses themselves.

Clearly, the actual membership values of configurations in the fuzzy kernel heavily depend on the actual choice for the ranking operator  $G$  and similarity relation  $S$ . Many have been proposed in the literature and most of them arrive at questionable results in difficult cases, e.g. if non-normalized fuzzy quantities are involved. Thus, choosing the "right" ranking method should be done with respect to a given fuzzy game. Even the possible range of  $\mu_{K^{FG,S}}$  depends on that choice and on the membership functions of the coalition values. For example, if we use  $PD$  and  $PS$ , and some of the coalition values are non-normalized, a configuration  $(C, u^F)$  with  $\mu_{K^{FPD,PS}}(C, u^F) = 1$  might not exist. For practical applications it might thus be necessary to allow only normalized fuzzy quantities.

*Example 4.* Consider the fuzzy configuration of example 2 again. The most interesting fuzzy comparison in this configuration is  $(s_{24}^{FPD} \approx_{PS} s_{42}^{FPD}) \approx 0.84$ . It causes the degree of fuzzy  $(PD, PS)$ -kernel-stability to be 0.84. As we have seen in example 2, the degrees of fuzzy  $PS$ -efficiency and individual  $PD$ -rationality are 1.0. Thus, the configuration is a  $(1.0, 1.0, 0.84, PD, PS, K^{FPD,PS})$ -solution

For a comparison with the definition of the crisp kernel, consider a game where all coalition values and the values of every payoff function are of the form  $r^F, r \in \mathbb{R}$ . Then again take  $G := PD$ , and  $S := R$  with for fuzzy quantities  $F_1 = r_1^F, F_2 = r_2^F$  let

$$(F_1 \approx_R F_2) := \begin{cases} 1 & \text{if } r_1 = r_2 \\ 0 & \text{otherwise} \end{cases}$$

Then the sets of maximal excesses have exactly one element, with a membership value = 1. Because the result of an addition or subtraction of fuzzy quantities of the form  $r^F$  is still of this form, the fuzzy surpluses are also such. Further, all of the comparisons involved to calculate the membership value of a fuzzy configuration in the fuzzy kernel result in degrees of either 0 or 1, resulting in membership values of 0 or 1. Thus, in such a situation, the fuzzy kernel yields, roughly spoken, the same result as the crisp kernel if for every fuzzy quantity  $r^F$  in the fuzzy game  $r$  is used in the crisp game. If a "real" fuzzy game is considered and the fuzziness is interpreted as possibility, the degree to which a fuzzy configuration is contained in the fuzzy kernel can be interpreted as the possibility that the configuration is kernel-stable upon realization of the game. "Realization" means that the agents have actually formed coalitions and executed their strategies so that the actual, non-fuzzy coalition values are known in the end.

## 5. COMPLEXITY

It is clear that in the general case, an actual computation of  $\mu_K(C, u^F)$  needs exponential time with respect to the number of agents in the game because this was already

the case with crisp games. However, it is worth to have a closer look on the complexity in order to point out the parts where optimizations and improvements could be made under certain assumptions.

LEMMA 1. For a fuzzy game  $(A, v^F)$ , a fuzzy configuration  $(C, u^F)$  and a fuzzy quantity  $F \in \mathbb{R}^F$  let

$$n_{agents} = |A|$$

$$n_{fuzzy\max} = \max(\max_{C \in 2^A} \{size(v^F(C))\}, \max_{a \in A} \{size(u^F(a))\})$$

The membership value of  $(C, u^F)$  in the kernel  $K$  for  $(A, v^F)$  can be computed in time

$$Comp_{kernel} = O(2^{2n_{agents}} \cdot n_{agents}^5 \cdot n_{fuzzy\max}^2)$$

Sketch of the proof:

The complexity of arithmetic operations and ranking operators on fuzzy quantities  $F_1$  and  $F_2$  can be assumed to be  $O(size(F_1) \cdot size(F_2))$  for approximate calculations with general fuzzy quantities.

For the calculation of a fuzzy excess  $e(C, u^F)$ , at most  $n_{agents}$  fuzzy subtractions and 1 fuzzy addition are required. The maximum size of the fuzzy quantities dealt with during the calculation can reach  $(n_{agents} + 1) \cdot n_{fuzzy\max}$ . The complexity of an operation on fuzzy quantities can thus be bounded by  $Comp_{fuzzyop} = O((n_{agents} \cdot n_{fuzzy\max})^2)$ .

Then, the complexity of a calculation of a fuzzy excess can be summarized with  $Comp_{excess} = O(n_{agents} \cdot Comp_{fuzzyop}) = O(n_{agents}^3 \cdot n_{fuzzy\max}^2)$ .

To compute the fuzzy surplus  $s_{ik}^F$ , the fuzzy excesses of an agent will have to be cross-compared. For a set of fuzzy quantities  $X$  of size  $m = |X|$ ,  $m^2$  comparisons are made. Since the number of excesses is exponential with respect to  $n_{agents}$ , the set of maximal excesses is found in time  $O(2^{2n_{agents}} \cdot Comp_{fuzzyop})$ . Then, there are maximum  $2^{n_{agents}-1}$   $\widetilde{\max}$  operations, which is covered by  $O(2^{2n_{agents}} \cdot Comp_{fuzzyop})$ . Thus, the complexity for the calculation of each fuzzy surplus is  $Comp_{surplus} = 2^{n_{agents}} \cdot Comp_{excess} + O(2^{2n_{agents}} \cdot Comp_{fuzzyop}) = O(2^{2n_{agents}} \cdot n_{agents}^3 \cdot n_{fuzzy\max}^2)$ .

For the fuzzy kernel the complexities of  $\vee$  and  $\wedge$  are  $O(1)$ . Then, the complexity of a membership value calculation of fuzzy kernel turns out to be  $Comp_{kernel} \leq n_{agents}^2 \cdot (2 \cdot Comp_{surplus} + 5 \cdot Comp_{fuzzyop}) = O(2^{2n_{agents}} \cdot n_{agents}^5 \cdot n_{fuzzy\max}^2)$ .

However, polynomial time can be achieved by limiting the size of the considered coalitions like it was done by Klusch and Shehory (see also [6]). This is especially plausible in situations where communication costs prevent coalitions larger than a certain size to be profitable.

## 6. TRANSFER SCHEMES

To actually compute stable configurations, iterative techniques have been developed for the crisp case (see also [11]). These are called transfer schemes and specify a sequence of payoff-distributions converging at a stable configuration for a given coalition structure.

For the kernel of a crisp game  $(A, v)$  with the configuration  $(C, u)$ , an upper bound  $t_{ki}^{max}$  for a transfer  $t_{ki}$  of agent  $a_k \in A$  to agent  $a_i \in A$  in one step of the sequence is thereafter given by

$$t_{ki}^{max} := \begin{cases} \min((s_{ik} - s_{ki})/2, u(k) - v(k)) & \text{if } s_{ik} > s_{ki} \\ 0 & \text{otherwise} \end{cases}$$



The transfer is sometimes also called *side-payment*. However, this term might be confusing because this amount is never really paid between the agents. It is only used to compute a stable configuration. In the end, only this stable configuration is part of the solution, and not the sequence of configurations to get there. This is especially important to note when we now come to transfer schemes for fuzzy configurations. A transfer in a fuzzy game is assumed to be a fuzzy quantity, but might be of a crisp form  $r^F, r \in \mathbb{R}$ . If a fuzzy transfer  $t^F$  with  $size(t^F) > 0$  is applied to its corresponding configuration, further fuzziness will be introduced into the game, making the transferring algorithm itself fuzzy. Also, a later defuzzification might be affected.

We here give a transfer scheme for fuzzy games where only fuzzy numbers are allowed, and  $PD$  and  $PS$  are used as ranking operator and similarity relation. This transfer scheme then yields an upper bound for  $r, r \in \mathbb{R}$  of a transfer which is of the form  $r^F$ . In this case, we can ensure the minimum degree of individual rationality of the agent  $a_k$  who is the source of the transfer, by examining the rightmost point where  $\mu_{u^F(a_k)} = ir_{min}$  and the leftmost point where  $\mu_{v^F(\{a_k\})} = ir_{min}$ , because if the modal value of  $u^F(a_i)$  is less than that of  $v^F(a_i)$ ,  $u^F(a_i) \succ_{PD} v^F(\{a_i\})$  is given by the intersection of the right side of  $u^F(a_i)$  with the left side of  $v^F(\{a_i\})$ . The crisp transfer scheme can then be reformulated to operate on the modal values:

*Definition 19.* Let  $\check{r}$  denote a real value  $r$  such that for the fuzzy number  $F$ ,  $\mu_F(r)$  is the modal value of  $F$ . To find a  $(ir_{min}, ef_{min}, st_{min}, PD, PS, K^{F^{PD, PS}})$ -solution for a fuzzy game  $(A, v^F)$  where only fuzzy numbers are allowed for  $v^F$  and  $u^F$ , given the fuzzy configuration  $(\mathcal{C}, u^F)$  with  $\mu_{effs}(u^F) \geq ef_{min}$  and  $\mu_{indrat}^{PD}(u^F) \geq ir_{min}$ , for a transfer  $t_{ki}$  of agent  $a_k \in A$  to agent  $a_i \in A$  which is of the form  $r^F, r \in \mathbb{R}$  an upper bound for  $r$  is given by:

$$t_{ki}^{r_{max}} := \begin{cases} \min((s_{ik}^{F^{PD}} - s_{ki}^{F^{PD}})/2, t_{ki}^{r_{max} ir_{min}}) \\ \text{if } (s_{ik}^{F^{PD}} \succ_{PD} s_{ki}^{F^{PD}}) > (s_{ki}^{F^{PD}} \succ_{PD} s_{ik}^{F^{PD}}) \\ 0 \text{ otherwise} \end{cases}$$

with

$$\begin{aligned} u_{Right}^{F} ir_{min}(a_k) &:= \max\{x \mid x \in \mathbb{R}, \mu_{u^F(a_k)}(x) = ir_{min}\} \\ v_{Left}^{F} ir_{min}(a_k) &:= \min\{x \mid x \in \mathbb{R}, \mu_{v^F(\{a_k\})}(x) = ir_{min}\} \\ t_{ki}^{r_{max} ir_{min}} &:= u_{Right}^{F} ir_{min}(a_k) - v_{Left}^{F} ir_{min}(a_k) \end{aligned}$$

The definition implies that  $ir_{min} \geq st_{min}$ , otherwise convergence towards a configuration which holds to both requirements cannot be guaranteed.

The termination criterion for the transfer scheme is given by means of  $st_{min}$ .

If  $(s_{ik}^F \succ_{PD} s_{ki}^F) = (s_{ki}^F \succ_{PD} s_{ik}^F)$ , the transfer scheme yields 0 as upper bound for both  $t_{ki}^{r_{max}}$  and  $t_{ik}^{r_{max}}$ . But then  $(s_{ki}^F \approx_{PS} s_{ik}^F) = 1.0$ , and thus the membership of the respective configuration in the kernel is also 1.0.

## 7. COALITION FORMATION

To show how the previously defined concepts can be used as a basis for coalition formation in games with fuzzy payoffs, we adopt the the Polynomial Kernel-Oriented Coalition Algorithm (KCA) which was introduced in [6], adjusting it

to the fuzzy Kernel. This adoption will be simplified in such a way that only the coalition values are considered in determining the most profitable coalitions and that we do not take distributed computation into account. However, it can easily be transformed into a distributed version. We bound the coalition size in order to obtain polynomial execution time.

The algorithm is a bilateral coalition formation algorithm, i.e. new coalitions are formed by merging two previously existing ones. In each iteration, a configuration which is a fuzzy solution to the game is formed and contains at most one new coalition. This is because the merging of two coalitions can affect the surpluses, and thus the payoffs, of agents in other coalitions. The configuration is calculated using the transfer scheme given in definition 19 and thus the algorithm is restricted to coalition values which are fuzzy numbers and the use of  $PD$  and  $PS$ . The algorithm consists of three main parts:

1. *Configuration proposal generation:* each coalition computes possible fuzzy kernel-stable (resp. to  $st_{min}$ ) configurations for coalition structures in which the active and another coalition are merged. If such a configuration is preferable to the current configuration by the active coalition, it is added to the other coalition's proposal set.
2. (a) *Proposal evaluation:* After the first part is completed, each coalition has a set of proposals from other coalitions and now evaluates which of these proposals are preferable for it, again compared to the current configuration. If a proposal is not preferable, it is deleted from the set.  
 (b) *Proposal acceptance:* Of the remaining proposals, each coalition accepts one with a maximum gain in benefits in the proposed configuration (by means of  $PD$ ).
3. *Configuration selection:* Finally, an accepted configuration is chosen to become the next configuration. If there are bilateral accepted coalition structures, i.e. structures in which coalitions  $C_1$  and  $C_2$  are merged and both of them accepted the respective proposals, only their corresponding configuration proposals remain in the set to be chosen from. A configuration with a maximal gain in benefits for the agents in the new coalition is then selected. If there are no accepted proposals, the algorithm terminates.

*Definition 20.* The Fuzzy Polynomial Kernel-Oriented Coalition Algorithm (KCA-F) is given by the following pseudocode in the context of the fuzzy game  $(A, v^F)$  with

- constants  $SMin, IMin \geq SMin, EMin$ : real; the minimum requirements on the degrees of fuzzy stability, individual rationality and effectiveness, respectively, for a configuration to be a fuzzy solution. The restriction  $IMin \geq SMin$  is required due to the definition of the transfer scheme.
- constant  $CSizeMax$ : integer; the maximum allowed coalition size (in number of agents).
- operator  $PrefROP$ ; the fuzzy quantity ranking operator used to evaluate the preference of a coalition to

merge another. It may be different to that one used for the evaluation of the fuzzy kernel.

- constant  $CfgPrefThreshold$ : real; the minimum degree of preference of a coalition to merge another coalition.
- function  $Size(C)$ : integer; returns the number of agents in coalition  $C$ .

- function  $Pref((C, u^F), (C^*, u^{F*}), C \in \mathcal{C})$ : boolean; returns TRUE iff the configuration  $(C^*, u^{F*})$  is preferred by coalition  $C$  to the configuration  $(C, u^F)$ . That is iff

$$\min_{a \in C} (u^{F*}(a) \succsim_{PrefROp} u^F(a)) \geq CfgPrefThreshold$$

- function  $EvalCfg(C, u^F)$ : boolean; returns TRUE iff  $(C, u^F)$  is a  $(IMin, EMin, SMin, PD, PS, K^{F^{PD, PS}})$ -solution where for the evaluation of  $K^{F^{PD, PS}}$ , only coalitions  $C$  with  $Size(C) \leq CSizeMax$  are considered.

- function  $ComputeCfg((C, u^F), C_1 \in \mathcal{C}, C_2 \in \mathcal{C})$ : fuzzy configuration; returns a configuration  $(C^*, u^{F*})$  with  $C^* = \{C \mid C = C_1 \cup C_2 \text{ or } C \in \mathcal{C}, C \neq C_1, C \neq C_2\}$ . If possible, for the returned configuration

$$EvalCfg(C^*, u^{F*}) = \text{TRUE}$$

holds. The payoff distribution is computed using the transfer scheme given in definition 19.

- function  $Gain((C, u^F), (C^*, u^{F*}), C \in \mathcal{C})$ : fuzzy number; returns the summarized gain in benefits of the agents in coalition  $C$  in the configuration  $(C^*, u^{F*})$  with respect to the configuration  $(C, u^F)$ , i.e.

$$\sum_{a \in C} (u^{F*}(a) - u^F(a))$$

- function  $Best((C, u^F), CfgGainSet = \{((C_1, u_1^F), Gain_1), \dots, ((C_n, u_n^F), Gain_n)\})$ : (fuzzy configuration, fuzzy number); returns a tuple of a configuration and the corresponding gain  $((C_i, u_i^F), Gain_i)$  with  $((C_i, u_i^F), Gain_i) \in CfgGainSet, 1 \leq i \leq n$  such that with  $GSet := \{Gain_1, \dots, Gain_n\}$ ,  $\mu_{\max(GSet)}((Gain_i)) = \max_{g \in GSet} \{\mu_{\max(GSet)}(g)\}$

Further, let  $n = |A|$ ,  $a_i \in A, 1 \leq i \leq n$  and  $u_{init}^F$  a fuzzy payoff distribution with  $u_0^F(a_i) := v^F(a_i), 1 \leq i \leq n$ .

```

rnum := 0; C0 := {{a1}, ..., {an}}; u0^F := u_init^F
repeat
  rnum := rnum + 1; Crnum := Crnum-1; urnum^F := urnum-1^F
  for all Ci in Crnum do
    PropSetCi := ∅;
  end for
  for all Ci in Crnum do
    for all Ck ≠ Ci in Crnum, Size(Ci ∪ Ck) ≤ CSizeMax do
      PropCfg := ComputeCfg((Crnum, urnum^F), Ci, Ck);
      if EvalCfg(PropCfg) and Pref(Crnum, Prop, Ci) then
        CGain := Gain(Crnum, PropCfg, Ci);
        PropSetCk := PropSetCk ∪ {(PropCfg, CGain)};
      end if
    end for
  end for
  UAProps := ∅; BAProps := ∅;
  for all Ci in Crnum do
    for all (PropCfg, OCGain) ∈ PropSetCi do
      PropSetCi := PropSetCi \ {(PropCfg, OCGain)};

```

```

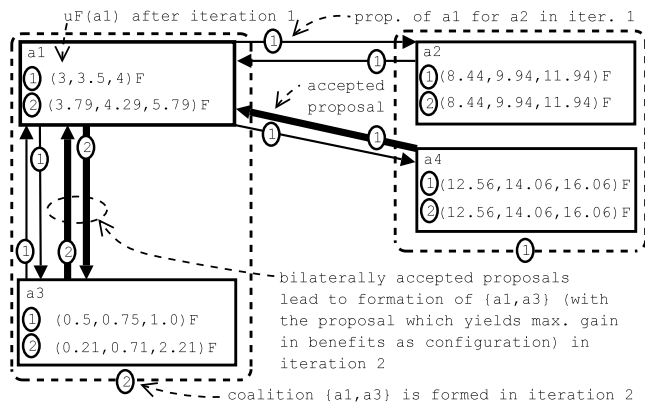
  if Pref(Crnum, PropCfg, Ci) then
    GainSum := OCGain ⊕ Gain(Crnum, PropCfg, Ci);
    PropSetCi := PropSetCi ∪ {(PropCfg, GainSum)};
  end if
end for
((Ca, ua^F), Gaina) := Best(Crnum, PropSetCi);
if ∃((Cpa, upa^F), Gainpa) ∈ UAProps : Cpa = Ca then
  UAProps := UAProps \ {(Cpa, upa^F), Gainpa};
  BAProps := BAProps ∪ {(Cpa, upa^F), Gainpa};
  BAProps := BAProps ∪ {(Ca, ua^F), Gaina};
else
  UAProps := UAProps ∪ {(Ca, ua^F), Gaina};
end if
end for
if BAProps ≠ ∅ then
  ((CBA, uBA), GainBA) := Best((Crnum, urnum^F), BAProps);
  Crnum := CBA; urnum^F := uBA;
else if UAProps ≠ ∅ then
  ((CUA, uUA), GainUA) := Best((Crnum, urnum^F), UAProps);
  Crnum := CUA; urnum^F := uUA;
end if
until Crnum = Crnum-1

```

A remark on complexity: let  $n_a$  denote the number of agents. In a configuration, there exist at most  $n_a$  coalitions. So for the first part of the KCA-F, the number of computed configurations is thus bounded by  $n_a^2$ . Let  $n_{cs} = 2^{CSizeMax}$  ( $CSizeMax$  is assumed to be independent from  $n_a$ ). For the transfer scheme, the complexity for computing a fuzzy surplus was given in section 5, wherein  $2^{n_{agents}}$  is to be replaced by  $n_{cs}$ :  $Comp_{surplus}^{cs} = O(n_{cs} \cdot n_a^3 \cdot n_{fuzzy}^{max^2})$ , with  $n_{fuzzy}^{max}$  as in lemma 1. All further calculations in a transfer step are less complex than this. So the complexity of a transfer step is bounded by  $Comp_{surplus}^{cs}$ . The termination criterion for the transfer scheme is given by means of  $SMin$ , which, together with  $PD$  and  $PS$ , plays a similar role as an *allowed error* in the crisp scheme. The crisp theme terminates within  $n_a \log(e)$  iteration steps (see [11]), where  $e$  is the quotient of the initial and the allowed error of the configuration of the first step. This  $n_a$ -independence of the logarithm is maintained in the fuzzy version. Thus  $O(n_a)$  transfer steps are made per agent for at most  $n_a$  agents. The overall complexity for part one is thus  $O(n_a^2 \cdot n_a^2 \cdot Comp_{surplus}^{cs}) = O(n_{cs} \cdot n_a^7 \cdot n_{fuzzy}^{max^2})$ .

Similarly, the complexity of  $Eval(Cfg)$  is  $O(n_{cs} \cdot n_a^5 \cdot n_{fuzzy}^{max^2})$  (see section 5). All other operations in the algorithm are of less complexity, thus the complexity of the algorithm is given by  $O(n_{cs} \cdot n_a^7 \cdot n_{fuzzy}^{max^2})$ .

*Example 5.* For the game given in example 1 and with  $SMin = 0.9$ ,  $IMin = 0.95$ ,  $EMin = 1.0$ ,  $CSizeMax = 4$ ,  $CfgPrefThreshold = 0.9$  and  $PrefROp = PD$ , the algorithm might proceed as follows (we say "might" because the transfer scheme inherits some nondeterminism due to the tolerances that are given by the choices of  $SMin$  and  $IMin$ ). In the first iteration of the repeat loop each agent makes a proposal to every other agent.  $\{a_2\}$  and  $\{a_4\}$  are chosen to merge because they get the maximum gain in benefits and accept the respective proposals bilaterally. In the second iteration,  $\{a_1\}$  and  $\{a_3\}$  propose a merge mutually. They are not able to compute any beneficial proposals for a merge with  $\{a_2, a_4\}$  (and vice-versa) because the value of all three- and four-agent coalitions is  $0^F$ . Thus,  $\{a_1\}$  and  $\{a_3\}$  bilaterally accept their proposals and merge. The complete procedure from the perspective of  $a_1$  together with the resulting configurations is illustrated in figure 2. As we can see there,  $a_3$  "risks" his individual rationality ( $\mu_{indrat}^{PD}(a_3) \approx 0.98$ ) and thus his increase in benefits.



**Figure 2: KCA-F coalition formation from the perspective of  $a_1$**

This is possible because the respective minimum requirements were set 0.95 and 0.9. This can be interpreted as an optimistic decision, because there still is a possibility that  $a_3$  is better off with this merge than staying alone. All other individual rationalities are 1.0. Further, we have  $(s_{13}^{F^{PD}} \approx_{PS} s_{31}^{F^{PD}}) \approx 0.98$  and  $(s_{24}^{F^{PD}} \approx_{PS} s_{42}^{F^{PD}}) \approx 0.96$ . Thus, the final configuration (in the third iteration, no proposals at all are made, and the algorithm terminates) is a  $(0.98, 1.0, 0.96, PD, PS, K^{F^{PD, PS}})$ -solution. The payoffs resemble the agents' abilities quite intuitively. Although  $a_4$  has slightly less game-relevant information available than  $a_2$ ,  $a_4$ 's threat to coalize with  $a_1$  instead is considerably larger than  $a_2$ 's.  $a_1$  and  $a_3$  clearly have much less game-relevant information, thus their coalition value and consequently their payoffs are much smaller. However, both  $a_1$ 's game-relevant information and coalitional possibilities are much more valuable than those of  $a_3$ , which explains the payoff-distribution in favor of  $a_1$  very well.

## 8. CONCLUSION

In the setting of fuzzy-valued cooperative game theory, we fuzzified some game-theoretic concepts such as configurations, individual rationality and efficiency to introduce the concept of fuzzy kernel-stable coalitions. With these definitions, it is possible to specify cooperative games involving uncertain information and to find good candidates of fuzzy configurations to be a solution for the game by evaluating their membership value in the fuzzy kernel. It has been pointed out that the choice of a fuzzy ranking method is an essential aspect of this procedure. A transfer scheme to calculate fuzzy kernel-stable configurations, and a coalition formation algorithm, the KCA-F, have been provided. The procedure of coalition formation of the KCA-F was illustrated with the help of an explanatory example. The complexities of an evaluation of a configuration's membership value, the calculation of fuzzy kernel-stable configurations and the KCA-F as a whole have been shown to be exponential, but could be reduced to polynomial time by limiting the size of considered coalitions. Because a precondition of our approach was that all agents in a game share the same understanding about the fuzziness of the data, further work includes the identification of suitable methods to compute

the game-wide accepted membership functions from agent-specific beliefs. Also, proper defuzzification methods for the fuzzy payoff-distribution, which will be needed once the coalitions performed their tasks and got their actual (crisp) payoffs, will have to be found. To sum it up more generally, exact specifications of environments for applications of fuzzy valued cooperative games need to be developed.

## 9. REFERENCES

- [1] G. Bortolan and R. Degani. A review of some methods of ranking fuzzy subsets. *Fuzzy Sets and Systems*, 15(1):1–19, 1985.
- [2] D. Butnariu and E. P. Klement. *Triangular Norm-Based Measures and Games with Fuzzy Coalitions*. Kluwer, Dordrecht, 1993.
- [3] M. Davis and M. Maschler. The kernel of a cooperative game. *Naval Research Logistics Quarterly*, 12:223–259, 1965.
- [4] D. Dubois and H. Prade. Ranking fuzzy numbers in the setting of possibility theory. *Information Sciences*, 30:183–224, 1983.
- [5] D. Dubois and H. Prade. Fuzzy numbers: an overview. In J. C. Bezdek, editor, *Analysis of fuzzy information*, volume I Mathematics and Logic, pages 3–39. CRC Press, Boca Raton, Florida, 1994.
- [6] M. Klusch and O. Shehory. A polynomial kernel-oriented coalition algorithm for rational information agents. In *Proc. 2. International Conference on Multi-Agent Systems ICMAS-96*. AAAI Press, 1996.
- [7] M. Mareš. *Fuzzy cooperative games : cooperation with vague expectations*, volume 72 of *Studies in fuzziness and soft computing*. Physica Verlag, Heidelberg ; New York, 2001.
- [8] T. Sandholm, K. Larson, M. Andersson, O. Shehory, and F. Tohme. Coalition structure generation with worst case guarantees. *Artificial Intelligence Journal*, 111 (1-2):209–238, 1999.
- [9] O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence Journal*, 101 (1-2):165–200, May 1998.
- [10] O. Shehory and S. Kraus. Feasible formation of coalitions among autonomous agents. *Computational Intelligence*, 15(3):218–251, 1999.
- [11] R. E. Stearns. Convergent transfer schemes for n-person games. *Transactions of the American Mathematical Society*, 134:449–459, 1968.
- [12] R. R. Yager, S. Ovchinnikov, R. Tong, and H. Nguyen. *Fuzzy Sets and Applications: Selected Papers by L.A.Zadeh*. John Wiley & Sons, New York, 1987.
- [13] L. A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

# BSCA-F: Efficient Fuzzy Valued Stable Coalition Forming Among Agents

Bastian Blankenburg and Matthias Klusch

DFKI - German Research Center for Artificial Intelligence

Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany

{blankenb,klusch}@dfki.de

## Abstract

*We propose a new low-complexity coalition forming algorithm, BSCA-F, that enables agents to negotiate bilateral Shapley value stable coalitions in uncertain environments, and demonstrate its usefulness by example. In particular, we show that utilizing the possibilistic mean value for defuzzifying negotiated fuzzy agent payoffs is reasonable, and fuzzy ranking methods can be utilized to implement optimistic, or pessimistic strategies of individual agents.*

## 1 Introduction

Game-theoretic coalition algorithms can be used by intelligent agents as coordination means in a variety of applications in different environments. Classic approaches such as in [12, 8] proposed solutions to the problem of how self-interested agents best form stable coalitions in the sense of cooperative game theory. However, negotiation during the coalition-forming process might be uncertain. Such uncertainties could be caused by the possibility of non-deterministic events that hamper the negotiation process and produce incomplete information. Agents might have uncertain knowledge about the share of coalition income in which they intend to participate or on the degree of their membership in one or multiple coalitions. An agent might determine the degree of its membership to potential coalitions by individually leveled commitments to other agents or bargains that indicate the degree of collaboration that the agents desire. The first case might imply the formation of fuzzy-valued coalitions, whereas the second case might induce the formation of fuzzy coalitions, which might partially overlap. In this paper, we focus on negotiations of game-theoretically stable fuzzy-valued coalitions.

In [5], agents learn about each other in a way that allows for uncertain environmental knowledge and different expectations of coalition values by different agents. However, coalition stability is based on the exponential Bayesian Core which may be empty in certain cases. [9] present a heuris-

tic approach that avoids computing stable payoffs in the sense of cooperative game theory at all. Based on the work of [10], in [1] a possibilistic fuzzy extension of the Kernel is defined that allows agents to negotiate fuzzy Kernel stable coalitions with low polynomial complexity. However, the reduction in computational complexity strictly depends on the maximum size of coalitions allowed.

In this paper, we propose an alternative algorithm, BSCA-F, that allow agents to negotiate fuzzy-valued coalitions in the setting of possibility theory. The BSCA-F does not require to constrain coalition sizes for negotiations of low computational and communication complexity. To achieve this, we utilize the fuzzy bilateral Shapley value which, however, implies that, in general, only subgame-stability can be achieved. We show that it is reasonable to utilize the possibilistic mean value [3] for defuzzifying the negotiated fuzzy payoffs to implement unambiguous coalition contracts among the agents.

The remainder of this paper is structured as follows.

## 2 Background

We extend game-theoretic concepts of coalition theory by means of possibilistic interpretation of fuzzy coalition values [10], and fuzzy ranking methods. Possibility theory [13, 7] evolved from a possibilistic interpretation of fuzzy set theory. There is empirical evidence for that people perform rather possibilistic than probabilistic reasoning though their subjective assessment of the probability and possibility of real world events closely coincide [11]. Possibilistic interpretation of any fuzzy quantity indicates the degree of its possibility but not the probabilistic degree of its truth. As a consequence, by modeling uncertainty in terms of fuzzy quantities negotiating agents may ignore certain situations they either do not understand, or are simply not interested in, rather than being enforced to assign individual probability values to each of them.

## 2.1 Fuzzy Quantities

In the following, we define fuzzy quantities, operations, and ranking methods that are needed to understand the notion fuzzy-valued coalition game we introduce in subsequent sections. For more details, we refer the reader to, for example, [10, 1].

**Definition 2.1** A fuzzy subset  $\tilde{s}$  of a set  $S$  is defined by its membership function  $\mu_{\tilde{s}} : S \mapsto [0, 1]$  where  $\mu_{\tilde{s}}(x)$ ,  $x \in S$  is called the degree of membership or membership value of  $x$  in  $S$ .  $x \in \tilde{s}$  iff  $\mu_{\tilde{s}}(x) > 0$ , with  $\text{support}(\tilde{s}) := \{x \mid x \in \tilde{s}\}$ .  $\mu_{\tilde{s}}(x)$  is also called possibility distribution for  $X$  ( $\Pi(X = x)$ ), if it denotes the degree of possibility that variable  $X$  of domain  $S$  takes the value  $x$ . Any fuzzy subset of  $\mathbb{R}$  is called a fuzzy quantity.

**Definition 2.2** Let  $\tilde{x}$  a fuzzy quantity;  $\tilde{\mathbb{R}}$  the set of all fuzzy quantities.

1.  $\text{size}(\tilde{x}) := \sup\{\text{support}(\tilde{x})\} - \inf\{\text{support}(\tilde{x})\}$
2.  $\tilde{x}$  is normalized iff  $\sup_{x \in \mathbb{R}}\{\mu_{\tilde{x}}(x)\} = 1$
3.  $x \in \mathbb{R}$  with  $\mu_{\tilde{x}}(x) = \max_{y \in \mathbb{R}}(\mu_{\tilde{x}}(y))$  is a modal value of  $\tilde{x}$ .
4.  $\tilde{r}, r \in \mathbb{R}$  denotes a fuzzy quantity with

$$\mu_{\tilde{r}}(x) = \begin{cases} 1 & \text{if } x = r \\ 0 & \text{otherwise} \end{cases}, x \in \mathbb{R}$$

5. A fuzzy interval  $\tilde{I}$  is a fuzzy quantity with  $\forall x_1, x_2, x_3 \in \mathbb{R}, x_1 < x_2 < x_3 : \mu_{\tilde{I}}(x_2) \geq \min(\mu_{\tilde{I}}(x_1), \mu_{\tilde{I}}(x_3))$
6. A trapezoid fuzzy interval  $((x_1, \widehat{(x_2, x_3)}, x_4))$ ,  $x_1, x_2, x_3, x_4 \in \mathbb{R}$  is a fuzzy interval with  $\forall r \in \mathbb{R}$ :

$$\mu_{((x_1, \widehat{(x_2, x_3)}, x_4))}(r) = \begin{cases} 1 & \text{if } x_2 \leq r \leq x_3 \\ \frac{r-x_1}{x_2-x_1} & \text{if } x_1 < r < x_2 \\ \frac{x_4-r}{x_4-x_3} & \text{if } x_3 < r < x_4 \\ 0 & \text{otherwise} \end{cases}$$

Arithmetic operations on fuzzy quantities follow Zadeh's extension principle.

**Definition 2.3** Let  $\tilde{x} \in \tilde{\mathbb{R}}^n, n \in \mathbb{N}$ . The function  $\tilde{f} : \tilde{\mathbb{R}}^n \mapsto \tilde{\mathbb{R}}$  is called a fuzzy extension of a function  $f : \mathbb{R}^n \mapsto \mathbb{R}$  iff  $\forall x \in \mathbb{R}^n : \mu_{\tilde{f}(\tilde{x})}(x) = \sup_{y \in \mathbb{R}^n} \{\min_{1 \leq i \leq n} \{\mu_{\tilde{x}_i}(y_i)\} \mid f(y) = x\}$  if  $f^{-1}(x) \neq \emptyset$ , and  $\mu_{\tilde{f}(\tilde{x})}(x) = 0$  if  $f^{-1}(x) = \emptyset$ .

**Definition 2.4** Let  $F_1, F_2 \in \mathbb{R}^F, x, y, z, a \in \mathbb{R}$ . Applying the extension principle, we define

$$\begin{aligned} \mu_{F_1 \oplus F_2}(x) &:= \sup\{\min(\mu_{F_1}(y), \mu_{F_2}(z)) \mid y + z = x\} \\ \mu_{-F_1}(x) &:= \mu_{F_1}(-x) \\ \mu_{F_1 \ominus F_2}(x) &:= \mu_{F_1 \oplus (-F_2)}(x) \\ \mu_{a \cdot F_1}(x) &:= \begin{cases} \mu_{F_1}(x/a) & \text{if } a \neq 0 \\ \mu_{\tilde{0}} & \text{if } a = 0 \end{cases} \end{aligned}$$

Agents are supposed to negotiate coalitions with expected fuzzy gains, hence to compute, compare, and select fuzzy utility values. That, in particular, requires means of ranking fuzzy quantities several of which being proposed for different applications such as in [2]. We adopt fuzzy quantity ranking operators that have been introduced by Dubois and Prade in the setting of possibility theory [6].

**Definition 2.5** Let  $F_1, F_2 \in \mathbb{R}^F, R$  a fuzzy subset of  $\tilde{\mathbb{R}} \times \tilde{\mathbb{R}}$ .  $R$  is a fuzzy ranking operator, or fuzzy similarity relation, if  $\mu_R(F_1, F_2)$  denotes the degree to which  $F_1$  is "greater" or "similar" than  $F_2$ , respectively. Further, let  $G$  a fuzzy ranking operator and  $S$  a fuzzy similarity relation. We define  $(F_1 \tilde{\succ}_G F_2) := \mu_G(F_1, F_2)$  and  $(F_1 \tilde{\approx}_S F_2) := \mu_S(F_1, F_2)$ . Regarding the possibility distributions  $F_1, F_2 \in \tilde{\mathbb{R}}$  of  $f_1$  and  $f_2$ , respectively, [6] define the

1. possibility of dominance  $\tilde{\succ}_P$  of  $f_1$  over  $f_2$  as  $\Pi(f_1 \geq f_2) = F_1 \tilde{\succ}_P F_2 = \sup\{\min(\mu_{F_1}(x), \mu_{F_2}(y)) \mid x, y \in \mathbb{R}, x \geq y\}$ ;
2. necessity of dominance  $\tilde{\succ}_N$  of  $f_1$  over  $f_2$  as  $N(f_1 \geq f_2) = F_1 \tilde{\succ}_N F_2 = \inf_x \{\sup_y \{\max(1 - \mu_{F_1}(x), \mu_{F_2}(y)) \mid x, y \in \mathbb{R}, x \geq y\}\}$ ;
3. possibility of strict dominance  $\tilde{\succ}_P$  of  $f_1$  over  $f_2$  as  $\Pi(f_1 > f_2) = F_1 \tilde{\succ}_P F_2 = \sup_x \{\inf_y \{\min(\mu_{F_1}(x), 1 - \mu_{F_2}(y)) \mid x, y \in \mathbb{R}, x \leq y\}\}$ ;
4. necessity of strict dominance  $\tilde{\succ}_N$  of  $f_1$  over  $f_2$  as  $N(f_1 > f_2) = F_1 \tilde{\succ}_N F_2 = \inf \{\max(1 - \mu_{F_1}(x), 1 - \mu_{F_2}(y)) \mid x, y \in \mathbb{R}, x \leq y\}$ ;
5. possibility of equality  $\tilde{\approx}_P$  of  $f_1$  and  $f_2$  as  $\Pi(f_1 = f_2) = F_1 \tilde{\approx}_P F_2 = \min((F_1 \tilde{\succ}_P F_2), (F_2 \tilde{\succ}_P F_1))$ ;
6. necessity of equality  $\tilde{\approx}_N$  of  $f_1$  and  $f_2$  as  $N(f_1 = f_2) = F_1 \tilde{\approx}_N F_2 = \min(\min(N(F_2 \geq F_1), 1 - \Pi(F_2 > F_1)), \min(N(F_1 \geq F_2), 1 - \Pi(F_1 > F_2)))$

A fuzzy set of maximal elements of a set  $X$  of fuzzy quantities  $X$ , and fuzzy logical operators "AND" and "OR" with operands in  $[0, 1]$  are defined as follows.

**Definition 2.6** Let  $X$  a set of fuzzy quantities,  $G$  a fuzzy ranking operator,  $x, y \in [0, 1], n \in \mathbb{N}$ .

1. The fuzzy subset  $\widetilde{\max}^G X$  of  $X$  is given by  $\mu_{\widetilde{\max}^G X}(F_1) := \min_{F_2 \in X, F_2 \neq F_1} (F_1 \widetilde{\succeq}_G F_2)$ ,  $F_1 \in X$  denoting the degree to which  $F_1$  is a maximal element of  $X$ .
2. The (crisp) set  $\max^G X$  of maximal elements of  $X$  is defined as  $\max^G X := \{F \mid \mu_{\widetilde{\max}^G X}(F) = \max \mu_{\widetilde{\max}^G X}\}$
3.  $x \widetilde{\wedge} y := \min(\widetilde{x}, \widetilde{y})$ ,  $x \widetilde{\vee} y := \max(\widetilde{x}, \widetilde{y})$ .  
For  $\widetilde{\odot} \in \{\widetilde{\wedge}, \widetilde{\vee}\}$ :  $\widetilde{\odot} \{\widetilde{x}_1, \dots, \widetilde{x}_n\} := (\widetilde{x} \widetilde{\odot} (\widetilde{x}_2 \dots (\widetilde{x}_{n-1} \widetilde{\odot} \widetilde{x}_n) \dots))$

**Definition 2.7** For  $\widetilde{x} \in \widetilde{\mathbb{R}}$ ,  $L_\alpha(\widetilde{x}) := \{x \mid x \in \mathbb{R}, \mu_F(x) \geq \alpha\}$  with  $\alpha \in [0, 1]$  is called an  $\alpha$ -level cut of  $\widetilde{x}$ . We also define  $L_\alpha(\widetilde{x}_*) := \inf\{L_\alpha(\widetilde{x})\}$  and  $L_\alpha(\widetilde{x}^*) := \sup\{L_\alpha(\widetilde{x})\}$

**Definition 2.8** Given a fuzzy interval  $\widetilde{x} \in \widetilde{\mathbb{R}}$ , with  $\mu_{\widetilde{x}}$  representing a possibility distribution for a variable  $X \in \mathbb{R}$ ,

$$E(X) := \int_0^1 \alpha(L_\alpha(\widetilde{x})_* + L_\alpha(\widetilde{x})^*) d\alpha$$

is called the possibilistic mean value of  $X$  [3]. Instead of  $E(X)$ , we also write  $e(\widetilde{x})$ .

The additive possibilistic mean value  $e$  is similar to the expected value of stochastic variables used in probability theory, though there is no common agreement on the semantics of exact degrees of possibility <sup>1</sup>. Since  $e$  maps fuzzy membership functions to crisp real values, we consider it as an appropriate method for defuzzification of fuzzy quantities in the setting of possibility theory.

**Remark 2.9** For any trapezoid fuzzy interval  $\widetilde{I} = ((x_1, x_2, x_3, x_4))$ ,  $x_1, x_2, x_3, x_4 \in \mathbb{R}$ ,

$$e(\widetilde{I}) = \frac{x_1 + x_2 + x_3 + x_4}{4} = \frac{\frac{x_2+x_3}{2} + \frac{x_1+x_4}{2}}{2}$$

This form makes clear that for trapezoid fuzzy intervals,  $e$  is the real value in  $\widetilde{I}$  minimizing the average distance to the bounds of the most possible values  $[x_2, x_3]$  of  $\widetilde{I}$  and the bounds of the support  $(x_1, x_4)$ , i.e. the values that are possible at all. In this sense,  $e$  can also be considered as a possibilistic error minimizing defuzzification method.

## 2.2 Fuzzy Coalition Games

**Definition 2.10** A fuzzy coalition game  $(\mathcal{A}, \widetilde{v})$  consists of a set of agents  $\mathcal{A}$ , a fuzzy characteristic function  $\widetilde{v} : 2^{\mathcal{A}} \mapsto \widetilde{\mathbb{R}}$ , and the membership function of the fuzzy quantities  $\widetilde{v}(C)$  that might be interpreted as expectation of the common coalition profit that is to be distributed among its members.

<sup>1</sup>Carlsson and Fuller introduced a weighted version [4] of  $e$  which allows for adjusting the importance of different possibility levels.

The worth  $\widetilde{v}(C)$  of a fuzzy-valued coalition  $C$  is a fuzzy set of its possible real-valued coalitional profits, represents a possibility distribution of the real coalition value  $v(C) \in \mathbb{R}$ , and has at least one modal value. If, for a given fuzzy coalition game, the coalition value  $v(C)$  is equal to one modal value of  $C$  for all possible coalitions  $C$ , it is equivalent to a (deterministic) coalition game <sup>2</sup>.

**Definition 2.11** A fuzzy configuration  $(\mathcal{C}, \widetilde{u})$  consists of a (crisp) coalition structure  $\mathcal{C}$  and fuzzy payoff distribution  $\widetilde{u} : \mathcal{A} \mapsto \widetilde{\mathbb{R}}$ .  $\widetilde{u}$  is called  $\widetilde{=}$ -efficient to a degree of

$$\mu_{eff}(\widetilde{u}) := \widetilde{\bigwedge}_{C \in \mathcal{C}} \left\{ \sum_{a_i \in C} \widetilde{u}(a_i) \widetilde{=} \widetilde{v}(C) \right\}$$

with fuzzy similarity relation  $\widetilde{=}$ . For  $a \in \mathcal{A}$  and fuzzy ranking operator  $\widetilde{\succeq}$ , the degree of individual  $\widetilde{\succeq}$ -rationality ( $\mu_{indrat}^{\widetilde{\succeq}}(a)$ ), and overall  $\widetilde{\succeq}$ -rationality ( $\mu_{indrat}^{\widetilde{\succeq}}(\widetilde{u})$ ) of the payoff distribution  $\widetilde{u}$  is defined as  $\widetilde{u}(a) \widetilde{\succeq} \widetilde{v}(a)$  and  $\widetilde{\bigwedge}_{a \in \mathcal{A}} \{\mu_{indrat}^{\widetilde{\succeq}}(a)\}$ , respectively.

The fuzzy Shapley value stable payoff distribution, introduced in [10], is  $\widetilde{\succeq}_P$ -rational as well as  $\widetilde{=}_P$ -efficient with degree 1 if the coalition values are normalized fuzzy intervals. For reason of efficient negotiation, we adopt the fuzzified bilateral Shapley value for stable payoff distribution among members of bilaterally formed fuzzy valued coalitions.

**Definition 2.12** The fuzzy Shapley value  $\widetilde{\sigma}(a)$  of agent  $a \in \mathcal{A}$  in a fuzzy game  $(\mathcal{A}, \widetilde{v})$  is  $\widetilde{\sigma}(a) = \sum_{C \subseteq \mathcal{A}} \frac{(|\mathcal{A}| - |C|)! (|C| - 1)!}{|\mathcal{A}|!} (\widetilde{v}(C) - \widetilde{v}(C \setminus \{a\}))$ .

The fuzzy bilateral Shapley value  $\widetilde{\sigma}_b(C_1 \cup C_2, C_i, v)$ ,  $C_i, i \in \{1, 2\}$  of the bilateral coalition  $C_1 \cup C_2$  is defined as the fuzzy Shapley value of  $C_i$  in the game  $(\{C_1, C_2\}, \widetilde{v})$ :

$$\widetilde{\sigma}_b(C_1 \cup C_2, C_i, \widetilde{v}) := \frac{1}{2} \widetilde{v}(C_i) \oplus \frac{1}{2} (\widetilde{v}(C_1 \cup C_2) \ominus \widetilde{v}(C_k))$$

with  $k \in \{1, 2\}, k \neq i$ .

Since the uncertainty denoted by  $\widetilde{v}(C_1)$ ,  $\widetilde{v}(C_2)$  and  $\widetilde{v}(C_1 \cup C_2)$  is now represented by the fuzzy bilateral Shapley value, it requires appropriate defuzzification of the payoff distribution at the end of the coalition negotiations to enable crisp side-payments among coalition members. If, for example, coalition  $C_1 \cup C_2$  has been negotiated, the agents may determine the crisp coalition value  $v(C_1 \cup C_2)$  from the actual costs and rewards after having carried out the agreed joint actions. However, the real coalition values  $v(C_1)$  and  $v(C_2)$ , in general, remain unknown to the agents, hence need to be defuzzified. For this purpose, we use a modified

<sup>2</sup>In the following, we use the term "fuzzy game" instead of "fuzzy coalition game".

fuzzy bilateral Shapley value that uses coalition values of subcoalitions defuzzified by the possibilistic mean, which leaves the fuzziness of the resulting payoffs to that of the joint coalition value only.

**Definition 2.13** *The fuzzy bilateral Shapley value of given fuzzy game  $(\mathcal{A}, \tilde{v})$  with defuzzified values of subcoalitions  $\tilde{\sigma}_b^e(C_1 \cup C_2, C_i, v), C_i, i \in \{1, 2\}$  in the bilateral coalition  $C_1 \cup C_2$  is defined as the fuzzy Shapley value of  $C_i$  in the game  $(\{C_1 \cup C_2, C_1, C_2\}, \tilde{v})$ . With  $k \in \{1, 2\}, k \neq i$ ,*

$$\tilde{\sigma}_b^e(C_i, \tilde{v}) := \frac{1}{2}e(\tilde{v}(C_i)) \oplus \frac{1}{2}(\tilde{v}(C_1 \cup C_2) \ominus e(\tilde{v}(C_k)))$$

Similar to the crisp bilateral Shapley value, we use a recursive payoff distribution of the modified fuzzy bilateral Shapley value based on the recursively bilateral formation of coalition structures. Each agent maintains its individual coalition history tree in due course of the bilateral negotiations it participates in as member of a coalition.

**Definition 2.14** *The fuzzy payoff distribution  $\tilde{u}$  within any bilateral coalition  $C$  in a fuzzy game  $(\mathcal{A}, \tilde{v})$  is called recursively fuzzy bilateral Shapley value stable iff for every non-leaf node  $C^*$  of the coalition history tree  $T_C : u(C_i^*) = \tilde{\sigma}_b^e(C^*, C_i^*, \tilde{v}_{C^*}), i \in 1, 2$  with  $\forall C^{**} \subseteq A$ :*

$$\tilde{v}_{C^*}(C^{**}) = \begin{cases} \tilde{\sigma}_b^e(C^p, C_k^p, \tilde{v}_{C^p}) & \text{if } C^p \in T_C, \\ & C^* = C_k^p, k \in 1, 2 \\ \tilde{v}(C^{**}) & \text{otherwise} \end{cases}$$

For each coalition  $C \subseteq \mathcal{A}$  we define the fuzzy local worth of individual agent  $a \in C$  as  $\widetilde{lworth}_C(C^*) := \sum_{a \in C} \widetilde{lworth}_a(C^*)$  with  $C^* \subseteq \mathcal{A}, C \subseteq C^*$ .  $\widetilde{lworth}_a(C)$  denotes the fuzzy gain of  $a$  for accomplishing its tasks in  $C$  on behalf of its user or other agents in  $C$ , including costs.

Each coalition value is the sum of the local worth of each of its members  $\tilde{v}(C) = \sum_{a \in C} \widetilde{lworth}_a(C)$ . Further, the expected gain in utility of any potential bilateral coalition candidate is the difference between what it may expect to obtain in the coalition merger in terms of the bilateral Shapley value and its expected self-value.

**Definition 2.15** *For a fuzzy game  $(\mathcal{A}, \tilde{v})$ , the bilateral Shapley value based expected utility gain of a subcoalition  $C$  in the coalition  $C \cup C^*, C, C^* \subset A$  is*

$$\tilde{g}_{\tilde{v}}(C, C \cup C^*) := \tilde{\sigma}_b^e(C \cup C^*, C, \tilde{v}) - e(\tilde{v}(C))$$

**Lemma 2.16** *Let a fuzzy game  $(\mathcal{A}, \tilde{v})$  and  $C_1, C_2 \subset \mathcal{A}$ . Then  $\tilde{g}_{\tilde{v}}(C_1, C_1 \cup C_2) = \tilde{g}_{\tilde{v}}(C_2, C_1 \cup C_2)$*

*Proof:* By definitions 2.15 and 2.13, and because of the properties of  $\oplus$  and  $\ominus$  when applied to at least one crisp

operand discussed e.g. in [7], we can rewrite

$$\begin{aligned} & \tilde{g}_{\tilde{v}}(C_1, C_1 \cup C_2) \\ &= \frac{1}{2}e(\tilde{v}(C_1)) \oplus \frac{1}{2}(\tilde{v}(C_1 \cup C_2) \ominus e(\tilde{v}(C_2))) \ominus e(\tilde{v}(C_1)) \\ &= \frac{1}{2}\tilde{v}(C_1 \cup C_2) \ominus \frac{1}{2}e(\tilde{v}(C_2)) \oplus \frac{1}{2}e(\tilde{v}(C_1)) \\ &= \frac{1}{2}\tilde{v}(C_1 \cup C_2) \ominus \frac{1}{2}e(\tilde{v}(C_1)) \oplus \frac{1}{2}e(\tilde{v}(C_2)) \\ &= \tilde{g}_{\tilde{v}}(C_2, C_1 \cup C_2) \end{aligned}$$

□

### 3 Fuzzy-Valued Stable Coalition Negotiation

**Algorithm 3.1 (BSCA-F).**

*Given  $\mathcal{A}$ , initial configuration  $(C_0, \tilde{u}_0)$  with singleton sets in  $C_0$  and  $\tilde{u}_0(a) = \tilde{v}(a)$ , fuzzy ranking operator  $\tilde{o} \in \{\tilde{\succ}_P, \tilde{\succ}_N, \tilde{\succ}_P, \tilde{\succ}_N\}$ , ranking threshold  $t$ , and negotiation round counting variable  $r := 1$ . Further, each coalition determines its representative  $Rep_C$  via voting; representatives are ranked according to given ascending order  $o : \mathcal{A} \mapsto \mathbb{N}$  of agents based on, for example, available computational resources.*

*Each agent  $a \in C \in \mathcal{C}_r$  performs:*

1. *Communication:* If  $a \neq Rep_C$  then go to step 3f; else do for all  $C^* \in \mathcal{C}_r, C^* \neq C$ :

- (a) send  $\widetilde{lworth}_C(C \cup C^*)$  to  $Rep_{C^*}$
- (b) receive  $\widetilde{lworth}_{C^*}(C \cup C^*)$  from  $Rep_{C^*}$
- (c) compute  $\tilde{v}(C \cup C^*) = \widetilde{lworth}_C(C \cup C^*) \oplus \widetilde{lworth}_{C^*}(C \cup C^*)$

2. *Proposal Generation*

- (a)  $Cand_C := \{C^* \mid C \in \mathcal{C} \setminus C, (\tilde{g}(C, C \cup C^*, \tilde{v}) \tilde{o} \tilde{0}) \geq t\}$
- (b) If  $Cand_C \neq \emptyset$ , send proposal to  $Rep_{C^+}$  of most beneficial  $C^+$  to form joint coalition  $C \cup C^+$ . In case of multiple possible choices, uniquely select best representative  $o(Rep_{C \cup C^+}) = \min\{o(Rep_{C^*}) \mid \tilde{g}_{\tilde{v}}(C, C \cup C^+) \in \max^{\tilde{o}}\{\tilde{g}_{\tilde{v}}(C, C \cup C^{**}) \mid C^{**} \in Cand_C\}\}$
- (c) Receive all proposals from all other  $Rep_{C^*}, C^* \in \mathcal{C}_r, C^* \neq C$

3. *Coalition Forming*

- (a) Set  $New := \emptyset$  and  $Obs := \emptyset$
- (b) If a proposal was sent to as well as received from  $C^+$ , form joint coalition  $C \cup C^+$ :

- i. If  $o(Rep_C) < o(Rep_{C^+})$  then  $Rep_{C \cup C^+} := Rep_C$  else  $Rep_{C \cup C^+} := Rep_{C^+}$
  - ii. inform all other  $Rep_{C^*}, C^* \in \mathcal{C}_r, C^* \neq C, C^* \neq C^+$  and all  $a^* \in C, a \neq Rep_C$  about the newly formed coalition and  $Rep_{C \cup C^+}$
  - iii.  $New := \{C \cup C^+\}, Obs := \{C, C^+\}, Cand_C := \emptyset$
- (c) Receive all messages about new coalition. For each new coalition  $C_1 \cup C_2$  and  $Rep_{C_1 \cup C_2}$  do:  $Cand_C := Cand_C \setminus \{C_1, C_2\}, New := New \cup \{C_1 \cup C_2\}$  and  $Obs := Obs \cup \{C_1, C_2\}$ .
  - (d) If no new coalition was formed, go to step 2b.
  - (e) Send the sets  $New$  and  $Obs$  to all other coalition members  $a^* \in C, a \neq Rep_C$
  - (f) If  $a \neq Rep_C$  then receive sets  $New$  and  $Obs$  from  $Rep_C$ .
  - (g) Set  $r := r + 1, \mathcal{C}_r := (\mathcal{C}_{r-1} \setminus Obs) \cup New$ , and  $u_r$  according to recursive fuzzy bilateral Shapley value based on the coalition structures  $\mathcal{C}_r \dots \mathcal{C}_0$ .
  - (h) If  $\mathcal{C}_r = \mathcal{C}_{r-1}$  then stop; else go to step 1

**Proposition 3.2** *Between step 2.b and 3.c in any round  $r \in \mathbb{N}$ , the coalition  $C_1 \cup C_2, C_1, C_2 \in \mathcal{C}_r$ , which is among the overall most profitable coalitions in the sense that  $\tilde{g}_{\bar{v}}(C_1, C_1 \cup C_2) \in \max^{\bar{v}} \{\tilde{g}_{\bar{v}}(C, C \cup C^{**}) \mid C \in \mathcal{C}_r, C^{**} \in Cand_C\}$ , and  $o(Rep_{C_1 \cup C_2})$  is minimal as compared to  $o$  of other overall most profitable coalitions, is formed, or no proposals are sent at all.*

*Proof:* Because of lemma 2.16, we have that if  $C_1 \cup C_2$  is in the set  $Cand_{C_1}$ , it is also in the set  $Cand_{C_2}$ . From the properties of  $\tilde{\geq}_P, \tilde{\geq}_N, \tilde{>}_P$  and  $\tilde{>}_N$  discussed in [6], it is clear that for a set of fuzzy quantities  $X$ , if  $F_1 \in X: F_1 \in \max^G X$ , then also  $F_1 \in \max^G Y \subseteq X$  with  $F_1 \in Y$ . Further,  $\tilde{g}_{\bar{v}}(C_1, C_1 \cup C_2) = \tilde{g}_{\bar{v}}(C_2, C_1 \cup C_2)$  because of lemma 2.16. Thus, with (a) it follows that  $\tilde{g}_{\bar{v}}(C_1, C_1 \cup C_2) \in \max^{\bar{v}} \{\tilde{g}_{\bar{v}}(C_i, C_i \cup C^{**}) \mid C^{**} \in Cand_{C_i}\}$  for both  $i = 1$  and  $i = 2$ . With the unambiguousness of the agent ordering  $o$  and (b), it is then clear that in step 2.b  $C_1$  and  $C_2$  send proposals to each other and thus form  $C_1 \cup C_2$  in step 3.c.  $\square$

**Lemma 3.3** *In round  $r \in \mathbb{N}$ , the iteration between step 2.b and 3.d is done at most  $\frac{|\mathcal{C}_r|}{2}$  times by each agent.*

*Proof:* Assume there have been  $k \in \mathbb{N}$  iterations in a given round of the BSCA-F and  $l \in \mathbb{N}$  new coalitions have been formed. Then proposition 3.2 implies that  $k \leq l$ . Step 3.b.iii implies that every coalition can merge with another

one only once in each round of the BSCA-F, and thus limits the overall number of new coalitions per round to at most  $\frac{|\mathcal{C}_r|}{2}$ . So we have  $k \leq l \leq \frac{|\mathcal{C}_r|}{2}$ .  $\square$

**Lemma 3.4** *The BSCA-F terminates after at most  $|\mathcal{A}|$  rounds.*

*Proof:* In each non-final round  $r \in \mathbb{N}$  of the BSCA-F at least one new coalition may form, i.e.  $|\mathcal{C}|_{r+1} \leq |\mathcal{C}|_r - 1$ . Thus, after  $|\mathcal{A}| - 1$  rounds, we have  $|\mathcal{C}|_{|\mathcal{A}|-1} \leq 1$ , which means that the BSCA-F terminates in round  $|\mathcal{A}|$ .  $\square$

**Theorem 3.5** *The worst-case runtime of the BSCA-F for each agent is in  $O(|\mathcal{A}|^4)$  assuming constant time for operations on fuzzy quantities.*

*Proof:* In step 2.b, each  $C$  has to find the (crisp) maximum set of the fuzzy gains for coalitions in  $Cand_C$ , with  $|Cand_C| \leq |\mathcal{C}_r|$ . From definitions 2.6 and ?? it follows that this can be done in  $O(|\mathcal{C}_r|^2)$ . Because all other individual operations are of less complexity and with lemma 3.3, the iteration between step 2.b and 3.d thus is in  $O(|\mathcal{C}_r|^3)$ . Since  $|\mathcal{C}_r| \leq |\mathcal{A}|$  and lemma 3.4, the overall runtime of the BSCA-F is then  $O(|\mathcal{A}|^4)$ .  $\square$

**Theorem 3.6** *The total number of messages sent by agents using the BSCA-F for coalition negotiation is  $O(|\mathcal{A}|^2)$ .*

*Proof:* In each round  $r \in \mathbb{N}$ , each representative of a coalition  $C$  sends  $|\mathcal{C}_r| - 1$  messages in step 1.a, a single proposal message in 2.b, at most one time  $|\mathcal{C}_r| - 2$  messages in step 3.b.ii and  $|C| - 1$  messages in step 3e. So the number of messages per representative per round is bound by  $|C| \leq |\mathcal{A}|$ . The number of messages sent by the  $|\mathcal{A}| - |C|$  non-representatives is zero. So with lemma 3.4, the overall number of messages sent is lower or equal  $|\mathcal{A}|^2$ .  $\square$

Coalition negotiation using the BSCA-F yields a coalition structure  $\mathcal{C}$  with recursively fuzzy bilateral Shapley value stable payoff distribution. Since these fuzzy payoffs originate from the fuzzy coalition values of coalitions in  $\mathcal{C}$  only (all other fuzzy coalition values are defuzzified by use of the possibilistic mean value) we have to defuzzify the values of exactly these coalitions in  $\mathcal{C}$  only. It appears plausible that the real coalition values become known to the corresponding coalition members after their coalitions have been formed and contracted actions are carried out. Otherwise, we may also use the possibilistic mean to derive at least a reasonable expectation of them. Thus, in both cases, we may obtain crisp coalition values for all coalitions in  $\mathcal{C}$ , hence crisp payoffs.

## 4 Example Application

### 4.1 Definition of the Game

In the following, we demonstrate how the BSCA-F could be applied to negotiate economically rational coalitions of



Cat.	$M_1$	$M_2$	$M_3$	$M_4$
a)	politics	column	column 1	photo mag
b)	feature sect.	travel mag	column 2	feature sect.

**Table 1. Content provided by magazines**

$I_1$	$\tilde{A}^1$	$I_2$	$\tilde{A}^2$
$M_2$ a)	(2.4, 3.6, 4.2, 4.8)	$M_1$ a)	(8.4, 12, 14.4, 16.8)
$M_2$ b)	(1.08, 1.2, 1.56, 1.8)	$M_3$ a)	(6.6, 7.2, 7.8, 8.4)
$M_3$ a)	(9.6, 12, 14.4, 15.6)	$M_4$ a)	(6, 9, 9.333, 10)
$M_4$ a)	(3.6, 7.2, 12.8, 16.4)		
$I_3$	$\tilde{A}^3$	$I_4$	$\tilde{A}^4$
$M_2$ b)	(13.2, 14.4, 15, 15.6)	$M_1$ b)	(3.6, 3.72, 4.08, 4.2)
		$M_2$ b)	(7.2, 7.56, 10, 12.5)

**Table 2. Additional income per category**

online magazines in the Internet. Consider four online magazines,  $M_1 - M_4$ , that are interested in exchanging content for different reasons such as customer recruitment. Suppose that each of them is reluctant to provide more content to potential partners than it would obtain in return for a certain payoff, hence agrees to contribute only two categories of the content to the coalition it participates in table 1). Content is provided to coalition partners on a daily basis, whereas coalition contracts in total will hold for one year after which negotiation may be restarted. To prevent antitrust matters, coalitions with more than three members are ruled out. Each magazine  $M_i$  is represented by an agent  $a_i$  which carries out negotiation on behalf of  $M_i$ . Each magazine  $M_i$  would publish only such content provided by coalition partners which is in line with the general style of  $M_i$ . The set of categories  $M_i$  is interested in is called  $I_i$ . For each  $x \in I_i$ ,  $M_i$  fuzzily estimates the number  $\tilde{A}_x^i$  of additional accesses for one year it can achieve by publishing  $x$ . For simplicity, we assume these estimations are independent of each other. The sets  $I_i$  and estimations  $\tilde{A}_x^i$  (in thousands) are given in table 2. Any single access to content of a magazine  $M_i$ ,  $1 \leq i \leq 4$ , is subject to a given price  $P_i$  (in Euros) determined by  $M_i$ . The prices are  $P_1 := 2$ ,  $P_2 := 1.5$ ,  $P_3 := 1.8$  and  $P_4 := 2.0$ . The additional income produced by a magazine  $M_i$  by coalescing with a magazine  $M_k$  is  $\sum_{M_{kx} \in I_i} \tilde{A}_{M_{kx}}^i \odot P_i$ , and the total additional income  $\tilde{a}i_i(C)$  for  $M_i$  in coalition  $C$  is given with  $\tilde{a}i_i(C) = \sum_{M_k \in C, k \neq i} \sum_{M_{kx} \in I_i} \tilde{A}_{M_{kx}}^i \odot P_i$ .  $M_1$  and  $M_2$  arguably have the best cooperation opportunities in this game. On the cost side, we consider only volume-based transfer costs (in Euros) with given transfer price  $T_i$  per MB depending on the internet connection of magazine  $M_i$ . The costs are  $T_1 := 0.02$ ,  $T_2 := 0.01$ ,  $T_3 := 0.025$  and  $T_4 := 0.012$ . Based on experiences in the past, each  $M_i$  estimates the amount of data  $V_x$  it would need to transfer

	$V_{M,a}$	$V_{M,b}$
$M_1$	(12, 18, 24, 42)	(6, 12, 30, 60)
$M_2$	(1.2, 3.6, 6, 7.2)	(96, 120, 156, 204)
$M_3$	(3.6, 8.4, 12, 14.4)	(2.4, 3.6, 4.2, 5.4)
$M_4$	(180, 192, 204, 216)	(18, 24, 30, 36)

**Table 3. Amount of category data (in 100MB)**

C	$\tilde{v}(C)$
$a_1, a_2$	(38060, 48552, 60739, 70454)
$a_1, a_3$	(18835, 23789, 28674, 31128)
$a_1, a_4$	(13338, 20976, 33022, 40552)
$a_2, a_3$	(32967, 36185, 38293, 40370)
$a_2, a_4$	(19010, 22932, 32981, 39114)
$a_3, a_4$	(-815, -751, -684, -612)
$a_1, a_2, a_3$	(89564, 108332, 127576, 141865)
$a_1, a_2, a_4$	(69646, 91830, 126204, 149649)
$a_1, a_3, a_4$	(30690, 43458, 60529, 70642)
$a_2, a_3, a_4$	(50331, 57648, 69967, 78328)
others	(0)

**Table 4. Coalition values (rounded)**

per category  $x$  during one year as shown in table 3. Every magazine  $M_i$  has to pay for both incoming and outgoing traffic, means  $(V_{M_i a} \oplus V_{M_i b} \oplus V_{M_k a} \oplus V_{M_k b}) \odot T_i$  for data transmitted to/from each coalition partner  $M_k$ . Hence, the total cost  $\tilde{c}_i(C)$  for  $M_i$  in coalition  $C$  is  $\tilde{c}_i(C) = \sum_{M_k \in C, k \neq i} (V_{M_i a} \oplus V_{M_i b} \oplus V_{M_k a} \oplus V_{M_k b}) \odot T_i$ . Having both total additional income and costs for each magazine  $M_i$  in a coalition  $C$ , we obtain their local worth individually by  $lworth_{a_i}(C) = \tilde{a}i_i(C) \ominus \tilde{c}_i(C)$ , and resulting fuzzy coalition values (cf. table 4).

## 4.2 Negotiation with the BSCA-F

For this example, we select the necessity of dominance  $\tilde{\succ}_N$  as fuzzy ranking operator, and  $o(a_i) := i$  for agent ordering. In the first round, in step 2a, all coalitions prefer each other except of  $a_3 \cup a_4$  which is clearly a non-profitable coalition. Both  $a_1$  and  $a_2$  mutually propose  $a_1 \cup a_2$  to each other as the most profitable joint coalition with payoff half of the coalition value:  $\tilde{u}(a_1) = \tilde{u}(a_2) = \frac{1}{2} \tilde{0} \oplus \frac{1}{2} ((38060, 48552, 60739, 70454) \ominus \tilde{0}) = (19030, 24276, 30369.5, 35227)$ . In the second round,  $a_1 \cup a_2$  sends a proposal to  $a_3$  rather than  $a_4$  according to  $\tilde{\succ}_N$  with  $e(\tilde{v}(\{a_1, a_2\})) = \frac{38060+48552+60739+70454}{4} = 54451$ ,  $\tilde{g}_{\tilde{v}}(a_1 \cup a_2, (a_1 \cup a_2) \cup a_3) = \frac{1}{2} (\tilde{v}(\{a_1, a_2, a_3\}) \ominus e(\tilde{v}(\{a_1, a_2\})) \ominus e(\tilde{v}(\{a_3\})))$

$= \frac{1}{2}((89564, 108332, 127576, 141865) \ominus 54451 \ominus 0)$   
 $= (17556, 26940, 36562, 43706) = \tilde{g}_{\tilde{v}}(a_3, (a_1 \cup a_2) \cup a_3)$   
 (cf. lemma 2.16). Similarly,  $\tilde{g}_{\tilde{v}}(a_1 \cup a_2, (a_1 \cup a_2) \cup a_4)$   
 $= \tilde{g}_{\tilde{v}}(a_4, (a_1 \cup a_2) \cup a_4) = (7597, 18689, 35876, 47599)$   
 Please note that with  $>_P$ , the choice would have been  $a_4$ .  
 The payoff of the new coalition is distributed as follows:  
 $\tilde{u}(a_1) = \tilde{u}(a_2) = \tilde{\sigma}_b^e(a_1 \cup a_2, a_1, \tilde{\sigma}_b^e(C^*, a_1 \cup a_2, \tilde{v}(C^*)))$   
 $= \frac{1}{2}(\frac{1}{2}e(\tilde{v}(a_1 \cup a_2)) \oplus \frac{1}{2}((89564, 108332, 127576, 141865)$   
 $\ominus 0)) = (17556, 26940, 36562, 43706)$ . In the third round,  
 the BSCA-F terminates, since the value of the grand  
 coalition is zero, thus is not a candidate for anyone.

### 4.3 Defuzzification

Since coalition contracts, in this example, are valid  
 for one year, there are two options to determine the real  
 coalition values: Either wait for one year and then ana-  
 lyze the additional income and the costs that were real-  
 ized in the period; or defuzzify the coalition values just  
 when negotiations are finished, using the possibilistic mean  
 value. Due to space limitations, we only discuss the sec-  
 ond case for the coalition  $C^* := \{a_1, a_2, a_3\}$  formed,  
 and the possibilistic mean value of  $\tilde{v}(C^*)$ :  $e(\tilde{v}(C^*)) =$   
 $\frac{89564+108332+127576+141865}{4} = 116834$ . Computing the re-  
 cursive bilateral Shapley value with  $e(\tilde{v}(C^*))$ , we obtain  
 $u(a_1) = u(a_2) = 42821$ , which is equal to  $e(\tilde{u}(a_1)) (=$   
 $e(\tilde{u}(a_2))$  due to the additivity of  $e$ . Similarly, it holds that  
 $u(a_3) = e(\tilde{u}(a_3)) = 31192$ . To summarize, when negotia-  
 tions are finished, the agent have to compute the possibilis-  
 tic mean values of the fuzzy payoffs only. this yields the  
 same result as if they would compute the recursively bilat-  
 eral Shapley value stable payoffs for the possibilistic mean  
 of the coalition values. The resulting payoffs appear to be  
 intuitively sound, since  $a_1$  and  $a_2$ , the agents with most ben-  
 efiticial cooperation opportunities, are assigned more payoff  
 than  $a_3$ . Further, let us consider computing the fuzzy pay-  
 offs by recursively applying the non-defuzzifying fuzzy bilat-  
 eral Shapley value as defined in ?? instead. This means  
 that the fuzzy payoffs now also contain the fuzzyness of the  
 values of the subcoalitions. Then we obtain the fuzzy pay-  
 offs  $\tilde{u}^*(a_1) = \tilde{\sigma}_b(a_1 \cup a_2, a_1, \tilde{\sigma}_b(C^*, a_1 \cup a_2, \tilde{v}(C^*)))$   
 $= (31906, 39221, 47079, 53080) (= \tilde{u}^*(a_2))$  and  $\tilde{u}^*(a_3)$   
 $= \tilde{\sigma}_b(C^*, a_3, \tilde{v}(C^*)) = (9555, 23797, 39512, 51903)$ .

## 5 Conclusions

We presented a new low-complexity coalition forming  
 algorithm, BSCA-F, that enables agents to negotiate bilat-  
 eral Shapley value stable coalitions in uncertain environ-  
 ments, and demonstrated it by example. In particular, we  
 showed that utilizing the possibilistic mean value for de-  
 fuzzifying negotiated fuzzy agent payoffs appears reason-

able. However, the choice of the fuzzy ranking operator is  
 supposed to be equal for each agent; future work includes  
 relaxation of this requirement.

## References

- [1] B. Blankenburg, M. Klusch, and O. Shehory. Fuzzy kernel-  
stable coalitions between rational agents. In *Proc. 2<sup>nd</sup> Int.  
Conference on Autonomous Agents and Multiagent Systems  
(AAMAS 2003), Melbourne, Australia, 2003*.
- [2] G. Bortolan and R. Degani. A review of some methods of  
ranking fuzzy subsets. *Fuzzy Sets and Systems*, 15(1), 1985.
- [3] Christer Carlsson and Robert Fuller. On possibilistic mean  
and variance of fuzzy numbers. *Fuzzy Sets and Systems*, 122,  
2001.
- [4] Christer Carlsson and Robert Fuller. On weighted possibilis-  
tic mean and variance of fuzzy numbers. *Fuzzy Sets and  
Systems*, 136, 2003.
- [5] Georgios Chalkiadakis and Craig Boutilier. Bayesian rein-  
forcement learning for coalition formation under uncertainty.  
In *Proc. 3<sup>rd</sup> Int. Conference on Autonomous Agents and Mul-  
tiagent Systems (AAMAS 2004), New York, USA, New York,  
USA, 2004*. ACM Press.
- [6] D. Dubois and H. Prade. Ranking fuzzy numbers in the set-  
ting of possibility theory. *Information Sciences*, 30, 1983.
- [7] D. Dubois and H. Prade. Fuzzy numbers: an overview. In  
James C. Bezdek, editor, *Analysis of fuzzy information*, vol-  
ume I Mathematics and Logic. CRC Press, 1994.
- [8] Matthias Klusch and Onn Shehory. A polynomial kernel-  
oriented coalition algorithm for rational information agents.  
In *Proc. 2. International Conference on Multi-Agent Systems  
ICMAS-96*. AAAI Press, 1996.
- [9] S. Kraus, O. Shehory, and G. Tasse. Coalition formation with  
uncertain heterogeneous information. In *Proc. 2<sup>nd</sup> Int. Con-  
ference on Autonomous Agents and Multiagent Systems (AA-  
MAS 2003), Melbourne, Australia, New York, USA, 2003*.  
ACM Press.
- [10] Milan Mareš. *Fuzzy cooperative games: cooperation with  
vague expectations*, volume 72 of *Studies in fuzziness and  
soft computing*. Physica Verlag, Heidelberg ; New York,  
2001.
- [11] Eric Raufaste, Rui Da Silva Neves, and Claudette Marin?  
Testing the descriptive validity of possibility theory in human  
judgments of uncertainty. *Artificial Intelligence*, 148(1-2),  
2003.
- [12] O. Shehory and S. Kraus. Feasible formation of coalitions  
among autonomous agents. *Computational Intelligence*,  
15(3), 1999.
- [13] Lotfi A. Zadeh. Fuzzy sets as a basis for a theory of possi-  
bility. *Fuzzy Sets and Systems*, 1, 1978.

---

## Negotiation of Fuzzy Coalitions

B. Blankenburg, M. He, M. Klusch, N. Jennings: Risk Bounded Formation of Fuzzy Coalitions Among Service Agents. Proceedings of the 10th International Workshop on Cooperative Information Agents, Edinburgh, UK, Lecture Notes in Artificial Intelligence (LNAI), 4149, pages 332 - 346, Springer, 2006.

# Risk-bounded Formation of Fuzzy Coalitions among Service Agents

Bastian Blankenburg<sup>1</sup>, Minghua He<sup>2</sup>, Matthias Klusch<sup>1</sup>, and Nicholas R. Jennings<sup>2</sup>

<sup>1</sup> German Research Center for Artificial Intelligence, Stuhlsatzenhausweg 3,  
66123 Saarbrücken, Germany; {blankenb,klusch}@dfki.de

<sup>2</sup> School of Electronics and Computer Science, University of Southampton,  
Southampton, SO17 1BJ, UK; {mh,nrj}@ecs.soton.ac.uk

**Abstract.** Cooperative autonomous agents form coalitions in order to share and combine resources and services to efficiently respond to market demands. With the variety of resources and services provided online today, there is a need for stable and flexible techniques to support the automation of agent coalition formation in this context. This paper describes an approach to the problem based on fuzzy coalitions. Compared with a classic cooperative game with crisp coalitions (where each agent is a full member of exactly one coalition), an agent can participate in multiple coalitions with varying degrees of involvement. This gives the agents more freedom and flexibility, allowing them to make full use of their resources, thus maximising utility, even if only comparatively small coalitions are formed. An important aspect of our approach is that the agents can control and bound the risk caused by the possible failure or default of some partner agents by spreading their involvement in diverse coalitions.

## 1 Introduction

In today's increasingly networked and competitive world, the appropriate utilization of pay per use Web services are considered as one major key to the success of commercial service oriented business applications in domains such as e-logistics, tourism, and entertainment. In the near future, intelligent service agents are not only supposed to search for, interact with, and compose, but also negotiate access to, and execute such Web services on behalf of its user, or other agents. In fact, they may exhibit some form of economically rational cooperation by forming coalitions to share the created joint monetary value while at the same time maximizing their own individual payoff. According to classical microeconomics, means and concepts of cooperative game theory are inherently well suited to this purpose. In this paper, we propose a protocol for resource-bounded computational rational agents to automatically form risk-bounded fuzzy coalitions in order to fulfill service requests with deadlines.

As opposed to traditional cooperative games, games with fuzzy coalitions allow the agents to be members of multiple coalitions with varying degrees of involvement. The notion of fuzzy coalitions was first introduced by Aubin and Butnariu (see [2, 5]) to overcome some problems of traditional cooperative games in real-world settings. For example, suppose that agent  $a_1$  can independently benefit from cooperations with agent  $a_2$  as well as with agent  $a_3$ . To realise both opportunities, a coalition of all three agents

has to be formed, requiring  $a_2$  and  $a_3$  to agree on a coalition contract although they do not cooperate otherwise. Another drawback of non-overlapping coalitions emerges in the case of failure. If, in the above example,  $a_2$  fails its task, thus reducing the coalition value, all members of the coalition are affected, including  $a_3$ , although it is not actually working together with  $a_2$ . In contrast, with fuzzy coalitions makes it is possible to form a coalition for each service request, without preventing other requests from being satisfied. This approach has the advantage that there are no unnecessary negotiations and contracts between agents which actually do not work together.

Additionally, using fuzzy coalitions allows the agents to lower their individual risk of monetary losses by participating in a number of coalitions, if coherent risk measure are considered. Assuming that agents are able to assess other agent's risk of failure in a coalition, we show how such risk-bounded coalition formation can be done. In particular, we consider the membership of an agent in a coalition as an investment, since the costly service execution takes place first. Rewards are received later only for successful and timely execution. We thus allow the agents to specify individual risk bounds in terms of the coherent financial risk measure *tail conditional expectation* (TCE). The adherence to these bounds is guaranteed by the proposed coalition formation protocol RFCF.

But as it turns out, we cannot directly use the existing solution concepts for cooperative games with fuzzy coalitions. The approaches taken by Aubin, Butnariu as well as Nishizaki and Sakawa (see [9]) all assume that the coalition value is a proportional function of the agents' membership degrees. As this assumption does not hold in our setting, we introduce appropriate extensions of the excess and surplus. We then show that it is possible to compute the surplus in polynomial time under some additional assumptions, similar to the approach taken in [10]. Stearns transfer scheme can then be used to compute Kernel-stable solutions for the game (see [11]).

The remainder of this paper is organized as follows: in section 2 we introduce our service agent and coalition model. In section 3, we introduce our notion of fuzzy coalition games among service provider agents. We then show how to compute the risk of fuzzy coalitions and fuzzy coalition structures in section 4. Section 5 is concerned with the stability of risk-bounded fuzzy coalitions. We propose our coalition formation protocol RFCF in section 6. In section 7 we discuss related work and conclude in section 8.

## 2 Agent Model

In this section we specify more precisely the environment of service agents that we consider in this paper.

We consider two types of agents: service request agents and service provider agents.

### **Definition 1** *Service Request Agent*

*A service request agent  $sra$  requests exactly one (possibly complex) service  $s$  and some deadline  $d$ . It will pay a certain monetary reward  $r \in \mathbb{R}$  for a successful execution of  $s$  before  $d$ . Otherwise, no reward is paid.*

*SRA denotes the set of all service request agents in the system.*

On the other hand, service provider agents offer the execution of exactly one type of service. They are assumed to be computationally bounded, i.e. to have only limited resources per time for the execution of their service. For simplicity, we assume that the execution time for a service instance is a linear function of the resources devoted to it. This is reasonable in the case where the bounded resources are computing power and/or memory, for example.

**Definition 2** *Service Provider Agent*

A service provider agent *spa* offers the execution of exactly one service  $s_{spa}$  and has the following properties:

1. *Service Composition*

- (a) *spa* is able to send service advertisements for  $s_{spa}$ .
- (b) given a requested service  $s$  and a set of service advertisements, *spa* has the ability to compute service composition plans; each such plan is a list of advertised services whose execution implements the requested service  $s$ .
- (c) each element of a plan  $\mathcal{P}$  is called a service instance of the respective service.

2. *Service Execution*

- (a) *spa* can spend only some max. amount of resources per time in service executions.
- (b) the minimum execution time of an instance  $i$  of  $s_{spa}$  is denoted  $t_i^{min}$  (i.e. this is the execution time if *spa* devotes all its resources to it).
- (c) *spa* can split its resources and execute multiple instances of  $s_{spa}$  at the same time. The fraction of resources per time (wrt. the maximum) devoted to the execution of service instance  $i$  is denoted  $r_i$ .
- (d) the execution time  $t_i$  of service instance  $i$  is

$$t_i = \frac{1}{r_i} \times t_i^{min}.$$

- (e) *spa* might not be able to determine  $t_i^{min}$  exactly in advance, but is able to specify a probability density function (PDF)  $pdf_{t_i^{min}}$  over the values it might take.
- (f) there is a monetary cost for resource consumption of *spa*. We assume this is constant, so that because of Definition 2.2(d) the cost  $cost_i$  for executing service instance  $i$  is also constant and does not depend on  $r_i$ .

*SPA* denotes the set of all service provider agents in the system.

Note that because of the linear relationship assumed in Definition 2.2(d), it is easy to obtain the PDF of the execution time of a service instance  $i$  with a given fraction of resources per time  $r_i$ :

$$pdf_{t_i}(x) = pdf_{t_i^{min}}(r_i * x) \tag{1}$$

*Example 1.* As an example, we consider a medical service provider agent scenario. We assume that there are a number of these agents in the system, each offering medical information in one or more specific medical domains. A specific set of symptoms of a patient might have possible diagnosis in several domains. Thus, a full diagnosis as response to a request from e.g. a medical doctor might require a set of provider agents

to collaborate. We assume that the medical personnel will request this information with specific deadlines to ensure the timely treatment of patients. Suppose that agent  $spa_1$  gets a request from a doctor and realizes that it also needs  $spa_2$  to provide a feasible diagnosis.  $spa_1$  then estimates the runtime for its own service on the request and sends coalition proposal to  $spa_2$ .  $spa_2$  then likewise estimates its runtime and sees that this coalition might actually fail, producing high costs. However,  $spa_2$  has a further request from agent  $spa_3$ . While forming just the coalition  $spa_1$  is too risky for  $spa_2$ , it is acceptable if the coalition with  $spa_3$  is also formed.

### 3 Fuzzy Coalition Games of SPA Agents

In our setting, the capability of service provider agents to split their resources among different service instance executions makes it possible for them to take part in several service composition plan executions. This suggests to allow the agents to be a (partial) member of several coalitions. For this purpose, a number of authors (most notably Aubin, Butnariu and Nishizaki and Sakawa [2, 5, 9]) extended concepts from cooperative game theory to allow for *fuzzy coalitions*, where each agent is a member only to a certain *membership degree*. In our model, each fuzzy coalition will execute exactly one service composition plan. The membership degree represents the relative amount of resources they spend for their respective service instance executions in the plan. If the same group of agents decides to execute an additional plan, it simply forms an additional fuzzy coalition. We also disallow any members that are not actually involved in the execution of  $P$ .

#### Definition 3 Fuzzy Coalition of Service Provider Agents

Let there be a request for a service  $ws$  from a service request agent  $sra$  and a plan  $\mathcal{P}$  whose execution satisfies  $ws$ .

1.  $SPA_{\mathcal{P}} \subseteq SPA$  is the set of service provider agents involved in  $\mathcal{P}$ .
2. The fuzzy coalition of service provider agents  $\tilde{C}$  for  $sra$  and  $\mathcal{P}$  is written as

$$\tilde{C} = (spa_1/mem_1, \dots, spa_k/mem_k, sra, \mathcal{P})$$

with  $k = |SPA_{\mathcal{P}}|$ ,  $spa_j \in SPA_{\mathcal{P}}$ ,  $1 \leq j \leq k$ ;  $mem_j \in [0, 1]$  is a guaranteed minimum for the fraction of resources per time  $r_i$  devoted by  $spa_j$  to any  $i$  of its service instances in  $\mathcal{P}$ .

3.  $mem(spa, \tilde{C})$  is agent  $spa$ 's membership in  $\tilde{C}$ .
4. We write  $spa \in \tilde{C}$  if  $spa$  is a member of  $\tilde{C}$  with some positive membership, i.e.  $mem(spa, \tilde{C}) > 0$ .
5.  $\tilde{C} \subseteq \tilde{C}'$  if  $\forall spa \in \tilde{C} : mem(spa, \tilde{C}) \leq mem(spa, \tilde{C}')$ , where  $\tilde{C}$  and  $\tilde{C}'$  are fuzzy coalitions for the same service request agent and plan.
6.  $\tilde{C}(sra, plan)$  denotes the set of all fuzzy coalitions  $\tilde{C} = (., sra, plan)$ .
7.  $|\tilde{C}|$  is the number of agents in  $\tilde{C}$ .

We also denote “fuzzy coalition” or just “coalition” instead of “fuzzy coalition of service provider agents” where the context is clear.

Because of the deadlines for service requests,  $\tilde{C}$  either earns the reward  $r$  for the successful and timely execution of  $\mathcal{P}$  from the requesting agent, or nothing otherwise. To specify coalition values for fuzzy service provider agent coalitions, we thus need to consider its probabilities of failure and success. For simplicity, we assume that the execution times of service instances are independent of each other and that the services in a plan  $\mathcal{P}$  are executed sequentially. Then, the total execution time of  $\mathcal{P}$  is the sum of the execution times of the individual service instances:

$$t_{\mathcal{P}} = \sum_{i \in \mathcal{P}} t_i \quad (2)$$

The PDF of the sum of two independent random variables  $A$  and  $B$  is given by the *convolution integral* over their individual PDFs  $pdf_A$  and  $pdf_B$  (see, e.g., [8], p. 113). I.e., with  $x \in \mathbb{R}$ :

$$\begin{aligned} pdf_{A+B}(x) &= (pdf_A * pdf_B)(x) \\ &= \int_0^{\infty} pdf_A(y) pdf_B(x-y) dy \end{aligned} \quad (3)$$

For a plan  $\mathcal{P}$  with  $m \in \mathbb{N}$  service instances, the PDF of its execution time is therefore an  $m - 1$  fold convolution over the individual service instance execution time PDFs. With  $x \in \mathbb{R}^+$  (it is sufficient to consider only positive values since execution times are always positive).

$$pdf_{t_{\mathcal{P}}}(x) = (\cdots (pdf_{t_{i_1}} * pdf_{t_{i_2}}) \cdots * pdf_{t_{i_m}})(x) \quad (4)$$

For specific cases, there exist simple analytical solutions of the convolution. E.g., the convolution of two normal PDFs is again normal, as is the convolution of a normal PDF with an exponential one. But this is not the case for arbitrary distribution types. Fortunately, there are alternative ways to obtain the convolution, such as the pointwise multiplication of the Fourier Transform  $F$  of the PDFs:

$$f * g = F^{-1}(F(f)F(g)) \quad (5)$$

The Fast Fourier Transform algorithm efficiently approximates the Fourier Transform with complexity  $k \log k$ , where  $k$  is the number of sample points taken from the functions.

Suppose the agents executing a plan  $\mathcal{P}$  agree to start the execution at time  $t_s$ . With the PDF of the execution time of a plan  $\mathcal{P}$  and the deadline given for the respective service request, it is then easy to determine the probability that the plan execution exceeds this deadline, which we call the probability of failure ( $PoF$ ):

$$PoF(\mathcal{P}, t_s, d) = \int_{d-t_s}^{\infty} pdf_{t_{\mathcal{P}}}(x) dx \quad (6)$$

Note that for  $d < t_s$ , we always have  $PoF(\mathcal{P}, t_s, d) = 0$ , since the plan execution time must be positive. Similarly, the probability of success ( $PoS$ ) is:

$$PoS(\mathcal{P}, t_s, d) = 1 - PoF(\mathcal{P}, t_s, d) \quad (7)$$



Given the membership degrees, the PDF over the upper bound  $\hat{t}_i$  for the execution time of a service instance  $i \in \mathcal{P}$  of agent  $spa_k$  is, analogous to 1,

$$pdf_{\hat{t}_i}(x) = pdf_{t_i^{min}}(mem_k * x) \quad (8)$$

According to 4, we can then obtain the PDF of the upper bound  $\hat{t}_{\mathcal{P}}$  for the execution time of the complete plan, and thus the probabilities of failure and success of the fuzzy coalition, denoted  $PoF(\tilde{C})$  and  $PoS(\tilde{C})$ , resp. This enables us to determine a lower bound for the expected reward for  $\tilde{C}$ , denoted  $r_{\tilde{C}}$ :

$$r_{\tilde{C}} = PoS(\tilde{C}) \times r \quad (9)$$

To specify a value for the fuzzy coalitions, we further have to consider the costs that are generated by the service executions. The agents should reasonably stop the execution once the deadline is reached, since no additional reward can be obtained by any further work. However, to simplify things, we consider only the worst case, i.e. the case where maximum costs have been produced even if the coalition fails.

**Definition 4** *Value of a Fuzzy Service Provider Agent Coalition*

Let there be a fuzzy coalition  $\tilde{C}$  with plan  $\mathcal{P}$ . The value  $v(\tilde{C})$  of  $\tilde{C}$ , also called coalition value, is defined as

$$v(\tilde{C}) = r_{\tilde{C}} - \sum_{i \in \mathcal{P}} cost_i$$

Although fuzzy coalition structures allow the agents to be a member in several coalitions at the same time, we still have to require that each agent does not allocate more resources to coalitions than it can actually provide. Formally, we have

**Definition 5** *Feasible Fuzzy Coalition Structure*

For a fuzzy coalition  $\tilde{C}$ , let  $mem_{spa}^{\tilde{C}}$  denote the membership degree of  $spa$  in  $\tilde{C}$ , with  $mem_{spa}^{\tilde{C}} = 0$  if  $spa$  is not member of  $\tilde{C}$ . A feasible fuzzy coalition structure  $\mathcal{S}$  for the agents in  $SPA$  is defined as a set of fuzzy coalitions with

$$\forall spa \in SPA : \sum_{\tilde{C} \in \mathcal{S}} mem_{spa}^{\tilde{C}} \leq 1 \quad (10)$$

## 4 Risk of Fuzzy Coalition Structures

Given a variety of combination of coalitions that the agent can possibly join, rational agents will prefer coalitions with a high reward and a low  $PoF$ , i.e. a high expected value. But assume there is a coalition with a high expected value, but which also involves very high costs. If an agent cannot afford to lose more than some amount without compromising liquidity, even a low  $PoF$  of the coalition might be still too risky. To control and avoid such situations, a number of financial risk measures have been introduced in the literature (for a recent overview, see [7] and references therein).

For the definitions in the remainder of this section, we follow Artzner et al.[1], omitting certain details which are not important in our setting. Also, where Artzner et al. speak of *positions* (meaning investment positions), we speak of *strategies*, meaning an agent's decision with whom to coalesce and service requests to work on. Lastly, note that the definitions of *VaR* and other measures in [1] include the reward of a reference investment (e.g. interest rates) as a scaling factor, which we omit here for simplicity.

**Definition 6** *Risk and Measure of Risk*

Let  $\Omega$  denote the set of states of nature, and assume it is finite. Considering  $\Omega$  as the set of outcomes of an experiment, we compute the final net worth of a strategy for each element of  $\Omega$ . Risk is the investor's future net worth, which is described by a random variable. Let  $G$  be the set of all risks, that is the set of all real valued functions on  $\Omega$ . A measure of risk  $r$  is a mapping  $r: G \mapsto \mathbb{R}$ .

According to [7], a widely known and used one is the Value-at-Risk (*VaR*), which also has become part of financial regulations. *VaR* calculates how much one may lose during a specified period given a probability and the capital should be used to control the risk.

**Definition 7** *Value-at-Risk (VaR)*

Given  $\alpha \in [0, 1]$ , the Value-at-Risk  $VaR^\alpha$  at level  $\alpha$  of the final net worth  $X \in G$  with distribution  $P$  is

$$VaR^\alpha(X) = -\inf\{x \in \mathbb{R} : P(X \leq x) > \alpha\}$$

Artzner et al. also introduce the notion of *coherent* risk measures.

**Definition 8** *Coherent risk measure*

With  $X, Y \in G, z \in \mathbb{R}$ , a risk measure  $r$  is called *coherent* if it satisfies

1. *subadditivity*: for all  $X, Y \in G: r(X + Y) \leq r(X) + r(Y)$
2. *translation invariance*:  $r(X + z) = r(X) - z$
3. *positive homogeneity*:  $\forall z \geq 0, r(zX) = zr(X)$
4. *monotonicity*: if  $X \leq Y$  then  $r(Y) \leq r(X)$

As has also been shown in [1], *VaR* is not coherent, since it does not fulfill subadditivity. As it turns out (see below), this lack of superadditivity constitutes a major drawback in the design of a risk-bound coalition formation algorithm. Fortunately, a number of coherent measures which are derived from *VaR* have been proposed. Here, we employ the tail conditional expectation (*TCE*) which is coherent for continuous distributions.

**Definition 9** *Tail Conditional Expectation*: given a probability measure  $P$  on  $\Omega$  and a level  $\alpha$ , the tail conditional expectation is defined by:

$$TCE^\alpha(X) = -E_P\{X|X \leq -VaR^\alpha(X)\}$$

Using this measure, each agent  $spa_i$  may individually specify a parameter  $\alpha_i$  and a  $TCE$ -threshold  $tTCE_i$ , expressing that it will only accept coalition structures which satisfy

$$TCE^{\alpha_i}(u_i) \leq tTCE_i$$

where  $u_i$  is agent  $spa_i$ 's final net worth, i.e. the total net result from all coalitions it is involved in.

**Proposition 1** *Let service provider agent  $spa_i$  be a member in a fuzzy coalition  $\tilde{C}$ , let  $cost_i$  be the cost for  $spa_i$  if  $\tilde{C}$  fails, and let  $u_i(\tilde{C}) > -cost_i$  be the payoff obtained by  $spa_i$  if  $\tilde{C}$  is successful. The  $TCE^{\alpha_i}(\tilde{C})$ , i.e. the  $TCE^\alpha$  restricted to consider only  $spa_i$  and  $\tilde{C}$ , can be computed as follows:*

$$TCE^{\alpha_i}(\tilde{C}) = \begin{cases} PoF(\tilde{C})cost_i(\tilde{C}) + PoS(\tilde{C})(-u_i(\tilde{C})) & PoF(\tilde{C}) \leq \alpha_i \\ cost_i(\tilde{C}) & PoF(\tilde{C}) > \alpha_i \end{cases}$$

*Proof.* Let  $X_i$  be  $spa_i$ 's net result from  $\tilde{C}$ , with  $X_i = u_i$  in case of success of  $\tilde{C}$  and  $X_i = -cost_i$  in case of failure. Consider the first case, i.e. assume that  $PoF(\tilde{C}) \leq \alpha_i$ . Then the Value-at-Risk, i.e. the  $TCE^\alpha$  restricted to consider only  $spa_i$  and  $\tilde{C}$ , is  $Var^{\alpha_i}(\tilde{C}) = -u_i$  because  $P(X_i \leq -cost_i) = PoF(\tilde{C}) \not\leq \alpha_i$ , but  $P(X_i \leq u_i) = 1$  (since  $PoS(\tilde{C}) = 1 - PoF(\tilde{C})$ ). Thus, the set of relevant outcomes considered in  $TCE^\alpha$  includes both  $X_i = -cost_i$  and  $X_i = u_i$ . In the second case, with  $PoF(\tilde{C}) > \alpha_i$ , we have  $Var^{\alpha_i}(\tilde{C}) = cost_i$  because  $P(X_i \leq -cost_i) = PoF(\tilde{C}) > \alpha_i$ . Thus, the set of relevant outcomes considered in  $TCE^\alpha$  contains only  $X_i = -cost_i$ , and the case  $X_i = u_i$  is disregarded.

To obtain the  $TCE^{\alpha_i}$  for a fuzzy coalition structure, we have to consider the probability of failure for each subset of fuzzy coalitions that  $spa_i$  is involved in, as well as the payoffs and costs for  $spa_i$  in these cases. The following follows directly from the independency of the  $PoF$  of different coalitions and the definition of  $Var$ .

**Corollary 1** *Let there be a fuzzy coalition structure  $S$  and let  $S_{spa_i} \subseteq S$  be the subset of all coalitions involving  $spa_i$ . For each  $S_{spa_i}^* \in 2^{S_{spa_i}}$  (including the empty set) let  $cost_i(S_{spa_i}^*)$  be the cost for  $spa_i$  if all coalitions in  $S_{spa_i}^*$  fail, and let  $u_i(S_{spa_i}^*)$  be the net payoff obtained by  $spa_i$  from the coalitions in  $S_{spa_i} \cup S_{spa_i}^*$  (i.e. the reward minus costs for the successful coalitions).*

*The probability  $PoF(S_{spa_i}^*)$  that the coalitions in  $S_{spa_i}^*$  fail while those in  $S_{spa_i} \cap S_{spa_i}^*$  succeed is*

$$PoF(S_{spa_i}^*) = \prod_{\tilde{C} \in S_{spa_i}^*} PoF(\tilde{C}) \times \prod_{\tilde{C} \in S_{spa_i} \cap S_{spa_i}^*} PoS(\tilde{C})$$

*The  $Var^{\alpha_i}(S)$ , i.e. the  $Var^\alpha$  restricted to consider only  $spa_i$  and  $S$ , is then*

$$Var^{\alpha_i}(S) = -\min_{S_{spa_i}^* \in 2^{S_{spa_i}}} \{u_i(S_{spa_i}^*) : \sum_{\substack{S'_{spa_i} \in 2^{S_{spa_i}} \\ u_i(S'_{spa_i}) \leq u_i(S_{spa_i}^*)}} PoF(S_{spa_i}^*) > \alpha_i\}$$

Having  $VaR^{\alpha_i}(S)$ , the computation of the  $TCE^{\alpha_i}(S)$  is straight-forward. Please note that  $VaR^{\alpha_i}(S)$  and thus also  $TCE^{\alpha_i}$  depend on the agent's payoff. But as becomes clear in section 5, computing a stable payoff depends on the risk. Also, we have to consider each element in the power-set of coalitions that  $spa_i$  is involved in, making the complexity of this computation exponential. However, by bounding the number of coalitions an agent might be involved in, we obtain polynomial complexity. This is also shown in section 5.

## 5 Stability of Fuzzy Coalitions Structures

In this section, we finally show how a coalition's payoff should be distributed among its members. Cooperative game theory traditionally deals with the question how this can be done in a *stable* way. Stable means that no agent has a reasonable incentive to break its coalition(s). For games with fuzzy coalitions, several such solution concepts, including the Core and the Shapley Value, have been introduced in the literature [2, 5, 9]. Unfortunately, these assume a linear or even proportional relationship of the membership and coalition values. This does not hold in our case, because the coalition either gets the payoff or not, while the membership values determine the involved risk. But even considering the expected values does not help, since (a) the execution time of a service instance is characterized by an  $\frac{1}{x}$ -relationship wrt. to the membership (see Definition 2.2(d)) and (b) the actual probability of failure also depends on the underlying distributions of the service instance runtimes which might be arbitrary. We thus introduce a new variant of the *excess* which is compliant with our setting. Since the excess is the basis for a number of solution concepts including the Core, Kernel and Nucleolus, this allows us to use these concepts. In this paper, however, we consider only the Kernel.

In crisp games, the excess of a coalition  $C$  wrt. a given coalition structure  $S$  with  $C \notin S$  quantifies the difference in payoff that the agents in  $C$  obtain by forming  $C$  and leaving their resp. coalitions in  $S$ . Because each agent can be a member of only one coalition in a crisp coalition game, they then do not obtain any payoff from their former coalitions. But this is not the case in fuzzy coalition games. Here, it is possible to withdraw just some membership and put it into a new coalition. However, not all coalitions might be feasible wrt. the involved agents' individual risk bounds. We consider such coalitions not to be a feasible threat. Also, we exclude the case that an agent threatens to withdraw any amount membership from an existing coalition such that its own risk bound would be exceeded. While this makes sure that the hard risk bounds are taken into account, we also have to consider that more membership means a better chance of success. Thus, we regard the expected coalition values.

### Definition 10 Excess of a fuzzy coalition

Let there be fuzzy coalition  $\tilde{C}$  and fuzzy coalition structures  $S$  and  $S'$  with  $\tilde{C} \in S'$ ,  $\tilde{C} \notin S$ ,  $S'$  is feasible, and  $\forall \tilde{C}' \in S', \tilde{C}' \neq \tilde{C} : \exists \tilde{C}'' \in S : \tilde{C}' \subseteq \tilde{C}''$ . Further, let there be a payoff distribution  $u$ . We define

$$\tilde{e}(\tilde{C}, S', u)_{TCE} := v_{TCE}(\tilde{C}, S') - \sum_{spa_i \in \tilde{C}} d_i(S, S')$$

with

$$\underline{v}|_{TCE}(\tilde{C}, S') = \begin{cases} \underline{v}(\tilde{C}) & \text{if } \forall spa_i \in \tilde{C} : TCE^{\alpha_i}(S' \cup \tilde{C}) \leq tTCE_i \\ 0 & \text{otherwise} \end{cases}$$

and

$$d_i = \sum_{\tilde{C}^* \in S, \tilde{C}' \in S', \tilde{C}' \subseteq \tilde{C}^*} \underline{v}(\tilde{C}') - \underline{v}(\tilde{C}^*)$$

In crisp games, for a given configuration  $(S, u)$ , the surplus of an agent  $a_i$  over another agent  $a_k$  with  $a_i, a_k \in C \in S$  is then defined as the maximum excess of all coalitions including agent  $a_i$  but without agent  $a_k$ . For games with fuzzy coalitions, however, it is possible to threaten with a number of alternative coalitions at the same time. Also, only a membership transfer from coalitions that include both  $a_i$  and  $a_k$  should be considered. Finally, we require that all membership of  $a_i$  from such coalitions is transferred.

**Definition 11** *Fuzzy coalition surplus*

Let there be a fuzzy coalition structure  $S$  and payoff distribution  $u$  and agents  $a_i$  and  $a_k$ .

1. A feasible fuzzy coalition structure  $S'$  with  $\forall \tilde{C} \in S', \tilde{C} \notin S : a_i \in \tilde{C}, a_k \notin \tilde{C}, \forall C \in S, a_k \notin C : C \in S'$  and  $\nexists \tilde{C} \in S' : a_i, a_k \in \tilde{C}$  is called an ik-fuzzy surplus structure.
2. The set of all ik-fuzzy surplus structures wrt.  $S$  is denoted  $SS_{ik}(S)$
3. The fuzzy coalition surplus of  $a_i$  over  $a_k$  is

$$\tilde{s}_{ik|TCE} := \max_{S' \in SS_{ik}(S)} \left\{ \sum_{a_i \in \tilde{C} \in S'} \tilde{e}(\tilde{C}, \tilde{u})|_{TCE} \right\}$$

To compute a fuzzy coalition surplus it is thus not only necessary to identify the best set of agents that should form alternative coalitions when excluding the other agent, but also to find the best membership values for them wrt. feasibility and the individual agent risk thresholds.

**Definition 12** Let  $Q_{ik}$  denote a set of pairs  $(sra, \mathcal{P})$  with  $\mathcal{P}$  satisfies the request from  $sra$ ,  $a_i \in SPA_{\mathcal{P}}$  and  $a_k \notin SPA_{\mathcal{P}}$ . For a feasible coalition structure  $S$ , let  $SS_{ik}(Q_{ik})$  denote the set of all ik-fuzzy surplus structures  $S'$  wrt.  $S$  such that for all pairs  $(sra, \mathcal{P}) \in Q_{ik}$  there exists  $\tilde{C} \in \tilde{C}(sra, \mathcal{P})$  with  $\tilde{C} \in S'$ . We define the function  $MaxS(Q_{ik}, S, u)$  to return  $S^* \in SS_{ik}(Q_{ik})$  such that  $\sum_{a_i \in \tilde{C} \in S^*} \tilde{e}(\tilde{C}, \tilde{u})|_{TCE}$  is maximized wrt. all other elements in  $SS_{ik}(Q_{ik})$ .

Because the service instance runtime depends on the spent resources and thus the membership values by a  $\frac{1}{x}$ -relationship (see Definition 2.2(d)),  $MaxS$  has to solve a non-linear optimization problem. The complexity to compute a fuzzy coalition surplus is thus even worse than in the crisp case, where we have exponential complexity wrt. the number of agents in the system because of the exponential number of possible coalitions and excesses. Shehory and Kraus proposed to reduce this to a polynomial complexity by limiting the maximum coalition size[10]. We achieve the same effect for the fuzzy

coalition surplus by not only bounding the number of agents in a coalition, but also the number of coalitions that an agent threatens to transfer membership to as well as the number of plans per set of agents.

**Proposition 2** *Let  $aMax \in \mathbb{N}$  be an upper bound for the number agents in a coalition and  $\tilde{C}Max \in \mathbb{N}$  be an upper bound for all sets  $|Q_{ik}|$ , i.e. the number of new coalitions including agent  $a_i$  and excluding agent  $a_k$  in the computation of  $\tilde{s}_{ik|TCE}$ . Let further  $\mathcal{P}Max$  be an upper bound for the number of plans that involve the same set of agents and let  $n \in \mathbb{N}$  be the number of agents. Then the number of sets  $Q_{ik}$ , constrained by  $\tilde{C}Max$  and  $\forall (sra, \mathcal{P}) \in Q_{ik} : \mathcal{P} \in PLANS$ , is less or equal than  $n^{(aMax \times \mathcal{P}Max)^{\tilde{C}Max}}$ .*

*Proof.* It was shown in [10] that the number of crisp coalitions with maximum size  $aMax$  among  $n$  agents is bounded by  $n^{aMax}$ . Because each set of agents might be involved in multiple plans, this has to be multiplied  $\mathcal{P}Max$  to obtain the upper bound for the number of considered coalitions. By the same argument as in the proof in [10], the number of sets of these coalitions with maximum size  $\tilde{C}Max$  is then bounded by  $n^{(aMax \times \mathcal{P}Max)^{\tilde{C}Max}}$ .

In crisp games, the *kernel* of a cooperative game  $(\mathcal{A}, v)$  with respect to a given coalition structure  $\mathcal{S}$  is a set of configurations  $(S, u)$  wherein each pair of agents  $a_i, a_k$  in each coalition  $C \in \mathcal{S}$  is in equilibrium wrt. their surpluses. That is the case if the agents cannot outweigh each other in  $(S, u)$  by having the option to get a better payoff in coalition(s) *not* in  $\mathcal{S}$  excluding the opponent agent (agent  $i$  outweighs  $k$ , if  $s_{ik} > s_{ki}$  and  $u_k > w_i(C)$ ). Fortunately, having defined the surplus also for fuzzy coalitions, we can substitute it in this definition to obtain a definition for the kernel for games with fuzzy coalitions.

**Definition 13** *Let there be a fuzzy coalition structure  $\mathcal{S}$  and payoff distribution  $u$ .  $(S, u)$  is in the kernel of the fuzzy coalition game iff each pair of agents  $a_i, a_k$  in each fuzzy coalition  $C \in \mathcal{S}$  is in equilibrium wrt. their fuzzy coalition surpluses.*

To make a payoff distribution kernel-stable for a given coalition structure, Stearns *transfer scheme* can be used in the case of crisp games. The same can be applied here, since a side-payment from one agent to another will increase the former agent's payoff while lowering the latter agent ones.

## 6 Coalition Formation Protocol RFCF

In this section, we propose a fuzzy coalition formation protocol that guarantees to form coalitions which are in compliance with the agents' individual risk bounds. The negotiation is to be finished in a fixed amount of time in order to ensure a timely start service executions. In order to achieve polynomial complexity in the negotiation, some compromises have to be made. In particular, upper bounds for the risk of a coalition structure can be obtained by either considering only the self-values of the agents instead the actual utilities or by computing the risk for subsets of the structure and utilizing the

subadditivity of TCE. The main drawback of using upper bounds for the risk is that it might prevent the formation of some coalitions which are then considered too risky although they are acceptable. We thus propose to execute a parallel process to continually improve the bound as long as there is time.

Before we give the actual definition of RFCF, we here provide a short outline of the protocol to emphasize the main ideas of the individual steps. In RFCF, each agent performs multiple tasks in parallel:

- **Composition Planning** - Composition plans are generated. Since only agents that can execute a plan together will form coalitions, this step is necessary to identify possibly worthwhile coalitions.
- **Coalition Negotiation**
  1. *Proposal generation* - The agent computes fuzzy coalitions such that their formation certainly leads to a feasible coalition structure while minimising the membership values. This way, no more membership (i.e. resources) than necessary is used, allowing the involved agents to possibly form additional coalitions later. A proposal is then send to the agents of the fuzzy coalition which maximises the value per membership.
  2. *Proposal evaluation* - From the received proposals, form feasible coalitions with acceptable risk the and maximal value per membership
  3. *Payoff distribution and risk bound update* - Use the transfer scheme to compute the Kernel-stable payoff distribution. Compute the single-coalition TCE and add it to previous coalition structure TCE bound to obtain an updated bound on the coalition structure TCE.
- **Risk Measure Computation** - Compute TCE for a new random subset of coalitions to obtain a tighter bound for the coalition structure TCE.

In the following definition of the algorithm, we use the following functions and constants:

- $\mathcal{P}Max$ : the maximum number of plans to be considered for a set of agents
- $aMax$ : the maximum coalition size
- $\tilde{C}Max$ : the maximum number of coalitions that an agent threatens to transfer membership to in the surplus computation
- $sra(\mathcal{P})$  Returns the service request agent for whose request  $\mathcal{P}$  was generated.
- $findFuzzyCoalition(S, \mathcal{P}, risk)$ : Computes a fuzzy coalition  $\tilde{C}$  such that the membership degrees in  $\tilde{C}$  are minimized while  $S \cup \tilde{C}$  is acceptable for all agents wrt. risk. Use  $\tilde{C}(sra(\mathcal{P}), \mathcal{P})$  as a starting point. If  $risk = nil$  then compute an upper bound for  $TCE^{\alpha_a}(S \cup \tilde{C}(sra(\mathcal{P}), \mathcal{P}))$ , otherwise use  $risk$  as this upper bound. It is possible to efficiently implement this function by exploiting the monotonicity of the TCE wrt. to the membership values. If this is not possible or  $|\tilde{C}| > MaxCSize$ , return  $nil$
- $makeStable(S)$ : Computes a new stable payoff distribution  $u^*$  for the fuzzy coalition structure  $S$  using the transfer scheme (see 5) and the bounds  $\mathcal{P}Max$ ,  $aMax$  and  $\tilde{C}Max$ .

**Algorithm 1 RFCF**

*Each agent a performs:*

*Initialization:*

1. *set*  $PLANS := \emptyset$
2. *set*  $PPPLANS := \emptyset$
3. *set*  $PPPLANSRISK := \emptyset$
4. *set*  $PROPS :=$  *new priority queue*
5. *set*  $risk_a := TCE(\{a\}/1)$

*Parallel Execution:*

- *Composition plan generation: repeat (until terminated)*
  1. *Generate a new composition plan*  $\mathcal{P}$  *for a random service request and for a set of agents for which the number of previously generated plans is less than*  $\mathcal{P}Max$ .
  2.  $PLANS := PLANS \cup \mathcal{P}$
- *Coalition negotiation: repeat (until terminated)*
  1. *Proposal generation*
    - (a) *set*  $BestCoalition := nil$ ,  $BestPayoffperMembership := 0$
    - (b) *for each*  $\mathcal{P}$  *in*  $PLANS$  *do:*
      - i.  $\tilde{C} := findFuzzyCoalition(S, \mathcal{P}, nil)$
      - ii. *if*  $\tilde{C} = nil$  *then*  $PLANS := PLANS \setminus \mathcal{P}$ ;  
 $POSTPONEDPLANS := \cup \mathcal{P}$  ;*next*  $1b$
      - iii. *if*  $v(\tilde{C})/|\tilde{C}| > BestPayoffperMembership$  *then*  
 $PLANS := PLANS \setminus \mathcal{P}$ ;  $BestCoalition := \tilde{C}$ ;  
 $BestPayoffperMembership := |\tilde{C}|$
    - (c) *if*  $BestCoalition = nil$  *then for each*  $\mathcal{P}$  *in*  $POSTPONEDPLANS$  *do:*
      - i. *if*  $PPPLANSRISK$  *contains*  $(\mathcal{P}, .)$  *then*  
 $\tilde{C} := findFuzzyCoalition(S, \mathcal{P}, PPPLANSRISK(\mathcal{P}))$
      - ii. *if*  $\tilde{C} = nil$  *then next*  $1b$
      - iii. *if*  $v(\tilde{C})/|\tilde{C}| > BestPayoffperMembership$  *then*  
 $PPPLANSRISK := PPPLANSRISK \setminus \mathcal{P}$ ;  
 $BestCoalition := \tilde{C}$ ;  $BestPayoffperMembership := |\tilde{C}|$
  2. *send*  $(BestCoalition, BestPayoffperMembership)$  *as a proposal to all other agents*
  3. *Proposal evaluation*
    - (a) *receive coalition proposals from all other agents and self*
    - (b) *for each non-nil proposal*  $(\tilde{C}, ppm)$ , *put*  $\tilde{C}$  *in*  $PROPS$  *with priority*  $ppm$ .
    - (c) *set*  $S^* = \emptyset$
    - (d) *while*  $PROPS$  *is not empty do*
      - i. *get and remove the highest priority coalition*  $\tilde{C}$  *from*  $PROPS$
      - ii. *if*  $\tilde{C}$  *is feasible, set*  $S^* := S^* \cup \tilde{C}$
  4. *Payoff distribution and TCE update*
    - (a) *set*  $u^* = makeStable(S \cup S^*)$
    - (b) *do atomically: set*  $S := S \cup S^*$  *and*  $u := u^*$
    - (c) *set*  $risk_a := risk_a + \sum_{\tilde{C} \in S_a^*} (TCE_a(\tilde{C}))$



- Risk measure computation of current structure: repeat (until terminated)
  1. randomly choose a previously unconsidered subset  $S^*$  from  $S_a$
  2.  $risk_a := risk_a - \sum_{\tilde{C} \in S^*} TCE_a(\tilde{C}) + TCE_a(S^*)$
- Risk measure computation of potential structures for postponed plans: repeat (until terminated)
  1. Randomly choose  $\mathcal{P}$  from PPPLANS such that  $(\mathcal{P}, \cdot) \notin PPPLANSRISK$
  2. Compute exact  $TCE^{\alpha_a}(S \cup \tilde{C}(sra(\mathcal{P}), \mathcal{P}))$  and put  $(\mathcal{P}, TCE^{\alpha_a}(S \cup \tilde{C}(sra(\mathcal{P}), \mathcal{P})))$  into PPPLANSRISK
- Termination of negotiation
  1. Wait(ExecutionStartTime)
  2. terminate all other tasks
  3. start service instance execution in my coalitions; terminate

**Proposition 3** *The runtime of the coalition negotiation section of the RFCF is polynomial.*

*Proof.* In the proposal evaluation, each agent orders the coalition proposals in the same way in the priority queue since the priority is defined as payoff per membership which is a global measure. Because of the bounds used in the surplus computation, the payoff distribution is done in polynomial time (see 5). All other steps in the coalition negotiation section are of less complexity.

## 7 Related Work

In the research field of fuzzy coalition formation, Nishizaki and Sakawa in [9] proposed a number of algorithms to compute solutions according to their concepts. They did however not propose a protocol that enables a coalition negotiation among computational autonomous agents. Also, as we have pointed out in section 1, they assume that the coalition value is a proportional function of the agents' membership degrees, which does not hold in our case.

Shehory and Kraus considered the formation of overlapping but non-fuzzy coalitions. They however focus on maximising the joint payoff of all agents rather than individual payoffs or minimising potential individual losses. In contrast, our approach focuses especially on the latter points. Thus, the motivations and the properties of the obtained solutions are very different.

There also exist approaches for the formation of non-overlapping coalitions which take uncertainty in the coalition values into account. These are also suitable to tackle the problem of reduced coalition values due to (partial) coalition failure in some cases. Probabilistic approaches, such as [6], usually consider the expected values of coalitions. This might lead to the case that a number of risk-neutral agents decide to form a high-risk coalition, excluding risk-averse agents to cooperate with them because overlapping coalitions are not allowed. In contrast, our approach allows for such cooperations by forming additional coalitions. Approaches that employ fuzzy coalition values, such as [3], account for a range of possible coalition values. However, the fuzzy coalition values are assumed to actually be fuzzy numbers or intervals. But this assumption is not compatible with our setting where a coalition value either produces a specific profit or a specific loss.

## 8 Conclusions

We have studied a setting of cooperative service provider agents that form fuzzy coalitions in order to share and combine resources and services to efficiently respond to market demands while bounding individual risk. We showed how a coherent risk measure, the TCE, can be used to assess the risk for agents when taking part in coalitions to satisfy service requests with deadlines. By splitting resources among different coalitions, an agent might lower its overall risk. Despite previous work on fuzzy coalitions in the literature, we found it necessary to give our own definitions for the fuzzy coalition game, including the excess and surplus for fuzzy coalitions. This is because of unrealistic assumptions in the cited models that do not hold in our setting. In the surplus computation, sets of alternative fuzzy coalitions have to be considered. As a consequence, we had to bound not only the maximum coalition size, but also the number of coalitions in these sets as well as the number of plans for a set of agents to obtain a polynomial computation time for the fuzzy coalition surplus.

## References

1. P. Artzner, F. Delbaen, S. Eber, and D. Heath. Coherent measures of risk. *Mathematical Finance*, pages 203–228, 1999.
2. J.-P. Aubin. *Mathematical Methods of Game and Economic Theory*. North-Holland, 1979.
3. B. Blankenburg, M. Klusch, and O. Shehory. Fuzzy kernel-stable coalitions between rational agents. In *Proc. 2<sup>nd</sup> Int. Conference on Autonomous Agents and Multiagent Systems, Melbourne, Australia, 2003*.
4. R. N. Bracewell. *The Fourier Transform and Its Applications*. McGraw-Hill Science/Engineering/Math, New York, 3rd edition, 1999.
5. D. Butnariu. Stability and shapley value for an n-persons fuzzy game. *Fuzzy Sets and Systems*, 4:63–72, 1980.
6. G. Chalkiadakis and C. Boutilier. Bayesian reinforcement learning for coalition formation under uncertainty. In *Proc. 3<sup>rd</sup> Int. Conference on Autonomous Agents and Multiagent Systems, New York, USA, 2004*.
7. S. Cheng, Y. Liu, and S. Wang. Progress in risk management. *Advanced Modelling and Optimization*, 6(1):1–20, 2004.
8. G. R. Grimmett and D. R. Stirzaker. *Probability and Random Processes*. Oxford University Press, 3rd edition, 2001.
9. I. Nishizaki and M. Sakawa. *Masatoshi Fuzzy and multiobjective games for conflict resolution*, volume 64 of *Studies in Fuzziness and Soft Computing*. Physica-Verlag, Heidelberg, 2001.
10. O. Shehory and S. Kraus. Feasible formation of coalitions among autonomous agents. *Computational Intelligence*, 15(3):218–251, 1999.
11. R. E. Stearns. Convergent transfer schemes for n-person games. *Transactions of the American Mathematical Society*, 134:449–459, 1968.

---

## Dynamic Coalition Forming

M. Klusch and A. Gerber: Dynamic Coalition Formation among Rational Agents. IEEE Intelligent Systems, 17(3), pages 42 - 47, IEEE CS Press, May/June 2002.

# Dynamic Coalition Formation among Rational Agents

Matthias Klusch and Andreas Gerber, *German Research Centre for Artificial Intelligence*

**D**esigning self-interested autonomous software agents that can negotiate rationally in stable coalitions can dramatically benefit end users. Rational agents are usually required to form beneficial coalitions in open, distributed, and heterogeneous environments, including scenarios in which dynamically occurring events might interfere

with the coalition processes. Dynamic coalition formation (DCF) methods promise to be particularly well suited for applications of ubiquitous and mobile computing, including mobile commerce in wireless environments.

In mobile-commerce settings, for example, personalized information agents, each representing a potential business partner, might dynamically form temporary profit-oriented coalitions to enhance a customer's purchasing and negotiating strategies in multiple electronic marketplaces. This vision for the common Internet user isn't far from being realized, especially with recent advances in wireless computing and communication appliances.<sup>1-3</sup>

## Static formation of stable coalitions

We categorize coalition formation into two approaches: utility-based and complementary-based. These two models divide the actors into those that follow the principle of *bellum omnium contra omnes* as it is largely favored, for example, by game theory, and those that rely on the collaborative use of complementary individual skills to enhance the power of each agent to accomplish its goals.<sup>4,5</sup> Until now, most classic methods for forming stable coalitions among rational agents follow the utility-based approach and rely on derived concepts from cooperative game theory, economics, and operations research. Utilitarian coalition formation covers two main activities: generating coalition structures and distributing gained benefit among the participants of each coalition.<sup>6,7</sup>

We define a *cooperative game* as a set  $A$  of agents in which each subset of  $A$  is called a coalition and a

characteristic function  $v$  assigns each coalition  $C$  in  $A$  its maximum gain. (We offer an introduction to cooperative game theory here, but other sources provide a more in-depth explanation.<sup>8,9</sup>) The value  $v(C)$  does not depend on the actions of agents outside the coalition. Any coalition  $C$  forms by a binding agreement on the distribution of its coalition value  $v(C)$  among its members.

The solution of a cooperative game with side payments is a *coalition configuration*, which consists of a partition  $S$  of  $A$ , the coalition structure, and an  $n$ -dimensional payoff distribution vector in which components are computed by a utility function  $u$ . The payoff distribution assigns each agent in  $A$  its utility  $u(a)$  out of the value  $v(C)$  of the coalition  $C$  in a given coalition structure  $S$ . The number or size of coalitions formed using a coalition formation method is often restricted to ensure, for example, polynomial complexity of the formation process.

In coalition configurations with so-called pareto-optimal payoff distributions, no agent is better off in any other valid payoff distribution for the given game and coalition structure. A coalition configuration  $(S, u)$  is stable if no agent has an incentive to leave its coalition in  $S$  due to its assigned payoff  $u(a)$ . Different characteristics and stability criterion define different solution spaces for cooperative games. Rational agents involved in a cooperative game  $(A, v)$  negotiate a stable payment configuration  $(S, u)$  as a solution that uses an appropriate coalition algorithm (CA). Each agent can execute the CA locally. Negotiation according to the CA is completely decentralized. The CA provides a stable coal-

*Dynamic coalition formation (DCF) promises to be well suited for applications of ubiquitous and mobile computing. This article proposes a simulation-based DCF scheme designed to let rational agents form coalitions in dynamic environments.*

tion configuration for any cooperative game at any time.

A coalition formation environment for a given set of agents  $A$  is the set of assumptions and constraints that are valid for any kind of coalition-forming activity between agents in  $A$ , including propositions on the task-related functionality of each individual agent in  $A$ , including its set of tasks, goals, actions, and methods to compute the individual utilities of task-related productions. The set of assumptions also includes valid methods for computing the values of coalitions and valid methods for determining coalition configurations, including methods for searching coalition structures.

In a given coalition formation environment, the agents agree on what kind of stable coalitions will be negotiated and what particular CA will be used for the negotiation. Different agents can compute their utilities of task execution and corresponding productions differently. However, most work on coalition formation relies on coalition formation environments in which all agents are homogeneous.

A coalition formation environment is super-additive or subadditive, depending on the type of all cooperative games it allows. In subadditive games, at least one pair of potential coalitions is not better off by merging into one. We define a coalition formation model by both the considered coalition formation environment and a given CA for this environment.

### Stable and static coalitions

The meaning of coalition stability depends on the considered discipline and application domain. Many (if not most) of the coalition formation algorithms rely on chosen game theory concepts for stable payoff division within coalitions according to, for example, the Shapley value, the core, the bargaining set, or the kernel.<sup>8</sup> All traditional approaches to coalition formation remain static in the sense that they do not allow for any type of interference with the running coalition formation process. In addition, known results for superadditive coalition formation environments must be transported into general or subadditive environments to gain practical relevance for the development and application of DCF algorithms to real-world open environments.

### Core-stable coalitions

One approach<sup>10</sup> to form stable coalition configurations consists of the following two

steps: searching for a coalition structure in a corresponding coalition structure graph for the given game  $(A, v)$  and then computing its payoff according to the stability concept of the core. We define the core of a game with respect to a given coalition structure as the set of coalition configurations that don't necessarily have unique payoff distributions. Only coalition structures that maximize the social welfare—the sum of all coalition values of coalitions in the considered structure—are core-stable.

However, searching for an optimal coalition structure (given a set  $A$  of agents) among the exponential number of  $|A|^{A/2}$  possible coalition structures is computationally difficult, because we have to try at least  $2^{|A|-1}$

In a given coalition formation environment, the agents agree on what kind of stable coalitions will be negotiated and what particular CA will be used for the negotiation.

coalition structures.<sup>10,11</sup> Another well-known problem with core-stable configurations is that the core might be empty for certain cooperative games, and is exponentially difficult to compute. Because of these problems, using the core-stable coalition is quite unpopular. In fact, we aren't aware of any CA for computing core-stable coalition configurations in DCF environments.

### Shapley-value-stable coalitions

Any payoff division scheme according to the Shapley value provides an agent with the added value that it brings to the given coalition structure, averaged over all possible joining orders. That makes the Shapley value exponentially hard to compute. Algorithms for forming stable coalitions that rely on the stability concept of the Shapley value, and a variation of it, the bilateral Shapley value<sup>12</sup> applied to arbitrary cooperative games, are proposed elsewhere.<sup>13</sup> Computing a proposed payoff division according to the bilateral Shapley value with equal or proportional his-

tory-based share among coalition members is efficient and rational for superadditive games. Because this does not necessarily hold for subadditive games, these algorithms are not suitable for DCF.

### Kernel-stable coalitions

The kernel of a cooperative game is the set of kernel-stable configurations  $(S, u)$  in which all coalitions in  $S$  are in equilibrium. Coalition  $C$  is in such an equilibrium if each pair of agents in  $C$  is in equilibrium—that is, if any pair of agents in  $C$  is balanced so that none of both agents can outweigh the other in  $(S, u)$  by having the option to get a better payoff. Each agent has to compare its surplus with those of other agents.

A game's kernel is exponentially hard to compute unless a constant limits the coalition's size. The kernel appears to be attractive because it is unique for any three-agent game, it assigns symmetric agents of some coalition in a given coalition structure for equal payoff, and it is locally Pareto-optimal. Polynomial CAs for polynomial kernel-stable coalition configurations have been developed and applied to the domain of cooperative information agents.<sup>14,15</sup>

### Fuzzy cooperative games

Negotiation during the coalition-forming process might be uncertain. Such uncertainties could be caused by the possibility of nondeterministic events that hamper the negotiation process and produce incomplete information. Agents might have uncertain knowledge about the share of coalition income in which they intend to participate or on the degree of their membership in one or multiple coalitions. An agent might determine the degree of its membership to potential coalitions by individually leveled commitments to other agents or bargains that indicate the degree of collaboration that the agents desire. The first case might imply the formation of fuzzy-valued coalitions, whereas the second case might induce the formation of fuzzy coalitions, which might partially overlap.

A fuzzy cooperative game<sup>16</sup> consists of a set of agents, a fuzzy characteristic function  $v$ , and the membership function  $m$  of the fuzzy quantities  $v(C)$  that might be interpreted as expectation of the common coalition profit that is to be distributed among its members. That is, the worth  $v(C)$  of a fuzzy-valued coalition  $C$  is a fuzzy set of its possible real-valued coalitional profits. This set of fuzzy quantity

$v(C)$  has at least one modal value determined by the membership function  $m$ . If, for a given fuzzy cooperative game, the coalition value  $v(C)$  is equal to one modal value of  $C$  for all possible coalitions  $C$ , it is equivalent to a (deterministic) cooperative game.

### Stochastic cooperative games

Another class of cooperative games arises from cooperative decision-making problems in stochastic environments. A game with stochastic payoffs<sup>17</sup> is defined by a set of agents, a set of possible actions coalitions might take, and a function assigning to each action of a coalition a real-valued stochastic variable with finite expectation, representing the payoff to a coalition when this particular action is taken. Thus, in contrast to a deterministic cooperative game, the payoffs can be random variables, and the actions a coalition can choose from are explicitly modeled, because the payoffs are not uniquely determined.

### Developing DCF Schemes

We define the DCF research domain as the set of cooperation methods, schemes, and enabling technologies designed to cope with the problem of dynamically building beneficial coalitions among agents in open, distributed, and heterogeneous environments. The DCF problem must be solved in any collaboration environment and scenario in which agents enter or leave coalition formation processes and in which the set of tasks that individual agents must accomplish change dynamically. Cooperation scenarios inducing uncertain, time-limited, context-based utilities and coalition values can exacerbate the DCF problem.

One challenge is to get agents to react to different kinds of changes in real time without having to restart the complete negotiation process. Doing so requires agents to handle uncertain environment knowledge through appropriate adaptation mechanisms. Another research challenge concerns the transformation of the traditional game-theory criterion of coalition stability to these dynamic environments. It hardly makes sense for an agent to determine stable coalitions according to, for example, the Shapley value or the kernel in a frequently changing environment.

Basic research must clarify which kinds of dynamic settings, and to what extent available algorithms for the static formation of stable coalitions, should be adopted. In particular, the developed methods must let agents deliberately restart their coalition

negotiations at any time depending on the environmental changes. There is a trade-off between the efficiency of DCF and the quality of computed stable coalition configurations. Though this trade-off seems intuitively clear, it must be investigated further.

The development of DCF schemes might benefit from adopting appropriate methods for quantitative or qualitative decision-making that is based on partial, uncertain, and tentative information. Reasonable solutions for fuzzy and stochastic cooperative games might be adopted for cooperation schemes that let agents deal with different uncertainty types. Such uncertainties might be induced in DCF environments by, for example, network faults, changes of trust, or the receipt of vague or

Basic research must clarify which kinds of dynamic settings, and to what extent available algorithms for the static formation of stable coalitions, should be adopted.

even incomplete data during task execution.

There are no CAs for fuzzy or stochastic coalitions available to date. Developing such CAs for DCF environments appears to be even more challenging. To our knowledge, no such work is available to date. Other relevant work for developing cooperation schemes for dynamic environments include, for example, utility-based schemes for dynamically reorganizing organizational structures and exception-tolerant reasoning and multicriteria decision-making.

Social-reasoning mechanisms are essential building blocks suitable to situations in which agents might dynamically enter or leave the society without any global control. Advances in social reasoning have a clear impact on developing DCF schemes. Social-reasoning mechanisms are often based on the notion of social dependence or aim at reputation and trust management. To acquire and use dependence knowledge on the considered agent society, each agent must explicitly represent some properties of the other

agents, exploit this representation and optimize its behavior according to the evolution of the society, and monitor and revise its representation to avoid inconsistencies.

Reputation management aims at avoiding interaction with undesirable participants and might complement other security technologies for authentication and authorization. Mechanisms for building, propagating, measuring, and maintaining reputation and trust are useful to apply to settings for coalition formation among self-interested agents in e-commerce applications in which trusted third parties are required but not available. Merging several individual trust matrices, which are commonly used as a means to assess trust relationships among agents, requires further research. In general, mechanisms that let agents react on frequent changes of reputation ratings and assessment of trustworthiness of potential coalition partners are rare. Rational agents might face many potentially beneficial choices related to the timing of events that might occur during the individual decision process and the negotiation with other potential coalition partners. The preliminary results and experiences reported in relevant work might help design more complex methods for customer coalition formation in real time.

### A simulation-based scheme

We designed the DCF-S scheme to help agents react to changes in their set of goals and in the agent society. We can instantiate the DCF-S scheme using different computational methods and negotiation protocols. Each instantiation yields a particular DCF-S-based CA. The development, implementation, and experimental evaluation of such DCF-S algorithms is part of our ongoing research efforts.

### Environment

For our research, we assume a coalition formation environment in which agents continuously receive a set of goals from their users or other agents. Furthermore, any agent can freely enter or leave the society at any time. Each agent must use appropriate mechanisms to cope with the uncertainties and gradually adapt its decision-making techniques to changes in the environment. We assume that each agent is equipped with an appropriate learning component for this purpose.

We also assume an additional set of special agents, called *world-utility agents*. Any WUA might receive, compile, and maintain information about each of its registered agents.

## DCF-S Scheme Functions, Numbers, and Sets

The local knowledge base of an agent  $a$  consists of  $GS(a)$ , the set of goals the agent must accomplish. Interleaved goals are aggregated by  $a$  into one goal. The list  $CLIG$  (*BestCLIG*) contains the candidates with which agent  $a$  might coalesce to accomplish a goal  $G$  in  $GS(a)$ . The list  $ACLIG$  of agent information records contains information about agent  $a$  on the capabilities of other agents  $a'$  with respect to  $G$ . Each record stores a finite-dimensional vector of real-valued attributes of an agent  $a'$  with respect to its estimated value of contribution to the accomplishment of goal  $G$  in  $GS(a)$ . Goal-related attributes of agent  $a'$  concern, for example, the estimated amount of its available resources, costs, quality, and efficiency with respect to goal  $G$ .

Other attributes of  $a'$  concern its reliability and trustworthiness in cooperation. The attributes include the following:

- The real value  $crv(a', ACLIG, C)$  in  $[0,1]$  denotes the risk of agent  $a$  to cooperate with agent  $a'$  in coalition  $C$  for goal  $G$  in  $GS(a)$  with respect to the information on  $a'$  in the list  $ACLIG$ .
- The real value  $crl(a', C)$  denotes the worst acceptable risk of agent  $a$  to cooperate with agent  $a'$  in coalition  $C$ .
- The real value  $rrl(a', C)$  in  $[0,1]$  denotes the worst acceptable risk of agent  $a$  to remove agent  $a'$  from valid coalition  $C$  the agent  $a$  is leading. This risk value might be computed with respect to the implied payment of trust penalty  $tp(a', a)$  and the penalty payment limit  $ppl(a)$ .
- The real value  $tp(a', a)$  is the trust penalty to be paid by  $a$  to  $a'$  in case  $a$  breaks a coalition agreement with  $a'$ .
- The real value  $ppl(a)$  denotes the upper limit of penalty payments by agent  $a$ .

In addition, we define the following values, sets, and functions:

- The integer value  $MaxSim$  denotes the maximum number of steps for each simulation round.
- $CCLIG$  is the set of candidates for forming a coalition with respect to goal  $G$ . These candidates come from the current set  $CS$  of valid coalitions and determined by the function  $Match$ .
- $Match(CS, G, ACLIG)$  determines the set of agents in the set  $CS$  of all trusted agents (individual agents and valid coalitions which are actually known to agent  $a$ ), each of

which is capable of contributing to the accomplishment of goal  $G$ . The capability-based matching determines to what degree the agents' capability descriptions in  $ACLIG$  match the description of the goal  $G$ .

- $Request(ACLIG, WUA)$  and  $Update\_Agt\_Information(ACLIG, RecentAC)$  concern the request of the nearest world-utility agent for information to (periodically) update the list  $ACLIG$ . The update relies, in particular, on an appropriate learning mechanism to approximate incomplete or vague information.
- $SelectAgt\_MinRisk\_MaxValue(CC, HCIG, ACLIG)$  returns an agent  $a'$  from the set  $CC$  of agents with estimated minimum risk of cooperation in, and maximum value of contribution to, the (simulated) joint coalition  $HC \cup \{a'\}$  with respect to goal  $G$  regarding the attributes of  $a'$  stored in the agent information list  $ACLIG$ . Only agents  $a'$  are selected which payoff in  $HC \cup \{a'\}$  is individual rational.
- $SelectAgt\_MaxRisk\_MaxValue(HCIG, ACLIG)$  returns an agent  $a'$  from the coalition  $HCIG$  with estimated maximum risk of cooperation in, and maximum value of, the coalition  $C \cup \{a'\}$  regarding the information on  $a'$  in  $ACLIG$ .
- $Events(BestCLIG)$  returns the set of events that have occurred, influencing the coalition's formation, which consists of all agents in the list  $BestCLIG$ .
- $Value(CLIG)$  determines the value  $v(C)$  of the coalition  $C$ , which consists of all agents in the list  $CLIG$ .
- $BilateralNegotiation(a, a', Value(BestCLIG), C \cup \{a'\})$  returns true if the bilateral multi-attribute negotiation of agent  $a$  with agent  $a'$  on a joint coalition  $C \cup \{a'\}$  with respect to its value is successful; otherwise, it returns false.
- $Evaluate(ACLIG)$  updates the agent information list  $ACLIG$  according to the local evaluation of the recent negotiation processes of the agent and returns the updated list  $ACLIG$ . This evaluation gives input to the agent's internal learning mechanism for adapting to changes in its environment.
- $StopNegotiation(BestCLIG)$  stops all running negotiation processes with all agents in  $BestCLIG$  on a coalition for the goal  $G$  and updates the list  $CLIG$  by keeping those agents with which the agent has already successfully negotiated.
- $RedundancyCheck(BestCLIG)$  returns a nonredundant list  $BestCLIG$ .

This information includes statements on an individual agent's problem-solving capability and the evaluation of its quality of service by other agents. Such evaluation might concern an agent's reliability and trustworthiness and affect its cooperation with other agents. These evaluation records are safe against possible manipulation and are securely distributed to and updated by the networked WUAs. However, each agent in the considered agent society is free to request its nearest WUA to obtain information on its environment.

We define a goal-oriented cooperative game  $(A, v)|G$  as a cooperative game with respect to a given goal  $G$ . Such a game is determined by a given set  $A$  of agents and a real-valued function  $v$  assigning each coalition  $C$  its total expected outcome with respect to the accom-

plishment of the goal  $G$ . In particular, computing the individual utility of the set of productions of coalition members in  $C$  is restricted to the set of productions related to  $G$ .

We can represent any coalition to the outside world with an appropriate coalition-leading agent (CLA). We consider each coalition to be one entity or agent. Because we can consider one agent to be a single-agent coalition, we can use the terms "agent" and "coalition" interchangeably. Initially, the set of all possible, nonempty coalitions is the set of single-agent coalitions. Each agent is a CLA of a stable coalition for accomplishing one of its goals as a solution of the corresponding game. Any CLA is supposed to act on behalf of the members of its coalition, including negotiating and controlling the dis-

tribution of resources and payoffs among the coalition members according to the coalition contract. This structure is similar to the structure of holonic multiagent systems.

### DCF-S scheme

In the DCF-S scheme, each CLA concurrently simulates, selects, and negotiates coalitions, each of which is able to accomplish one of its goals with an acceptable ratio between estimated risk of failure and individual profit. Figure 1 outlines the scheme in pseudocode, and the "DCF-S Scheme Functions, Numbers, and Sets" sidebar defines the necessary concepts. We can summarize the main steps of the DCF-S scheme executed by each CLA as follows: preparation, simulation, negotiation, and evaluation.



```

for each G in GS(a) do concurrently until external termination
{ halt:= false;
while not halt do
{

Preparation
CCIG = ∅; CLIG, LastCLIG, BestCLIG := null; op:=""; z, penalties:=0;
if periodic(date, ACLIG) then
{RecentAC := Request(ACLIG, WUA); ACLIG := Update_Agt_Information(ACLIG, RecentAC);}
CCIG := Match(CS, G, ACLIG); HCIG:=CLIG;

Simulation
for z:= 1 to MaxSim do
{ op := Random({noop, add_agent, remove_agent}); LastCLIG:= CLIG;
if op = add_agent then
{ agt:= Select_Agt_MinRisk_MaxValue(CCIG, HCIG, ACLIG);
if crv(agt, ACLIG, HCIG) ≤ crl(agt, HCIG) then{CLIG:= CLIG + (agt, add); HCIG:=HCIG∪{agt};}
else
if op = remove_agent then
{ agt:= Select_Agt_MaxRisk_MaxValue(HCIG, ACLIG);
if crv(agt, ACLIG, HCIG) > rrl(agt, HCIG) then { CLIG := CLIG + (agt, remove); HCIG:= HCIG\{agt}; penalties:=
penalties + tp(agt, a)
}
}
}
if Value(CLIG) > Value(LastCLIG) then BestCLIG:= RedundancyCheck(CLIG);
if Value(BestCLIG) >> v(CIG) && penalties < ppl(a) && Events(BestCLIG) =∅ then halt:= true;
}

Negotiation
for each (a', op) in BestCLIG do concurrently
{ try
if op = add then {if BilateralNegotiation(a, a', Value(BestCLIG), CIG∪{a'}) then CIG:= CIG ∪ {a'}}
else { CIG:= CIG \ {a'}; penalty_payment( tp(a', a), a'); }
catch(event: if Events(BestCLIG) <>∅ then { StopNegotiation(BestCLIG); Goto (Evaluation) });
}

Evaluation
EvalRes:=Evaluate(ACLIG); [if desired the Send(EvalRes, WUA);];
}

```

Figure 1. The DCF-S scheme in pseudocode. Each coalition-leading agent *a* executes the steps illustrated here, where CIG is a valid coalition led by *a* for one of its goals *G* in *GS(a)*.

In the preparation phase, the CLA determines the set of goals to be accomplished in cooperation with other agents and periodically updates its knowledge of the environment. The local knowledge base includes information on the partially known problem-solving capabilities of other agents as well as individual evaluations of past collaborations with these agents. To obtain this information, the CLA might request its nearest WUA. Because this environment knowledge might be incomplete or vague, the agent uses appropriate learning mechanisms for approx-

imating the needed information.

In the simulation phase, the CLA simulates the formation of coalitions, each of which might be able to accomplish a given goal with an acceptable ratio between the estimated individual profit and risk of forming the coalition.

In the negotiation phase, the CLA negotiates all coalitions it has determined in the previous simulation step. The CLA negotiates each goal-oriented coalition bilaterally with each potential member of the coalition. The complete set of negotiation sequences

can be performed concurrently. The result of a successful negotiation is a binding agreement between agents on the constraints and attributes of their cooperation in the new coalition.

In the case that one bilateral negotiation fails or an event changing the value or structure of the considered coalition is detected, the negotiation process for that coalition is immediately halted. The CLA then evaluates the negotiation process for this coalition and restarts the simulation of potential coalitions for the particular goal. For the restart, it keeps those agents in its coalitions with which it has already successfully negotiated and considers the current situation of the environment. This way, the CLA might avoid a complete restart, thereby avoiding possible penalty payments for removing agents from valid coalitions and a corresponding decrease of its reliability.

In the evaluation phase, the CLA evaluates its recent negotiations and reports these evaluations to the nearest WUA for distribution. Concurrently, it controls the distribution of payoffs and resources to members of the newly formed coalitions according to the successfully negotiated contracts.

### Discussion of the scheme

In the DCF-S scheme, each agent simulates, selects, and negotiates coalitions, each of which is able to accomplish one of its goals with an acceptable ratio between estimated risk of failure and individual profit. In other words, the agents strive to solve a set of single goal-oriented cooperative games  $(A, v) \setminus G$  by forming potentially overlapping coalitions with stable payoff distributions. Each of these goal-oriented cooperative games might change at any time subject to different kinds of nondeterministically occurring events such as agents leaving or entering the society. Each detected change can induce new cooperative games for the agents to solve.

According to the DCF-S scheme, each CLA reacts to these changes through a partial rather than complete restart. The agent tries to keep those agents in the affected coalitions with which it has already successfully reached a coalition agreement. However, the DCF-S scheme does not guarantee in general an optimal solution to these games. Rather, the agents continuously approximate the best solutions given their current knowledge of the dynamic environment. A different, but similarly opportunistic and high-risk DCF scheme, has been proposed elsewhere.<sup>11</sup>



The DCF-S scheme assumes the existence of a set of networked WUAs, which each agent in the society is free to contact for obtaining needed information on the environment. In addition, the update of local knowledge by an agent is assumed to use results from a continuous adaptation process to approximate the needed information on its environment. That might improve the quality of its decision-making independent from the WUAs, and thereby reduce the overall complexity in computation and communication.

The complexity of any DCF-S based CA as an instantiation of the DCF-S scheme largely depends on the complexity of the implemented methods that the designer chooses for capability-based matching, learning, selection, and negotiation. For example, for low-complexity computation of stable payoff distribution in superadditive environments, we propose adopting the distribution according to the bilateral Shapley value with equal or proportional share among coalition members.<sup>15</sup>


**A**pplication-specific instantiations of the DCF-S scheme lead to the development of different DCF-S-based CAs. For this purpose, relevant approaches and theoretical work stemming from different disciplines are available, including work on temporal social reasoning, machine learning, and fuzzy and stochastic cooperative games.

DCF algorithms promise to be particularly well suited for applications of ubiquitous and mobile computing, including mobile commerce in wireless network environments. However, further basic research is needed to investigate the potential of the new research field of DCF, which remains in its infancy. ■


## References

1. M. Tsvetov and K. Sycara, "Customer Coalitions in the Electronic Marketplace," *Proc. 4th Int'l Conf. Autonomous Agents*, ACM Press, New York, 2000, pp. 263–264.

The Authors



**Matthias Klusch** is a senior researcher in the Multiagent Systems Research Group at the German Research Center for Artificial Intelligence. He is also assistant professor in the department for Artificial Intelligence at the Free University of Amsterdam, Netherlands. His research interests include the application of AI and agent technology to databases and intelligent information systems. He received a PhD in computer science from the University of Kiel, Germany. Contact him at [DFKI GmbH](mailto:dfki.de), Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany; [klusch@dfki.de](mailto:klusch@dfki.de); [www.dfki.de/~klusch](http://www.dfki.de/~klusch).



**Andreas Gerber** is a researcher in the Multiagent Systems Research Group at the German Research Centre for Artificial Intelligence. His research interests include distributed artificial intelligence, agent-based coordination mechanisms for electronic markets, and integrated services for e-business. He received a diploma from the University of the Saarland. Contact him at [DFKI GmbH](mailto:dfki.de), Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany; [agerber@dfki.de](mailto:agerber@dfki.de); [www.dfki.de/~agerber](http://www.dfki.de/~agerber).

2. J. Yamamoto and K. Sycara, "A Stable and Efficient Buyer Coalition Formation Scheme for E-Marketplaces," *Proc. 5th Int'l Conf. Autonomous Agents*, ACM Press, New York, 2001, pp. 576–583.

3. O. Shehory, "Optimality and Risk in Purchase at Multiple Auctions," *Proc. 5th Int'l Workshop Cooperative Information Agents (CIA 2001)*, vol. 2, 182, Springer-Verlag, New York, 2001, pp. 142–153.

4. R. Conte and J.S. Sichman, "DEPNET: How to Benefit from Social Dependence," *J. Mathematical Sociology*, vol. 20, 1995, pp. 161–177.

5. R.D. Luce and H. Raiffa, *Games and Decisions: Introduction and Critical Survey*, John Wiley & Sons, New York, 1957.

6. S. Kraus, "Negotiation and Cooperation in Multi-agent Environments," *J. Artificial Intelligence*, vol. 94, nos. 1–2, 1997, pp. 79–98.

7. G. Vauvert and A. El Fallah-Segrouhni, "Coalition Formation among Strong Autonomous and Weak Rational Agents," *Proc. 10th European Workshop Modelling Autonomous Agents in a Multi-agent World (MAAMAW)*, Springer-Verlag, Heidelberg, 2001.

8. J.P. Kahan and A. Rapoport, *Theories of Coalition Formation*, Lawrence Erlbaum Associates, London, 1984.

9. M.J. Osborne and A. Rubinstein, *A Course in Game Theory*, MIT Press, Cambridge, Mass., 1994.

10. T. Sandholm, "Distributed Rational Decision Making," in *Multiagent Systems: A Modern Introduction to Distributed Artificial Intelligence*, MIT Press, Cambridge, Mass., 1999, pp. 201–258.

11. L-K. Soh and C. Tsatsoulis, "Real-Time Satisficing Multiagent Coalition Formation," *Proc. Int'l AAAI Workshop Coalition Formation in Dynamic Multiagent Environments*, AAAI Press, Menlo Park, Calif., 2002.

12. S. Ketchpel, "Coalition Formation among Autonomous Agents," *Proc. European Workshop Modeling Autonomous Agents in a Multi-agent World (MAAMAW 93)*, Springer-Verlag, Heidelberg, Germany, 1993, pp. 73–88.

13. M. Klusch and O. Shehory, "Coalition Formation among Rational Information Agents," *Proc. European Workshop Modeling Autonomous Agents in a Multi-agent World (MAAMAW 96)*, *Lecture Notes in Artificial Intelligence*, Springer-Verlag, Heidelberg, Germany, vol. 1,038, 1996, pp. 204–217.

14. M. Klusch and O. Shehory, "A Polynomial Kernel-Oriented Coalition Formation Algorithm for Rational Information Agents," *Proc. Int'l Conf. Multi-agent Systems*, AAAI Press, Menlo Park, Calif., 1996, pp. 157–164.

15. J. Contreras, M. Klusch, and J. Yen, "Multi-agent Coalition Formation in Power Transmission Planning: a Bilateral Shapley Value Approach," *Proc. 4th Int'l Conf. Artificial Intelligence Planning Systems*, AAAI Press, Menlo Park, Calif., 1998, pp. 19–26.

16. M. Mares, *Fuzzy Cooperative Games-Cooperation with Vague Expectations*, Physica Verlag, 2001.

17. J. Suijs et al., "Cooperative Games with Stochastic Payoffs," *European J. Operational Research*, vol. 113, 1999, pp. 193–205.

For more information on this or any other computing topic, please visit our digital library at <http://computer.org/publications/dlib>.



**Agent-Based Business Application Services**



---

## Introduction

Intelligent agent technology is considered a key enabler of both the Semantic Web and service oriented computing. In particular, agents are responsible for proactive and meaningful coordination, maintenance, and provision of Web resources including services on behalf of their human users or other agents. Apart from this often touted claim, it still remains to be shown whether agent coordinated business applications and services in the Web 2.0 and Semantic Web are profitable, outstripping the respective development costs by a magnitude, preferably in the short term. Though there is some evidence for achieving it in terms of best practices of applied agent technology in the past.

### Agent-Based Business Applications

Since the mid 1980s, a vast number of agent-based business applications and systems has been developed world wide many of which by major business stakeholders only in the past decade. Main reason for this is that the characteristics of autonomous and deliberative actors capable of bounded rational decision making, collaboration, and adaptation to dynamic changes in open, resource limited computing and communication environments perfectly match with the requirements of a large variety of business application landscapes (cf. chapter 1).

#### *Practical use of agent systems in industry and commerce*

According to the AgentLink Delphi survey conducted in 2005, manufacturing, logistics, telecommunications, and healthcare are the most significant business application domains for agent technology [2, 246]. The following examples provide a representative glimpse at implemented agent-based business applications in practice but are by no means complete.

- IBM uses autonomous software agents to support its autonomic computing systems, which have the intelligence to reconfigure themselves in response to changing conditions.

- Daimler-Chrysler implemented an agent-based system on one manufacturing factory floor to allow individual workpieces to be directed dynamically around the production area to rapidly meet changing operation targets which resulted in an 20% increase in productivity on average.
- American Air Liquide, an international group specialised in industrial and medical gases, used agent technology to adapt production schedules to changing conditions and deliver products more cost-effectively on demand and responsive to unexpected events.
- The telecommunication giant AT & T prepared its bidding strategy in a federal communications commission auction in part based on its work in the first trading agents competition in 2000.
- At NYSE, the New York stock exchange, automated agent traders account for over 50% of portfolio trades by value most weeks, in some weeks even as much as 70% while outdoing human commodity traders by 7%.
- NASA satellite control software uses autonomous agents to balance multiple demands, such as staying on course, keeping experiments running, and dealing with the unexpected, thereby avoiding waste or loss worth some billions of dollars; the agent controlled probe Deep Space 1 autonomously operated at a distance of 96 million kilometres from earth over two full days in 1999 without any remote ground control.
- Tankers International, which operates one of the world largest oil tanker pools, has applied an agent-based optimiser for adaptive real-time planning of cargo assignment to vessels in the fleet in response to transport cost fluctuations, or changes to vessels, cargo, or ports. This did not only provide improved responsiveness, but also reduced the human effort necessary to deal with the vast amount of information required, thus reducing costly mistakes of scheduling performed by humans.
- The multi-million European project HealthAgents develops an agent-based DDS (Distributed Decision Support) system for early brain tumour diagnosis and prognosis for a better non-invasive diagnosis, prognosis and treatment plan.<sup>1</sup>
- A recent example in the heavy industry is the deployed multi-agent system AgentSteel which supports the daily targeted steel production management of the German steel manufacturer Saarstahl AG [176].

#### *Information agents for the Internet and Web*

One particular application area of intelligent agents is information discovery and management in the Internet and Web. Intelligent information agents (Klusck, 2001)[193] are a special kind of agents capable of accessing one or multiple, potentially heterogeneous and distributed information sources in the Internet, proactively acquiring, mediating, and maintaining relevant information or services on behalf of its human users, or other agents, preferably just

---

<sup>1</sup> [www.healthagents.net](http://www.healthagents.net)

in time and anywhere. Originally, the term "intelligent information agent" has been coined in the seminal paper of Papazoglou, Laufman and Selis (1992) introducing the paradigm of intelligent and cooperative information systems. One key challenge of developing such systems is to balance the autonomy of networked data, information, and knowledge sources with the potential payoff of leveraging them by the appropriate use of intelligent agents.

As mentioned in chapter 2, the majority of existing intelligent information agents capable of knowledge-based reconciliation of semantic data and system heterogeneities heavily draws upon research results in the fields of multidatabase systems and intelligent information integration dated back in the 1980s. Prominent examples are Web agents for adaptive personal search and recommendation like Personal WebWatcher and Letizia, as well as systems of cooperative information agents that are coordinated by middle agents (broker, matchmaker, mediator) such as TSIMMIS, OBSERVER, IMPACT, InfoSleuth, ARIADNE, MOMIS/MIKS, and MIA. For a survey on prominent information agent systems, we refer to (Klusch et al., 2003; Klusch, 1999)[198, 192]. For more information on commercial products and research prototypes of applied agent technology, we refer to the AgentLink website<sup>2</sup>, relevant anthologies such as [365], journals and proceedings of major conferences and workshops of the field like JAAMAS [17] and WIAS [378], respectively, AAMAS<sup>3</sup>, IAT<sup>4</sup>, and CIA<sup>5</sup>. Due to the inherent inter-disciplinarity of the field, reports on agent-based business applications and systems can also be found in relevant sources of related domains including information systems, information retrieval, grid computing, P2P computing, HCI, and AI in general. The same holds, in particular, for the emerging fields of agent-based Web services and Semantic Web applications both of which we take a brief look at in the following.

### Agent-Based Web Service Applications

Agent technologies are considered mission critical to service-oriented computing for which Web services are central. Since Web services are passive until invoked, rational agents are supposed to play a key role in their intelligent discovery, composition, negotiation, and provisioning to the human user for the application at hand. Though we can clearly distinguish between services and agents (cf. chapter 1), both terms are still often used interchangeably in the literature which makes it hard to identify examples of agent coordinated

---

<sup>2</sup> Portal: [www.agentlink.org](http://www.agentlink.org); Industry related agent applications and software, as of January 2007, are listed at <http://eprints.agentlink.org/view/type/project.html>

<sup>3</sup> International Joint Conference on Autonomous Agents and Multi-Agent Systems, since 1997; [www.aamas-conference.org](http://www.aamas-conference.org)

<sup>4</sup> International IEEE/ACM Conference on Intelligent Agent Technology, since 1999; [wi-consortium.org](http://wi-consortium.org)

<sup>5</sup> International Workshop on Cooperative Information Agents, since 1997; [www.ags.dfki.uni-sb.de/klusch/IWS-CIA-home.html](http://www.ags.dfki.uni-sb.de/klusch/IWS-CIA-home.html)

Web services. In fact, only few deployed Web and Web 2.0 service applications explicitly use agents. Prominent examples are the personalized Web services of Amazon, Google and A9 provided by personal profiling agents.

#### *Agent-based steel manufacturing*

A more complex example in the heavy industry is the multi-agent system AgentSteel developed at DFKI in Saarbruecken. AgentSteel supports the daily targeted steel production management of the German steel manufacturer Saarstahl AG in Voelklingen [176]. Different types of rational (BDI) agents represent different entities involved in the internal supply chain of Saarstahl. The real-time multiagent system AgentSteel calculates solutions for given daily target schedules based on customer orders by combined means of distributed online planning and scheduling, guards its realisation, and supports the reorganisation after operational faults by suggesting new solutions to the responsible human conductor of the production process. Complementary interaction with external customers in due course of negotiating production sales or required resources on demand is realized by means of a service-oriented architecture in which the partner services are described in WSDL and respective business processes are modelled in BPEL.

#### *Agent-based distributed data mining*

Cheung and his colleagues (2006)[66] propose a service-oriented architecture for privacy preserving distributed data mining (DDM). Distributed local data sources are protected by probabilistic data abstraction, and each local data mining service implemented as a Web service in WSDL is invoked according to a given client-server style interaction protocol to achieve the overall mining result which choreography is described in BPEL. The problem is to perform active distributed data mining upon a service-oriented platform with privacy preservation and low bandwidth demand. The security related part of this work bases on the approach for secure distributed data clustering by Klusch, Lodi and Moro (2003) which also proposes the use of rational agents to respect the autonomy of the local data sites and flexibly coordinate the distributed data mining process on demand (cf. chapter 18).

More information on agent-based Web services for business applications can be found in the proceedings of major conferences on Web services like ECOWS<sup>6</sup> and ICSOC<sup>7</sup>, the sources of relevant standardisation organisations like OASIS and W3C dedicated to the subject, and other relevant events such as the recently started workshop series on service-oriented computing and agent-based engineering (SOCABE), as well as relevant research projects like the European funded integrated project ATHENA<sup>8</sup>.

---

<sup>6</sup> European Conference on Web Services, since 2003.

<sup>7</sup> International Conference on Service Oriented Computing, since 2003.

<sup>8</sup> [www.athena-ip.org](http://www.athena-ip.org)



## Agent-Based Semantic Web Applications and Services

As mentioned in chapter 2, the Semantic Web promises to make the Web more accessible to intelligent software agents allowing them to deliberate about the meaning of Web resources including services based on shared formal ontologies, as well as the observed environmental and given application context. In other words, quoting Tim Berners-Lee, Jim Hendler and Ora Lassila from their seminal paper on the Semantic Web vision published in 2001 [31]:

”The real power of the Semantic Web will be realized when people create many programs that collect Web content from diverse sources, process the information and exchange the results with other programs. The effectiveness of such *software agents* will increase exponentially as more machine-readable Web content and automated services (including other agents) become available. The Semantic Web promotes this synergy: even agents that were not expressly designed to work together can transfer data among themselves when the data come with semantics.”

Remarkably, the majority of currently deployed Semantic Web applications and services does not make explicit use of agent technology yet. Prominent exceptions are the RETSINA Semantic Web calendar agent developed at Carnegie Mellon University<sup>9</sup> which is sometimes claimed to be the first Semantic Web agent developed ever, and a few other research projects including SCALLOPS, CASCOM (cf. chapter 17), and SmartWeb.

### *SmartWeb*

The national government funded project SmartWeb<sup>10</sup> coordinated by DFKI in Saarbrücken (Germany) targets Semantic Web services applications that include the multimodal and context-aware shopping assistance for the user of future cybershops in the so called Internet of Things. Such shops are instrumented by means of RFID tagged products, smart sensors, embedded speech and gesture recognition, and user tracking devices. The required characteristics of such an ambient intelligence environment for meaningful service provision like context-awareness, pro-activeness and adaptivity naturally correspond to those of rational agency, hence can be manifested by appropriate lightweight agents that coordinate the access to, and processing of relevant semantic services and content.

### *SCALLOPS*

The national project SCALLOPS<sup>11</sup> developed innovative means for flexible agent-coordinated Semantic Web service discovery and composition, and effective data and service privacy enforcement in open, large scale pervasive

<sup>9</sup> [www.daml.rli.cmu.edu/Cal/](http://www.daml.rli.cmu.edu/Cal/)

<sup>10</sup> [www.smartweb-project.de](http://www.smartweb-project.de)

<sup>11</sup> [www.dfki.de/scallops](http://www.dfki.de/scallops)

computing environments. The core components of service coordination in the Health-SCALLOPS system are the semantic service discovery and composition planning agents using OWLS-MX (cf. chapter 5), respectively, OWLS-XPlan (cf. chapters 8 and 9) The application oriented interplay between these agents has been successfully demonstrated in selected use cases of the e-health domain. These use cases concern (a) the medical repatriation of patients across the world, and (b) the negotiation of medical services between hospital physicians and online pharmacies with particular focus on privacy preservation of both service provider and consumer.

For privacy preserving semantic service negotiation, appropriate means for uncertain, trusted and privacy preserving coalition formation as we have presented in part 4 are used for negotiations. Privacy preserving OWL-S composition planning is achieved by complementary checking of generated service plans based on information flow analysis (Hutter, Klusch & Volkamer, 2006)[174].

### *CASCOM*

The European project CASCOM<sup>12</sup> (cf. chapter 17) developed personal emergency medical assistance services in OWL-S that are coordinated by different types of agents and provided to the user on demand on connected mobile computing device such as HP iPAQ and Nokia series 9 PDA. The agent-based Semantic Web service coordination system has been applied to the real world use cases of personal emergency medical assistance and coordinated medical patient transport and repatriation across Europe. Restricted user trials of the CASCOM system were successfully executed by emergency and hospital physicians, patients, and a simulated emergency medical assistance center in Innsbruck, Austria.

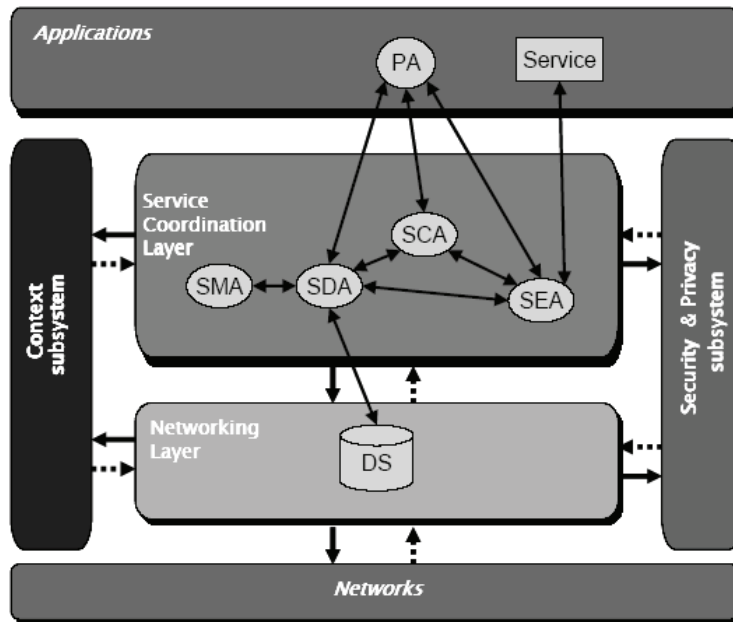
### **An Agent-Based Semantic Web Service Coordination Architecture**

To date, there does not exist any reference architecture for agent-based semantic service coordination. Figure 13.1 shows the layered CASCOM generic coordination architecture for Semantic Web services.

The Networking Layer realizes an intelligent P2P network that provides an efficient, secure and reliable communication independent of the access technology. The Service Coordination Layer, situated above the networking layer, provides flexible semantic service discovery, matchmaking, composition and execution functionalities. Orthogonal to both layers, the Context Subsystem is in charge of acquiring, storing and providing context information to both layers. The Security and Privacy Subsystem, also orthogonal to the Networking and Service Coordination Layer, is responsible for ensuring security and privacy of information throughout the different components of the CASCOM

---

<sup>12</sup> [www.ist-cascom.org/](http://www.ist-cascom.org/)



**Fig. 13.1.** CASCOM semantic service coordination architecture (PA: Personal Agent, SMA: Service Matchmaking Agent, SDA: Service Discovery Agent, SCA: Service Composition Agent, SEA: Service Execution Agent, DS: Directory Service)

infrastructure. The agents are implemented in JADE/LEAP since it complies with the FIPA standard and allows agents to be efficiently executed on resource constrained computing devices.

Semantic service coordination in CASCOM is realized by four different types of agents: (1) Service Discovery Agents (SDA) perform a context dependent, keyword-based search in service directories of the networking layer, (2) Service Matchmaking Agents (SMA) complement this search by a more fine grained logic-based service matching, (3) Service Composition Agents (SCA) create value-added composite services that match service specifications, in case the SMAs fail to discover relevant existing services, and store them in some directory for later use, and (4) Service Execution Agents (SEA) manage the execution of both atomic and composite value-added service descriptions generated by SCAs. Whenever necessary, SEAs will use SDAs to discover appropriate available service providers for each of the simpler services evoked from the compound service description. The distributed service execution model is based on principles of the OSIRIS process management system.

#### Further Readings and Outlook

For more information on agent coordinated Semantic Web applications and services, we refer the interested reader to, for example, the proceedings of

major conferences in the field like ISWC and ESWC, and relevant research projects such as SmartWeb, SCALLOPS, KnowledgeWeb, DIP, SEKT, CASCOM. There are also a few volumes on the topic in press by Springer publisher. However, to the best of our knowledge, no agent-based Semantic Web service application made it into the real world of common Internet users broadly yet. However, according to the hypecycle for emerging technologies published by Gartner in mid 2006, that may gradually change within the next five to ten years. Whereas the Semantic Web and Web 2.0 are actually seen at their highest peak of raised expectations, Web service technology is considered already mature enough for its use within enterprise business application landscapes. It remains to be seen whether Semantic Web services can reach the same level of maturity within the next decade.

Though agent technology and Semantic Web services were not mentioned in this survey, we again stress that the full application potential of the Semantic Web can only be reached by synergetic effects from both fields. Similarly, P2P facilities may come as Web services, while their reliable, task-oriented, resource-bounded, and adaptive coordination-on-the-fly characteristics call for agent-based software technology. The trading offs between full rational agency and ultra-limited resource boundedness of many pervasive computing environments including smart sensor networks still remain to be investigated but first results in this direction appear promising. As a result, we strongly believe in the convergence of the Web 2.0, Semantic Web technologies, pervasive computing, P2P computing, and service-oriented architectures to become the major driving force of the development of agent-based business applications and services for the future Internet and Web.

### **The Contributions**

In the following chapters, we present selected systems of agent-based information services in different application domains such as agriculture and e-health. The application services range from timber production and trading via resource planning for cereal harvesting to emergency medical assistance. These systems have been jointly developed with several Master and PhD students of my research team at DFKI, and research partners in different institutions and companies. We conclude this part with an innovative solution for privacy preserving distributed data clustering.

*Chapter 14: CASA - Integrated Timber Production and Trading Services.* In this chapter, we present the agent-based information and trading network CASA ITN in which teams of collaborating agents provide personalized and integrated mobile services for dynamic timber production, and auction-based timber and cereal trading. The CASA ITN has been the first implemented agent-based system for this kind of application services in the agricultural domain. The mobile CASA services were tested on WAP enabled cell phones within a virtual private network and a Web application server of Deutsche

Telekom. Other co-authored publications on CASA not included in this chapter are (Klusch & Gerber, 2001)[203], (Gerber et al., 2002)[128].

*Chapter 15: AGRICOLA - Mobile Resource Planning for Cereal Harvesting.* The AGRICOLA multi-agent system provides mobile services for dynamic resource planning in due course of cereal harvesting. AGRICOLA features two modes of resource scheduling: Centralized coordination of required machines and personnel by means of a distinguished mediator, or peer-to-peer collaborative by means of dynamic formation of temporary coalitions among resource providers and consumers on demand. For the latter purpose, AGRICOLA agents use the DCF-S coalition scheme as described in chapter 14. The services of AGRICOLA are accessible via mobile devices such as HP iPAQ or Fujitsu-Siemens PocketLoox. As with the CASA system, AGRICOLA was the first implemented agent-based system of its kind in the agricultural application domain.

*Chapter 16: CASCOM - Mobile e-Health Assistance Services.* The essential approach of CASCOM is the innovative combination of agent technology, Semantic Web services, peer-to-peer, and mobile computing for intelligent peer-to-peer mobile service environments. Conventional peer-to-peer computing environments are extended with components for mobile and wireless communication. The services of CASCOM environments are provided by peer software agents which exploit the CASCOM coordination infrastructure to efficiently operate in highly dynamic environments. The generic CASCOM intelligent peer-to-peer infrastructure includes efficient communication means, support for context-aware adaptation techniques, as well as dynamic and secure service discovery and composition planning. In fact, the CASCOM system uses the OWLS-MX matchmaker (cf. chapter 5) and OWLS-XPlan2 composition planner (cf. chapters 8 & 9) The CASCOM demonstrator for mobile health care application services was successfully field tested for a selected use case of emergency medical assistance. For more information on the CASCOM system, we refer to the project web site and the book (Schumacher et al., 2008)[330] including several coauthored chapters on service discovery, composition, and execution of the system.

*Chapter 17: KDEC - Secure Distributed Data Clustering.* In this chapter, we argue for the use of agents in distributed data mining, provide an example of secure agent-based distributed data clustering, called KDEC, and investigate possible privacy threats of the KDEC mining process. Key idea of KDEC is to communicate Kernel-based data density estimations of original data stored at local sites only such that its reconstruction by eavesdropping agents is limited to a given extent. While the field of agent-based data mining is currently regaining some momentum, KDEC has been the first implemented solution to the problem of privacy preserving distributed data clustering (Klusch, Lodi & Moro, 2003)[208]. Other co-authored publications

on agent-based distributed data mining not included in this chapter are, for example, (Klusch, Lodi & Moro, 2003)[208], (Klusch et al., 2003a)[209], (Costa Da Silva et al., 2004)[75], (Costa Da Silva & Klusch, 2006)[73], and (Costa Da Silva & Klusch, 2007)[74].

## CASA: Integrated Timber Production and Trading

A. Gerber and M. Klusch: Agent-based Integrated Services Network for Timber Production and Sales. *IEEE Intelligent Systems*, 17 (1), pages 32 - 39, IEEE CS Press, January/February 2002.

# Agent-Based Integrated Services for Timber Production and Sales

Andreas Gerber and Matthias Klusch, *German Research Centre for Artificial Intelligence*

**T**he ability to access and send information from just about anywhere—one of the prime advantages of pervasive computing—is transforming the way we live and work. Pervasive computing's advantages have strengthened countless industries, and now there is a palpable need to enhance just-in-time production and trading in agriculture

and forestry. To solve some of the problems in forestry and agriculture, the German Research Center for Artificial Intelligence and the Saarland Ministry of Economics founded the Cooperative Agents and Integrated Services for Logistic and Electronic Trading in Forestry and Agriculture (Casa) project.

The Casa project focuses on developing an agent-based information and trading network (ITN). In particular, the project seeks to establish mobile, integrated services in selected application scenarios within forestry and agriculture. Casa ITN supports the main business processes that users perform in customer-oriented dynamic timber production, mobile timber trading, and electronic cereal trading.

The paradigm that informs the Casa ITN project is *integrated commerce*, an operational extension of traditional e-commerce that entails getting customers more involved in ordering activities so that contractors can more efficiently fulfill orders. I-commerce also entails more effective practical integration of supply-chain processes. In this light, Casa ITN offers its users several i-commerce techniques for negotiating, communicating, and exchanging information more effectively. We anticipate that these techniques will lead to more effective integration of production, logistics, and trading processes in other industries as well.

## Holonic agents and services

The Casa ITN system differentiates between the following participant groups: producers, buyers, retailers, and logistics companies. Casa ITN represents each member of these groups with an appropriate kind of software agent called a *holonic agent*.<sup>1-3</sup> Holonic

agents accomplish complex (mostly hierarchically decomposed) tasks and resource allocations in the selected application scenarios. They also coordinate and control the activities and information flows of their subagents. In a holonic multiagent system, autonomous agents can join other agents to form, reconfigure, or leave a holon.

This holonic agent technique lets personal assistants that represent human users act on behalf of their users even if those users are offline. Personal assistants operate as the coordinating heads over a set of other specialized agents designed for enhancing individual negotiation, auction participation, information location, and strategic trading.

In addition to representing users with agents, Casa ITN represents each corporation with a special holonic agent system according to each corporation's task-oriented classification. These subdivisions treat corporations from the perspective of information management, logistics, and production planning. Information management services provide information either on certain products and related production processes or on current market situations and potential competitors. Logistics services support coordinating machines for production and transport, human resources, and storage capacities. Finally, production planning services support short-, middle-, and long-term product planning cycles.

A corporation holon consists of several holonic agents, each of which represents a special department and its corresponding tasks and services. Because Casa ITN can use the roles of buyer and retailer and seller and producer interchangeably, we model both

*I-commerce extends e-commerce by getting customers more involved so that contractors can more efficiently fulfill orders. The Casa project focuses on developing an agent-based information and trading network to help establish integrated services in forestry and agriculture.*





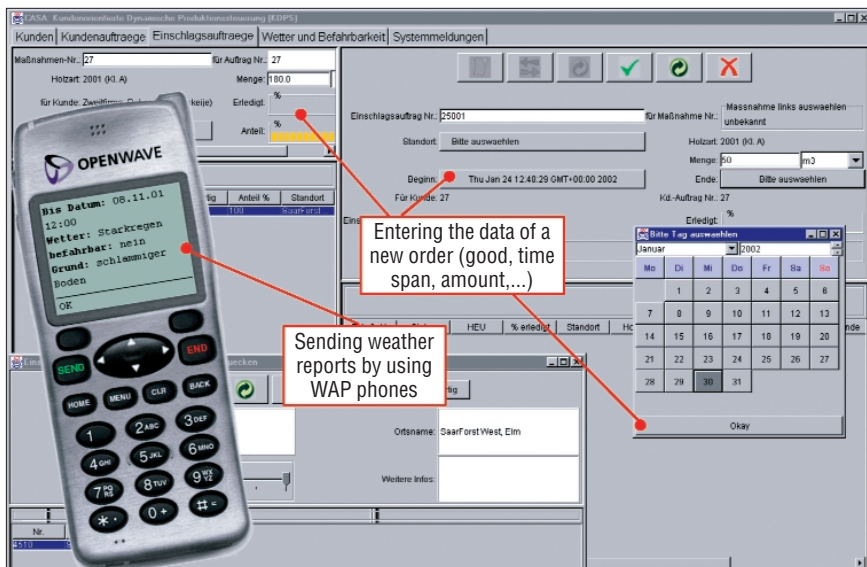


Figure 2. On the Casa ITN, you can send a weather report with a WAP phone and organize the resources and orders in a forestry district.

region when the ground is too wet (the ground might be damaged), the foresters must provide continuous reports on the ground's appearance. On the basis of this information, the harvesters must dynamically plan and schedule the cutting sequence. During the cutting process it is essential that the forester and harvester efficiently exchange data about the weather and ground conditions. In this context, Casa ITN helps with the use of mobile services. These services let the forester send information to the harvester's server, where a planning and scheduling program can automatically process this information.

During the cutting process, the harvester sends regular status messages to the forester with information about where all the cut trees are lying and how much of the task is complete. Because of these updates, the forester gets a better view of overall progress. When the harvester completes a task, the forester receives a report. Then the forester and the customer exchange information about the trees and the financial guarantees, so that the customer gets permission to carry these trees out of the forest. The customer then has to find a shipping company that can transport the trees.

After the trees arrive at the customer's site, the customer and forester negotiate the real cost of the goods. Figure 1 shows events that are susceptible to problems during the process; these problems can cause scheduling failures. In a supply-chain scenario like the one we've just described, where one event can dynamically influence another, the systems must prevent or minimize the impact of uncertainty.

### Agent society

Because of the uncertainties inherent in the timber-production scenario, the system must be flexible enough to reorganize itself quickly. We use a multiagent system because such systems are well suited for dealing with complex tasks that can be divided into a set of subtasks. Multiagent systems exhibit stability and robustness because one agent can often take over the role of another in case the latter is inhibited or suspended for some reason. Agents in a multiagent system can also exchange messages to coordinate their efforts more effectively.

We use complex agent structures—consisting of holonic agents—to represent the abstract groups of participants. We represent the forester (or forest district) with several holonic agents: a personal assistant agent to help the user while using the system; a holonic agent for information management; a planning holonic agent to build production plans and schedules; and a holonic agent to represent the company as a whole and to build an interface to the outer agent society. In addition, four holonic agents represent the harvester company: a company holonic agent to present the company's internal agent society to the outer agent society with a communication interface; an information managing holonic agent; a logistics holonic agent that coordinates and controls the company's resources; and a planning holonic agent to build an execution schedule.

Another group of holonic agents represents the customer's company in the Casa ITN. The agent structure of the customer's company is analogous to the harvester com-

pany's agent structure. But the components of the customer's company are not as detailed as the harvester's. Rather, the customer agents must provide an information service that helps the trader obtain knowledge about resources and needs, including required dates and timing information. The involved parties use this information during the trading and negotiating process.

The logistics company—which offers services for transporting trees—has a structure similar to the harvester's. Both companies must coordinate, control, and schedule the resources of the executing companies. The harvester and the logistics companies receive task requests, but they are not involved in any trading process. Instead, they provide services that other companies use.

### Implementation details

To use the Casa ITN system, a participant has to be member of the ITN to access the ITN services. Because sensitive information must be exchanged across the network, there is a strong need for security and safety. We built the Casa ITN as a secure extranet. We implemented the Casa ITN services in Java JDK 1.3 and the system prototype on Windows machines using various kinds of modern WAP phones from different producers.

We designed the Casa holonic agents in accordance with the InteRRaP agent architecture and we implemented them using the standard open-source FIPA-compliant agent system development environment FIPA-OS.1 (<http://fipa-os.sourceforge.net>). We made sure that the implementation of Casa ITN's mobile application services maintained compliance with the WAP 1.1 and WAP 1.2 standards. Users access the Casa ITN services through the secure T-D1 WAP-gateway of the Deutsche Telekom AG. In the prototype, we used Checkpoint's VPN suite for data security.

Figure 2 shows how Casa sends a weather report on the extranet by means of a WAP phone. A PC connected to the Internet can also send weather reports. Such report information can be very important for the harvester company, which might need to adjust its plans to avoid idle times. To handle all the information and planning, the software package must consist of three parts: WAP services, forestry software, and harvester software. The forester gets a software bundle that lets them organize their goods, incoming orders, and the activities of the commissioned harvester companies, including WAP



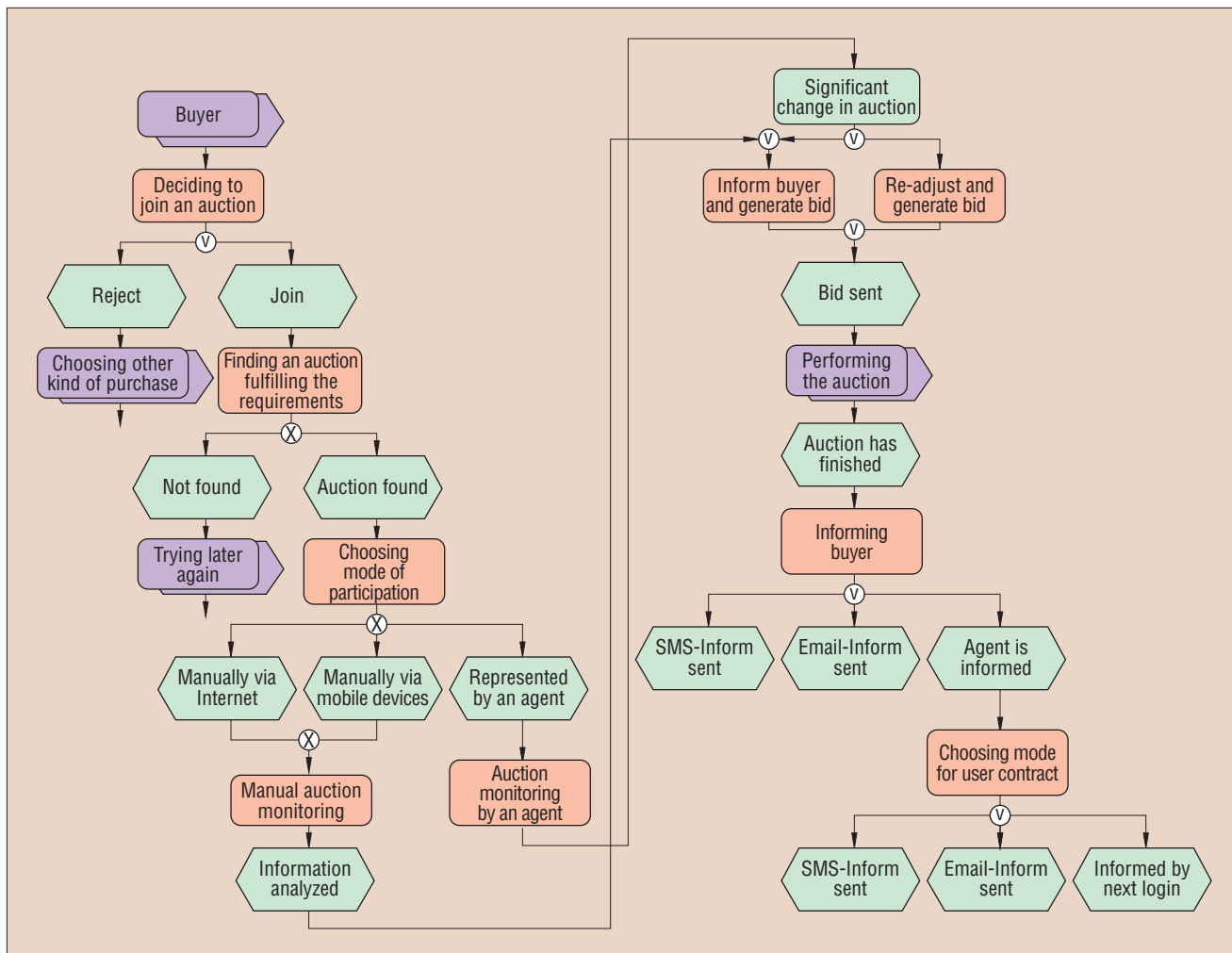


Figure 5. Interaction between participants in a mobile timber auction.

One of these computers acts as a server connected to the ITN. This server consists of a planning component built from a resource and an order management component. The order management unit handles all incoming orders and offers interfaces for receiving weather and status reports sent by mobile devices. The resource management unit schedules all the machines and workers in the company. The workers in turn have access to WAP devices for sending reports directly from their working location and to their company’s server.

When the harvester company receives an order, the order builds a connection between forestry and the selected harvester company. The weather service builds a connection between forestry and any harvester company registered to the weather-service component on the ITN. So everyone who subscribes to the weather service gets automatically updated information about the actual weather situation when reports are sent.

### Mobile timber sales

When the project is finished, the owner can sell the wood using a fixed-price negotiation, a predefined contract, or an auction. Typically, at least part of the wood production will be sold at auction. But the auctions in the forestry industry are usually performed by hand, which means they are error-prone and require a long time to finalize. On the Casa ITN system, however, each owner can sell timber through different kinds of auctions to other registered users. The main benefits of this agent-based service include concurrent price monitoring in addition to the ability to compute optimal transport costs during bidding. Furthermore, the use of mobile services during an auction enables new possibilities for auction monitoring and participation.

In general, Casa ITN’s mobile timber sales services let registered users initiate or participate in one or multiple timber auctions and sell or buy timber at fixed or negotiable

prices. As mentioned earlier, Casa ITN support the following types of auctions: Dutch, English, Vickrey, and First-Price-Sealed-Bid. The auction server relies on a general holonic coordination server for supply webs.<sup>6</sup>

Users can invoke integrated services for decision support while participating in one or more auctions. For example, a personal Casa ITN agent might concurrently determine the optimal transportation costs and delivery dates of some auction good for each individual user bid. As a result, the agent can notify its user in real time if estimated optimal transport costs exceed the allowed limit or if some preset deadlines are at risk of being violated.

The holonic agents provide integrated trading services and facilitate participant interactions, namely those between seller and auctioneer, buyer, and shipping company. Figure 5 summarizes the possible interactions at an auction. First, a seller typically initiates an auction on a trusted auction site and informs potential buyers about the offered goods, the



auction type, and the related bidding policies. During an auction, the bidders can evaluate their bids, monitor the current auction process, and place bids according to their individual bidding strategies. If users access the auctions through mobile phones or WAP devices, appropriate Casa ITN agents perform the necessary synchronizations. At any point, the users can delegate their participation in any trading process to their personal agents, which then take over the process of negotiating prices or bidding at an auction. These agents notify their users (when needed) either by SMS or email.

Generally, a seller has three different options to begin an auction: manually using a personal computer connected with the Internet, manually using a mobile phone, or charging an agent with that task using a PC or mobile device. The buyer can access the system in a similar manner (see Figure 6). When the buyer has found an auction to participate in, that buyer can join the auction using the same methods as the seller.

The advantages of using an agent combined with mobile services are obvious. Buyers and sellers can join an auction and let their agents participate in the auction unattended. When a significant change in an auction happens, the agent can inform its user immediately, which means that the user—whether buyer or seller—does not have to wait to access a PC at a later point. The users simply send new instructions to the agent over a WAP device. Users can therefore react much more flexibly to any changes in an auction.

## Related work

The existing projects relating to agriculture and forestry mostly attempt to facilitate and enhance supply-chain management and business issues. For example, the main objective of the Agriflow project ([www.agriflow.com](http://www.agriflow.com)) is to help Europe's agricultural industry adopt better e-business practices with a series of dynamic products. Another approach, the Virtual Agricultural Market (VAM) system, helps facilitate business-to-business transactions in agricultural markets. The scalable VAM architecture<sup>7</sup> provides mechanisms for Internet-based trading and distribution of perishable agricultural products.

In addition, there are multiagent forestry management systems. These include the Phoenix project ([eksl-www.cs.umass.edu/research/phoenix.html](http://eksl-www.cs.umass.edu/research/phoenix.html)), which helps fight forest fires with a sophisticated computer simulation system based on satellite data. The

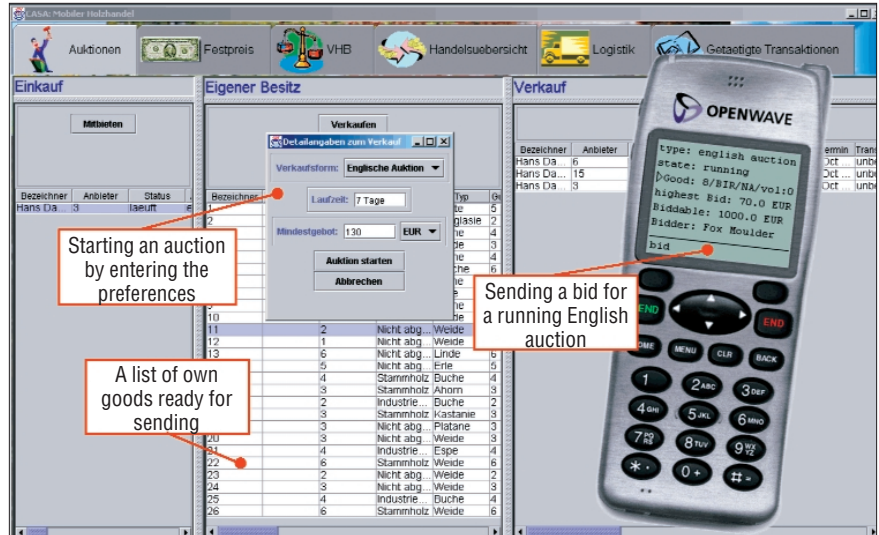


Figure 6. Buyer participation in a running auction.

Phoenix system consists of an instrumented discrete event simulation, an autonomous agent architecture that integrates multiple planning methods, and a hierarchical organization of agents capable of improving their fire-fighting performance by adapting to the simulated environment. The Phoenix agent architecture includes several innovative features that support adaptable planning within real-time constraints, including a least-commitment planning style called lazy skeletal refinement and a combination of reactive and deliberative planning components that can operate at different time scales.

To some extent, these approaches resemble the Casa ITN with respect to the kind of services provided to the user. However, they significantly differ from the Casa ITN project in terms of integration and coordination of services for logistics and trading. In this respect, the Casa ITN system not only offers unique i-commerce services to its users—such as mobile timber auctions with integrated support of optimizing logistics—but also provides the first agent-based framework for e-business in these special application domains.

The Casa ITN system illustrates how agent-based services can help increase the flow of information and help save time in just about any industry. These techniques let buyers, sellers, and all other participants have better control over their products. We plan to extend the Casa ITN system to the agricultural domain using dynamic resource planning through integrated mobile services to enable, for example, flexible and cost-efficient preci-

sion farming. Future research and development of the follow-up prototype, called Agricola.net, will include approaches for agent-based dynamic coalition formation, distributed constraint satisfaction, and adaptive replanning in open, real-time environments. ■

## Acknowledgments

The Ministry of Economics of Saarland, Germany, under Grant 032000, sponsored this research project. We acknowledge the support of the Saarland Ministry of Economy, state-owned business SaarForst, and the Saarland Department of Agriculture. We thank our project partner Trend-Plus.com AG Saarbrücken, Germany, for providing the latest mobile telecommunications equipment and use of the T-D1 WAP gateway of the Deutsche Telekom AG.

## References

1. M. Klusch et al., "Applications of Information Agents and Systems," in *Practical Applications of Intelligent Agents*, L.C. Jain, ed., Physica-Verlag, location of publisher, 2002, pp. XX-XX.
2. H.-J. Bürckert, K. Fischer, and G. Vierke, "Holonc Transport Scheduling With TeleTruck," *Applied Artificial Intelligence*, vol. 14, no. 7, 2000, pp. 697-725.
3. C. Gerber, J. Siekmann, and G. Vierke, "Flexible Autonomy in Holonic Agent Systems," *Proc. 1999 AAAI Spring Symp. Agents with Adjustable Autonomy*, AAAI Press, Menlo Park, Calif., 1999, pp. XX-XX.
4. A. Bachem, W. Hochstättler, and M. Malich, *Simulated Trading: A New Approach for Solving Vehicle Routing Problems*, tech. report 92.125, Mathematisches Inst. der Univ. zu Köln, 1992.

5. H.-J. Bürkert, K. Fischer, and G. Vierke, "Transportation Scheduling with Holonic MAS—The TeleTruck Approach," *Proc. 3rd Int'l Conf. Practical Applications of Intelligent Agents and Multiagents, PAAM'98, publisher?, location of publisher?* 1998, pp. XX-XX.
6. A. Gerber and C. Ruß, "A Holonic Multi-agent Infrastructure for Electronic Procurement," *Proc. 14th Biennial Conf. Canadian Soc. Computational Studies of Intelligence*, Springer-Verlag, Heidelberg, Germany, 2001, pp. 16-25.
7. C.I. Costopoulou and M.A. Lambrou, "An Architecture of Virtual Agricultural Market Systems: The Case of Trading Perishable Agricultural Products," *Information Services and Use*, vol. 20, no. 1, 2000, pp. 39-48.

## The Authors



**Andreas Gerber** is a researcher in the Multiagent Systems group at the German Research Center for Artificial Intelligence. His research interests include distributed artificial intelligence, agent-based coordination mechanisms for electronic markets, and other integrated services for e-business. He received a diploma in computer science from the University of the Saarland, Germany. Contact him at the Multiagent Systems Group, DFKI GmbH, Stuhlsatzenhausweg 3, D-66123 Saarbruecken, Germany; [www.dfki.de/~agerber/](http://www.dfki.de/~agerber/); [agerber@dfki.de](mailto:agerber@dfki.de).



**Matthias Klusch** is a senior researcher in the Multiagent Systems group at the German Research Center for Artificial Intelligence and assistant professor in the Department for Artificial Intelligence at the Free University of Amsterdam, Netherlands. His research includes the application of AI and agent technology to databases and intelligent information systems on the Internet and Web. He received a PhD in computer science from the University of Kiel, Germany. Contact him at the Multiagent Systems Group, DFKI GmbH, Stuhlsatzenhausweg 3, D-66123 Saarbruecken, Germany; [www.dfki.de/~klusch/](http://www.dfki.de/~klusch/); [klusch@dfki.de](mailto:klusch@dfki.de).

---

## AGRICOLA: Mobile Resource Planning for Cereal Harvesting

A. Gerber and M. Klusch: AGRICOLA: Agenten für mobile Planungsdienste in der Landwirtschaft. *Künstliche Intelligenz*, 1/04, pages 38 - 42, arendtap Verlag, 2004.

# **AGRICOLA - Agenten für mobile Planungsdienste in der Landwirtschaft**

Andreas Gerber und Matthias Klusch  
DFKI GmbH Saarbrücken  
Stuhlsatzenhausweg 3, 66123 Saarbrücken

## **Einleitung**

Das Projekt AGRICOLA (2002 - 2003) am Deutschen Forschungszentrum für Künstliche Intelligenz in Saarbrücken hatte die Entwicklung eines agentenbasierten Netzwerks mobiler, integrierter Dienste für eine dynamische Ressourcenplanung in der Landwirtschaft des Saarlandes zum Ziel<sup>1</sup>. Die Schwerpunkte des Projekts waren die

- innovative Forschung, Entwicklung und Evaluierung von generischen, integrierten und mobilen Diensten für eine dynamische, optimale Einsatzplanung von Maschinen, Personal und Material in der landwirtschaftlichen Produktion, sowie die
- prototypische Implementierung eines entsprechenden, regionalen Dienstnetzwerks AGRICOLA.NET im Internet für die an einer Einsatzplanung beteiligten Landwirte, Maschinenbesitzer, Maschinenhersteller und Maschinenring.

Das AGRICOLA.NET Netzwerk bietet Landwirten eine innovative, softwaretechnische Unterstützung für bedarfsgerechte just-in-time Einsatzplanung von Maschinenparks in der landwirtschaftlichen Wertschöpfung. Die lokalen Maschineneinsatzplanungen von Landwirten vor Ort werden mit den partiell globalen Planungen der Maschinenbesitzer unter sich dynamisch verändernden Bedingungen hinsichtlich Witterung, sowie Ausfall von Maschinen, Personal und sonstigem Material bedarfsgerecht und just-in-time aufeinander abgestimmt. Eine Maschineneinsatzplanung durch Maschinenbesitzer umfasst die Faktoren Personal, Wartung und geographische Positionierung von Maschinen; sie ist während der Erntezeit zeitkritisch und kostenaufwendig. Ein für eine bestimmte Region zuständiger Maschinenring vermittelt zwischen Landwirten, die Maschinen für ihre operativen Prozesse benötigen, und einem oder mehreren Maschinenbesitzern mit entsprechend verfügbaren Maschinenparks.

## **Anwendungsszenario**

Im Projekt AGRICOLA wird von einer Aufteilung eines landwirtschaftlichen Betriebs in landwirtschaftliche Planungsprozesse, sowie Verwaltung und Koordination des Maschinenparks und des Personals als Dienstleister ausgegangen. Die Einsatzplanung findet auf drei Ebenen statt:

- Planung auf Aufgabenebene
- Koordination innerhalb eines Prozesses
- Übergreifende Planung (Koordination der Abhängigkeiten zwischen den Prozessen)

Das Anwendungsszenario von AGRICOLA befasst sich mit der Organisation der Arbeitsschritte, die zur Bearbeitung eines Feldes während der Ernte notwendig sind. Dazu sind allen registrierten Teilnehmer im Dienstnetzwerk AGRICOLA.NET spezielle Rollen zugewiesen (siehe Abbildung 1). Um eine konzeptionelle Trennung zwischen Koordination, Ressourcen- und Aufgabenverwaltung zu erreichen, wird eine Aufteilung der Funktionsgruppen in entsprechende Rollen wie folgt vorgenommen:

---

<sup>1</sup> Das Projekt ist vom Saarländischen Ministerium für Wirtschaft gefördert worden. Die inhaltliche Kooperation mit Anwendern von AGRICOLA.NET erfolgte mit Unterstützung des Saarländischen Ministeriums für Umwelt, Verbraucherschutz und Landwirtschaft.



- *Landwirte* führen die Aufgabenverwaltung ihrer Prozesse durch. Sie besitzen dabei keine eigenen Ressourcen zur Bearbeitung dieser Aufgaben.
- *Maschinenbesitzer* bieten Ressourcen für die Ausführung von Aufgaben den Landwirten an.
- *Koordinatoren* vermitteln Teilnehmern mit Ressourcenbedarf und Ressourcenangebot.

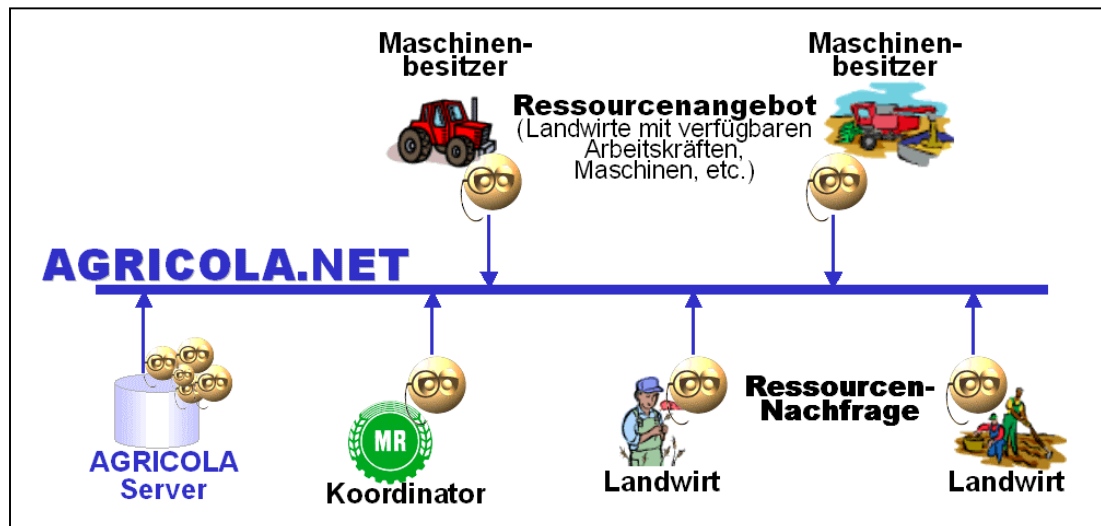


Abbildung 1. Teilnehmer im AGRICOLA.NET

Im Internetbasierten AGRICOLA.NET werden alle Teilnehmer und Ressourcen durch geeignete Agenten repräsentiert. Persönliche *AGRICOLA Agenten* unterstützen die Teilnehmer in ihrer Planung von Ausleihe und Einsatz benötigter Nutzfahrzeuge für die Getreideernte, wie beispielsweise Drillmaschinen zur Aussaat, Mähdrescher, Traktoren zum Schwaden und Wenden, oder Schlepper zum Transport der Ernte oder sonstiger Güter. Alle agentenbasierten Dienste von AGRICOLA sind über mobile Telekommunikationsgeräte jederzeit an jedem Ort verfügbar. Der Landwirt wird unter Einsatz des AGRICOLA Systems in der Lage versetzt, seine individuelle Planung von Ernterelevanten Arbeitsverfahren auf seinen Feldern in sogenannten *Verfahrensketten* (teil-) automatisiert durchzuführen. Die Planung einer solchen Verfahrenskette setzt sich aus der Auswahl einzelner Aufgaben (wie z.B. Drillen, Ernten, Schwaden, usw.) und der Festlegung ihrer chronologischen Reihenfolge zusammen. Steht die Verfahrenskette fest, müssen geeignete Dienste von Ressourcenanbietern zur Bearbeitung der einzelnen Aufgaben gefunden werden. Dabei stehen dem Anwender folgende Möglichkeiten zur Verfügung.

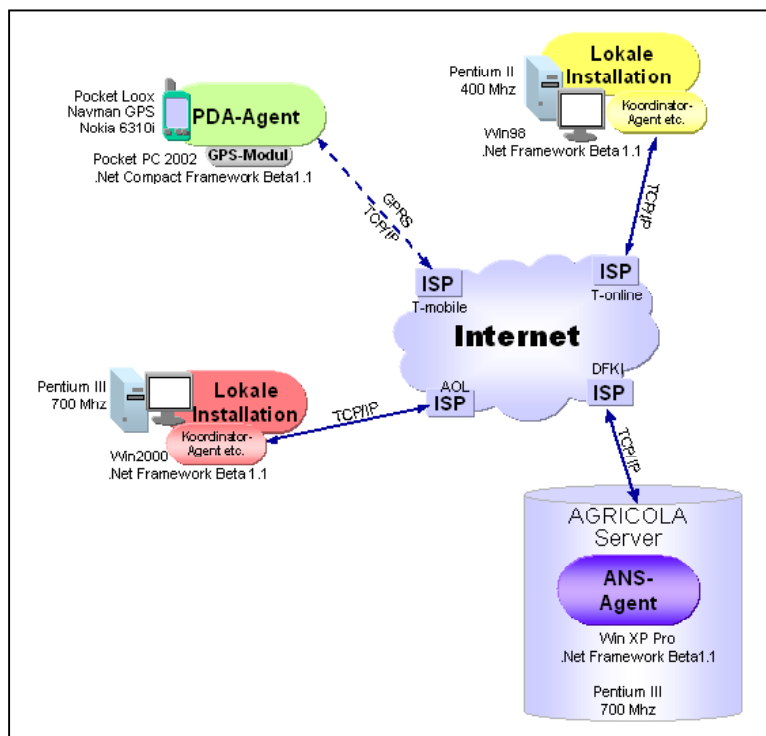
- Eigenhändige Auswahl und Zuweisung von gewünschten Ressourcen zu Aufgaben. Diese Vorgehensweise beschränkt sich auf Ressourcen, auf die dieser Teilnehmer uneingeschränkter Zugriff besitzt. In der Regel sind dies seine eigenen Ressourcen, bzw. Ressourcen von Teilnehmern (im System als „Freunde“ markiert), die ihm die vollständige Kontrolle ihrer Ressourcen überlassen.
- Zuweisung von Aufgaben zu Gruppen von Ressourcen, die für die Bearbeitung dieser Aufgabe relevant sind und auf die zugegriffen werden kann. Das System führt dann automatisch eine Ausschreibung der Aufgaben an diese Gruppen aus und plant sie ein.
- Globale Ausschreibung von Aufgaben an alle im Netz vertretenen Teilnehmer, die Ressourcen besitzen. Je nach dem Grad des Vertrauens der Teilnehmer im AGRICOLA.NET untereinander, können eingehende Anfragen durch das System

entweder direkt beantwortet oder lediglich als Anzeige für eine spätere manuelle Bearbeitung durch den Ressourcenanbieter bereitgehalten werden.

Im AGRICOLA.NET können Ressourcen und Aufgaben gezielt an Teilnehmer des Netzwerks ausgeschrieben und ihnen jederzeit die Zugriffsrechte darauf wieder entzogen werden. Darüber hinaus können einzelne Teilnehmer zwischen Angebot und Nachfrage von Ressourcen verschiedener Teilnehmer vermitteln. Ist die Zuweisung abgeschlossen, müssen die Ressourcen real gebucht werden. Das System unterstützt den Anwender hierbei durch automatisierte Verhandlungen nach einem erweiterten Kontraktnetzprotokoll für Multiagenten-Systeme.

## Systemarchitektur und Implementierung

Die Anwendung von AGRICOLA.NET wird auf verschiedene über das Internet verbundene Computersysteme (PC, Notebook, und PDA) der Anwender verteilt. Die AGRICOLA Agenten werden auf den lokalen Systemen jeweils als eine Art *fat Clients* von AGRICOLA.NET installiert, die eigenständig umfangreiche Funktionen ausführen können. Jeder Agent kann über einen definierten, pro Agent eindeutigen Port mit anderen Agenten kommunizieren und verfügt über einen Gelbe-Seiten Dienst für die ihm bekannten Agenten. Ein zentraler *Agent-Name-Service* Agent (ANS) auf dem sog. AGRICOLA Server in

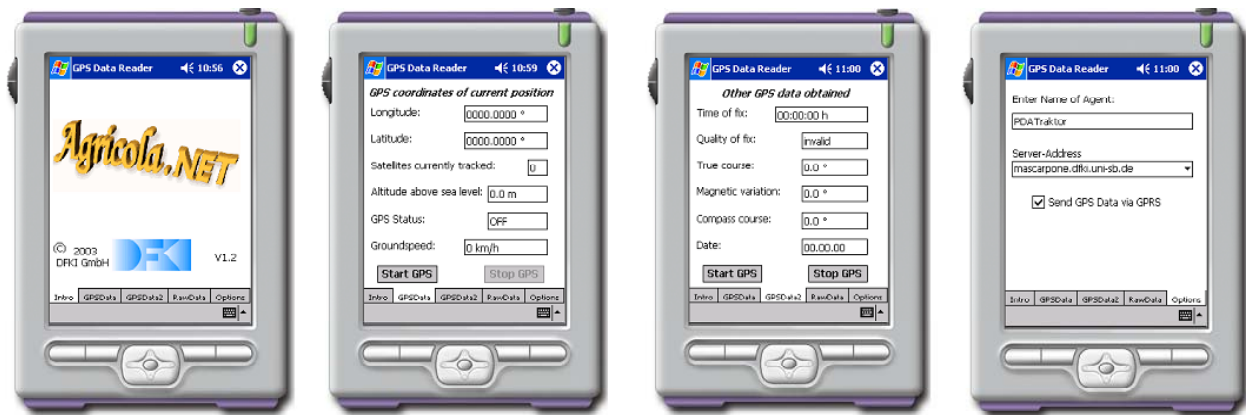


AGRICOLA.NET verwaltet zudem alle Adressen der aktiven Agenten des Multiagentensystems und kann von jedem Agenten im Netz erreicht werden. Dies ermöglicht die Einbeziehung von im Einsatz befindlichen Ressourcen in den kooperativen Planungsprozess über eine Satellitengestützte Positionsbestimmung (GPS). So können z.B. die Daten von PDA oder GPS-Agenten zunächst an den zentralen und statischen ANS-Agenten geschickt werden, mit der Zusatzinformation, diese Daten an den jeweiligen Ressourcen-Agenten des GPS-Agenten weiterzuleiten.

**Abbildung 2.** Technische Infrastruktur von AGRICOLA.NET

Die Topologie von AGRICOLA.NET kann sich ohne Einschränkung der Dienste verändern, beispielsweise bei Ausfall oder Neustart von Systemen registrierter Teilnehmer mit eventuell veränderter Lokation und dynamischer Vergabe von IP Adressen durch den *Internet Service Provider* (ISP).

Zur mobilen Datenerfassung und -übertragung werden speziell ausgestattete PDAs mit integrierten GPS-Empfängern und einer integrierten Mobilfunkeinheit zur Datenübertragung eingesetzt. Für einen Datenabgleich werden die betreffenden Daten per Funk über eine TCP/IP-Verbindung zu den jeweiligen Agenten übertragen. Analog können Agenten auf stationären PCs Daten an die Agenten auf den mobilen Endgeräten senden und so z.B. einen Anwender vor Ort rechtzeitig von einer Planänderung informieren. Der Einsatz von GPRS / GSM-Modems in Kombination mit PDA ermöglicht, dass die Teilnehmer sich jederzeit und von jedem Ort mit dem Internet verbinden können und auf die Dienste des Systems direkt zugreifen können.



**Abbildung 3:** Funktionen eines AGRICOLA Agenten auf einem PDA

In dem Anwendungsszenario wird eine lokale Bedarfs- und Maschineneinsatzplanung von Landwirten vor Ort mit aktuell verfügbaren, partiell globalen Planungen von geeigneten Anbietern bedarfsgerecht und *Just-in-Time* aufeinander abgestimmt. Die Maschineneinsatzplanung wird von den Anbietern durchgeführt und die Ergebnisse den Teilnehmern zugänglich gemacht. Die Planung einer Ressourcenverteilung kann durch eine Menge von sich dynamisch verändernden Bedingungen stark beeinflusst werden. Betrachtete Störfaktoren für die Planung in AGRICOLA sind

- Witterung
- Ausfall von Personal
- Ausfall von Maschinen
- Ausfall sonstiger notwendiger Materialien
- Biologische Störungen der Pflanzenproduktion

Auf Anfrage kann ein für eine Region zuständiger Koordinatoragent als eine globale Koordinationsstelle vermittelnd zwischen Landwirten und Maschinenbesitzern eingreifen. Eine derartige Vermittlung reicht von einer Auswahl von geeigneten Anbietern landwirtschaftlicher Nutzmachines bis hin zu Vorschlägen für die Maschineneinsatzplanung. Als Ergebnis von Bedarfs- und Einsatzplanungen der Teilnehmer werden entsprechende aufgabenorientierte Kooperationen gebildet. Eine orts- und zeitgenaue GPS-basierte Datenerfassung ermöglicht zudem eine präzise Abrechnung zwischen den Kooperationspartnern nach Abschluss eines Wertschöpfungsprozesses.

Das AGRICOLA.NET ist in der Forschungsgruppe Multiagentensysteme am Deutschen Forschungszentrum für Künstliche Intelligenz (DFKI) in Saarbrücken entwickelt und vollständig implementiert worden. Es basiert auf dem agentenbasierten Informations- und Handelsnetzwerk für die Forstwirtschaft CASA [2] und ist von Anwendern in der Region

erfolgreich getestet worden. Das Multiagentensystem und die Dienste des AGRICOLA.NET sind in der Programmiersprache C# unter .NET für die Betriebssystemplattformen Windows 98/2000/XP entwickelt worden. Alle Dienste des AGRICOLA.NET sind nach Installation der AGRICOLA Software auf dem jeweiligen Client über das Internet verfügbar. Die mobilen Dienste sind exemplarisch für den PDA "Pocket Loox" der Firma Fujitsu-Siemens AG entwickelt worden. Die mobile Datenübertragung vom PDA zum Netzwerk erfolgt mit Hilfe eines per Bluetooth mit dem PDA verbundenen GPRS-fähigen Mobiltelefons Nokia 6310i. Für die Ortsabhängigen, mobilen Dienste von AGRICOLA.NET ist der PDA mit einem zusätzlichen GPS-Empfänger ausgerüstet worden.

## **Ressourcenplanung in dynamischen Koalitionen von Agenten**

Um das Problem einer möglichst flexiblen Koordination von komplexen Abläufen und ihrer Planung in sich stetig ändernden Umgebungen zu lösen, wurden im Rahmen des Projekts effiziente Verfahren zur dynamischen Koalitionsbildung entwickelt und getestet. Ziel dieser Verfahren ist es, den einzelnen Agenten zu erlauben, rational kooperative Verbände (Koalitionen) zu bilden, mit deren Hilfe sie komplexe Aufgabenstellungen gemeinsam bearbeiten, die sie allein nicht oder nur partiell lösen können. In offenen Umgebungen können jedoch (a) die Agenten jederzeit einer Koalition beitreten bzw. sie verlassen, (b) die Aufgabenbeschreibung, wie auch die Ressourcen der Agenten sich fortwährend ändern und (c) die kommunizierten Informationen zum Teil unsicher bzw. vage sein. Es besteht daher das Problem, stabile Koalitionen zwischen Agenten zu bilden, ohne im Fall von Störungen des Systems ständig alle Koalitionen wieder neu verhandeln zu müssen.

Zur Lösung dieses Problems wurde ein Schema für eine *simulationsbasierte, dynamische Koalitionsformierung* (DCF-S) entwickelt, das als Grundlage für verschiedene DCF-S Koalitionsalgorithmen dient [1]. Die Kernidee des Schemas ist, dass jeder Koalitionsführer neue Koalitionen mit potentiellen Partnern erst simuliert und anschließend (falls eine ‚bessere‘ Koalitionsstruktur gefunden wurde) verhandelt. Dabei entsprechen Koalitionen sog. Holonen, d.h., hierarchisch strukturierten Agentenverbänden, die aufgabenorientiert miteinander in geeigneten Unterverbänden kooperieren.

Die Simulation derart bestimmter hypothetischer Koalitionsstrukturen erfolgt anhand der erlernten (unsicheren) Wissensbasis des Koalitionsführers einer jeden Koalition. Bevor eine solch hypothetische Koalition durch Verhandlungen realisiert wird, wird zwischen dem Risiko des Fehlschlags bei der Umsetzung der simulierten Struktur und dem Nutzen für die zu erweiternde Koalition abgewogen. Scheitert eine Verhandlung, treten Störungen auf, oder wurde keine Verbesserung durch die Simulation gefunden, werden solange neue Simulationen und Verhandlungen ausgeführt, bis eine Verbesserung hinsichtlich der Kosten-/Nutzenfunktion erreicht wird. Im Anschluss an die Koalitionsverhandlungen werden diese evaluiert und das erworbene Wissen der Agenten über ihre jeweilige Umwelt entsprechend aktualisiert.

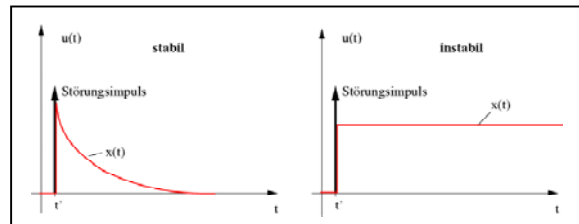
Das DCF-S Schema wurde in vier verschiedenen DCF-S Algorithmen realisiert, die sich im Wesentlichen in der Art und Weise unterscheiden, wie individuelles Wissen über die Verhandlung von den Agenten erworben und effektiv eingesetzt wird. Beispielsweise setzt der Algorithmus SVM-DHF-S ein um eine Support-Vector-Machine (SVM) erweitertes Reinforcement-basiertes Lernverfahren ein, um die für die jeweiligen Aufgaben geeignetsten Agenten gezielt statt nur rein zufällig nach „trial-and-error“ auszuwählen.

## **Stabilität von dynamischen Kooperationen zwischen Agenten**

Was bedeutet „Stabilität“ von Multiagentensystemen? Welche Maße für Stabilität sind auf die sich im AGRICOLA System dynamisch bildenden Agentenverbände sinnvoll anzuwenden?

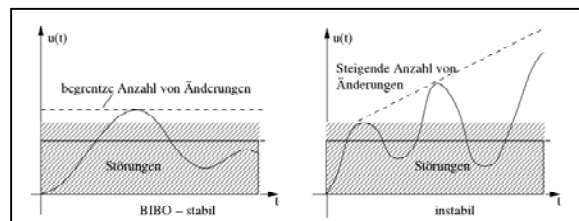
In Anlehnung an klassische Definitionen von Stabilität, wie beispielsweise die eines gedämpft schwingenden Systems in der Elektrotechnik, kann die Stabilität von Multiagentensystemen anhand von unterschiedlichen Störfällen in ihren Aktionen beschrieben werden. Generell besitzt ein stabiles System die Fähigkeit sich selbst zu regulieren und dem äußeren Einfluss von Störungen zu widerstehen bzw. nach einmaligen oder kontinuierlichen Störung in angemessener Zeit wieder zu einem wohldefinierten originären Zustand zurückzukehren. So verharrt beispielsweise ein stabiles Multiagentensystem solange in seinem Ruhezustand, bis es durch Störungen (Ausscheiden oder Neueintritt von Agenten, Modifikation von bereits eingeplanten Aufgaben) „angeregt“ wird. In Abhängigkeit zu der jeweils betrachteten Störungsart werden im Folgenden verschiedene Arten von Stabilität für Kooperationen zwischen Agenten des Multiagentensystems von AGRCIOLA.NET verwendet.

**Asymptotische Stabilität.** Ein Multiagentensystem ist asymptotisch stabil, wenn es nach *einer* Störung mit fortschreitender Zeit wieder in eine „Ruhelage“ kommt, in der keinerlei Änderungen der Strukturen (Agentenverbände) mehr erforderlich sind.

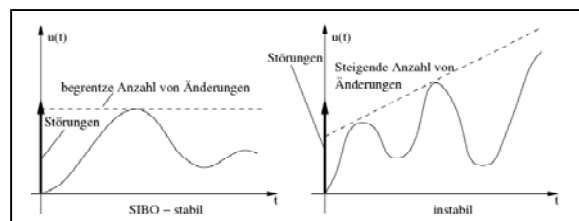


Alternativ kann man die asymptotische Stabilität eines Systems an der Systemreaktion auf eine begrenzte Menge von Störungen innerhalb eines Zeitintervalls untersuchen. Für die Anwendung im Projekt bedeutet dies, dass mit einem asymptotisch stabilem Agentensystem mit begrenztem Zeitaufwand eine neue Verteilung der Aufgaben auf die aktuell verfügbaren Maschinen der Fuhrparks der Landwirte bestimmt werden kann. Je schneller die AGRICOLA Agenten sich in Reaktion auf eine Störung in Koalitionen für geeignet modifizierte Maschineneinsatzpläne umorganisieren können und je geringer dabei die Anzahl noch offener Aufgaben wird, desto stabiler wird das System. Damit steigt auch der Mehrwert des Systems für die betreffenden Landwirte hinsichtlich Planungsflexibilität, Auslastung, Zeit- und Kosteneffizienz des Maschineneinsatzes bei Störungen des Betriebes.

**Bounded Input Bounded Output (BIBO) - Stabilität.** Ein System ist BIBO-stabil, wenn *jede begrenzte* Störung mit fortschreitender Zeit nur zu einer *begrenzten Änderung* der Agentenverbände im System führt. Eine Änderung ist begrenzt, wenn lediglich eine beschränkte Anzahl der Agenten in andauernden Verhandlungen involviert ist, so dass eine obere Schranke  $\epsilon$  für die Anzahl der sich restrukturierenden Agenten angegeben werden kann. In der Anwendung bedeutet dies, dass im Falle von gehäuften Störungen des Maschineneinsatzes aufgrund von Witterung, Personal- oder Maschinenausfall, ein BIBO-stabiles Agentensystem zumindest einen Teil der Einsatzpläne für die Landwirte unverändert ausführbar belässt.



**Single Input Bounded Output (SIBO)-Stabilität.** Ein System ist SIBO-stabil, wenn es nach *einer* beliebigen Störung mit fortschreitender Zeit zu einer begrenzten Änderung der Agentenverbände im System führt, so dass eine obere Schranke  $\epsilon$  für die Anzahl der sich restrukturierenden Agenten angegeben werden kann. Analog der BIBO-Stabilität kann mit Hilfe eines SIBO-stabilen Systems gewährleistet werden, dass zumindest ein Teil der Maschineneinsatzpläne von einer Störung unbeeinflusst bleibt und weiterhin ausgeführt werden kann. Z.B. führt der Ausfall eines Mähdreschers unter Umständen dazu, dass es in einer Region nicht mehr möglich ist, alle Felder termingerecht zu bearbeiten. Es



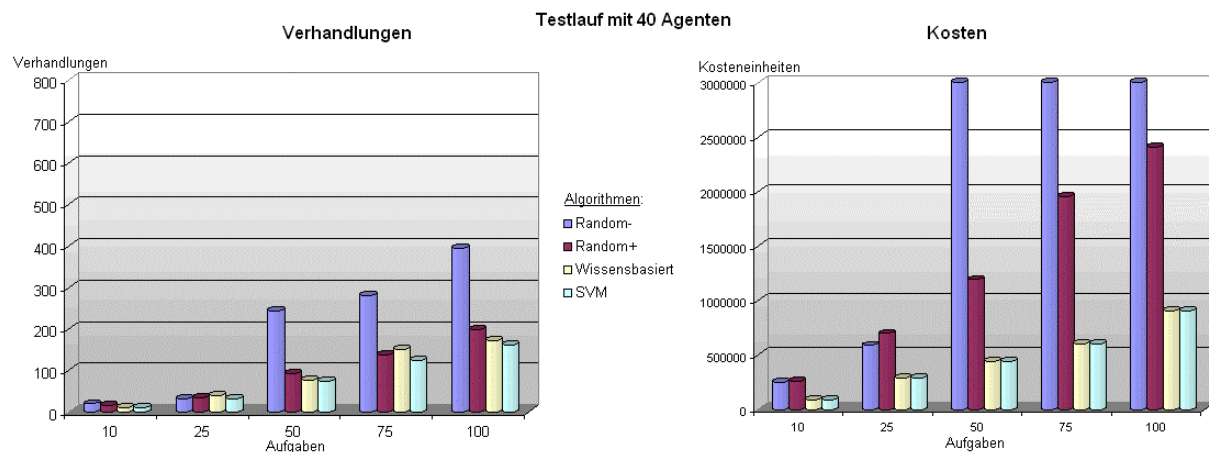


wird dann solange keine Ersatzressource der betreffenden Region zugewiesen, solange es keine vollständige Lösung des Problems gibt. SIBO-Stabilität des AGRICOLA Systems bedeutet, dass die Zahl der entsprechend unbearbeiteten Felder hinsichtlich der den Agenten vorgegebenen Kosten und persönlich Präferenzen minimal ist.

Wie schnell ein Multiagentensystem auf eine Störung reagiert und neue, holonische Koalitionsstrukturen bildet, hängt im Wesentlichen davon ab, wie gut das Wissen jedes Agenten über die Umwelt ist. Je genauer die Prognosen/Abschätzungen der Eigenschaften der Agenten sind, desto präziser können potentielle Agenten für die Bearbeitung einer Aufgabe ausgewählt werden und somit unnötige Verhandlungen vermieden werden. Insbesondere werden so weniger destabilisierende Umstrukturierungsversuche vorgenommen, die aufgrund falscher Einschätzungen durchgeführt, jedoch mit realistischen Annahmen nie begonnen würden.

## Evaluation von dynamischen Koalitionsbildungen

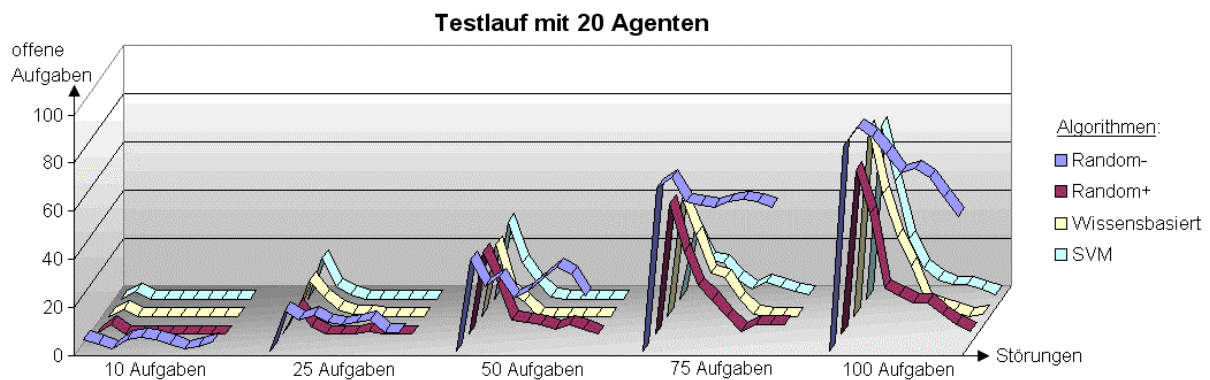
Erste umfangreiche Testläufe der im Projekt entwickelten DCF-S Koalitionsalgorithmen haben ihren Nutzen in Theorie und Anwendung bereits gezeigt. Es wurde vor allem untersucht, inwieweit sich die Umstrukturierungen der Koalitionen auf die Stabilität des gesamten Systems auswirken, ob bereits einzelne Störungen zu einer totalen Destabilisierung der Holonen führen und welche Effizienz hinsichtlich der notwendigen Anzahl von Verhandlungen, sowie Kosten/Nutzen der Koalition erreicht werden kann. Die Evaluierung der Koalitionsalgorithmen wurde mit verschiedenen Mengen von bis zu 40 Agenten durchgeführt, die jeder für sich unterschiedliche Fähigkeiten zur Bearbeitung gegebener Aufgaben besitzen. Pro Agentenmenge wurden fünf Testreihen mit einer unterschiedlichen Anzahl (10, 25, 50, 75 und 100) von Aufgaben ausgewertet.



**Abbildung 4.** Aufwand und Kosten von dynamischen Koalitionsverhandlungen

Abbildung 4 zeigt die Anzahl der Verhandlungen in Relation zu den vier Koalitionsalgorithmen als Instanzen des DCF-S Schemas sowie das entsprechende Verhältnis der Kosten, die einer Koalition durch die resultierende Aufgabenverteilung entsteht. Die Analyse der Ergebnisse von insgesamt mehreren hundert Testläufen ergab, dass durch den Einsatz des SVM-DHF-S Algorithmus im Mittel zwischen 1.56 und 2.48 Verhandlungen pro Aufgabe ausgeführt werden mussten (je nach dem Verhältnis der unterschiedlichen Operationen zur Bildung von hypothetischen „simulierten“ Koalitionen), um den kostenoptimalen Agenten zu finden, der diese Aufgabe ausführen kann. Für das AGRICOLA-System bedeutet dies, dass die Agenten in ihren Koalitionen bei Auftreten von Störungen die geplante Bearbeitungsreihenfolge schnell und derart modifizieren, dass nur benachbarte

Äcker bearbeitet werden. Damit können beispielsweise kosten- und zeitaufwändige Transporte eines Mähdreschers zwischen den betreffenden Feldern entfallen.



**Abbildung 5.** Stabilität von dynamisch gebildeten Koalitionen bei Störungen

Abbildung 5 zeigt die Ergebnisse eines Testszenarios, in dem 20 AGRICOLA Agenten für die Bearbeitung einer verschiedenen Anzahl von Aufgaben in Zeitintervallen von 10 Sekunden auftretenden und zufälligen Arten von Störungen unterworfen waren. Arten von Störungen sind die Eingabe neuer Aufgaben, das Löschen bereits eingeplanter Aufgaben, Hinzufügen von neuen Agenten in die Agentengesellschaft und der Ausfall von Agenten, denen bereits Aufgaben zugewiesen wurden. Diese Störungen können Umstrukturierungen von Koalitionen (Personen- und Maschinenagenten) verursachen, falls Aufgaben nicht mehr vollständig bearbeitet werden können oder kosteneffizientere Koalitionen möglich sind. Hinsichtlich der Stabilität des System ist von Interesse, wie viele der gegebenen Aufgaben nach dem Auftritt einer Störung wie lange offen bleiben, d.h. nicht in Bearbeitung sind. Die Testergebnisse zeigen nicht überraschend, dass sich mit zunehmendem Wissen und Lernfähigkeit das Gesamtsystem asymptotisch stabil mit Bezug auf die Anzahl offener Aufgaben verhält. Dies unterstreicht die Einsatztauglichkeit des Systems im betrachteten landwirtschaftlichen Anwendungsszenario, in der bereits mittelfristige Maschineneinsatzplanungen als nicht möglich erachtet werden. Die asymptotisch stabile Umorganisation der AGRICOLA Agenten mit entsprechender Umplanung der durch sie vertretenen Ressourcen (Maschinen und Personal) garantiert, dass sogar bei mehreren Störungen innerhalb kürzester Zeit noch Just-In-Time ausführbare Einsatzpläne für die Maschinen der Landwirte erzeugt werden. Dabei bleibt ein Maximum an Kontinuität bezüglich der Planausführung gewahrt, d.h. die Umstellung der Maschineneinsatzpläne wird so gering wie möglich gehalten.

## Fazit

Das AGRICOLA.NET ermöglicht eine kosteneffiziente Auslastung von landwirtschaftlichen Maschinenfuhrparks bei kurzfristigen Änderungen in der Planungsumgebung, wie beispielsweise Witterungsumschwung, sowie Ausfall von Maschinen und Personal. Insbesondere wird die Satellitengestützte Positionsbestimmung (GPS) von im Einsatz befindlichen Ressourcen in den von AGRICOLA Agenten für die einzelnen Landwirte koordinierten Planungsprozess einbezogen. Die im Projekt entwickelten Verfahren eignen sich insbesondere für große Agentengesellschaften. Das AGRICOLA Agentensystem hat sich in der Regel als asymptotisch, sowie SIBO- und BIBO-stabil erwiesen: Es findet nach jeder anwendungsbezogenen Störung in der kooperativen Planung eine optimale Lösung hinsichtlich gegebener Kosten-/Nutzenkriterien in angemessener Zeit. Insgesamt bietet das AGRICOLA.NET mit seinen mobilen Planungsdiensten eine hoch innovative,

softwaretechnische Unterstützung für eine bedarfsgerechte und flexible Einsatzplanung von Ressourcen in der landwirtschaftlichen Wertschöpfung.

## Literatur

Klusch, M., Gerber, A. (2002): Dynamic Coalition Formation among Rational Agents. IEEE Intelligent Systems, 17(3), May/June 2002

Klusch, M., Gerber, A. (2002): CASA: Agent-based Integrated Services Network for Timber Production and Sales. IEEE Intelligent Systems, 17(2), January/February 2002.



**Andreas Gerber** ist Wissenschaftler in der Forschungsgruppe Multiagentensysteme am Deutschen Forschungszentrum für Künstliche Intelligenz in Saarbrücken. Aktuelle Forschungsschwerpunkte liegen in den Gebieten Verteilte KI, Agentenbasierte Koordinationsmechanismen und integrierte Dienste für elektronische Märkte. [www.dfki.de/~agerber/](http://www.dfki.de/~agerber/); agerber@dfki.de



**Matthias Klusch** ist leitender Wissenschaftler in der Forschungsgruppe Multiagentensysteme am Deutschen Forschungszentrum für Künstliche Intelligenz in Saarbrücken. Aktuelle Forschungsschwerpunkte liegen in den Gebieten Semantisches Web und Dienste, Intelligente Agenten und Informationssysteme im Internet, sowie verteilte Wissensentdeckung und Anwendungen von Quantum Information Processing. [www.dfki.de/~klusch/](http://www.dfki.de/~klusch/); klusch@dfki.de



## CASCOM: Mobile Emergency Medical Assistance

T. Möller, H. Scholdt, A. Gerber, M. Klusch: Next Generation Applications in Healthcare Digital Libraries using Semantic Service Composition and Coordination. *Health Informatics*, 12 (2), pages 107-119, SAGE publisher, 2006.



## Next-generation applications in healthcare digital libraries using semantic service composition and coordination

*Thorsten Möller, Heiko Schuldt, Andreas Gerber and Matthias Klusch*

Healthcare digital libraries (DLs) increasingly make use of dedicated services to access functionality and/or data. Semantic (web) services enhance single services and facilitate compound services, thereby supporting advanced applications on top of a DL. The traditional process management approach tends to focus on process definition at build time rather than on actual service events in run time, and to anticipate failures in order to define appropriate strategies. This paper presents a novel approach where service coordination is distributed among a set of agents. A dedicated component plans compound semantic services on demand for a particular application. In failure, the planner is reinvoked to define contingency strategies. Finally, matchmaking is effected at runtime by choosing the appropriate service provider. These combined technologies will provide key support for highly flexible next-generation DL applications. Such technologies are under development within CASCOM.

### Keywords

agent systems, compound service execution, service composition planning, service coordination, service matchmaking

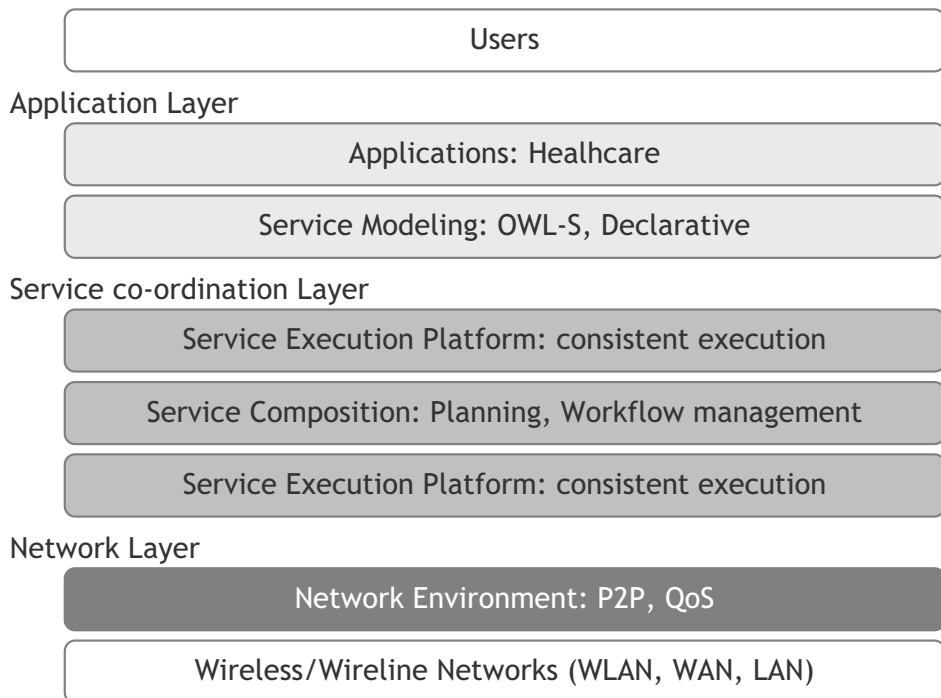
### Introduction

The paradigm of service-oriented computing encapsulates functionality and makes use of it in a well-defined way by providing standardized interfaces for access and description. Digital libraries (DLs) in general and healthcare digital libraries in particular increasingly exploit services that encapsulate functionality and/or data. Enriching the conventional

syntactic description of a service with information on its semantics allows for a more focused search for appropriate services in a large-scale network. However, complex (healthcare) DL applications usually require the combination and composition of several (semantic) services into compound services or processes. The traditional workflow and process management approach considers the definition of a process at build time but does not take into account the service instances that are actually available at run time. Failures have to be anticipated and appropriate failure handling also has to be defined at build time. In this approach, unforeseen failures cannot be handled. In addition, usually a centralized approach is followed that implies a single point of failure and that does not scale well with the number of processes to be executed and the number of semantic services available.

The goal of the CASCOM project (Context-Aware Business Application Service Co-ordination in Mobile Computing Environments) [1] is to overcome these limitations by implementing, validating, and testing a value-added supportive infrastructure for business application services for mobile workers and users across mobile and fixed networks. The driving vision of CASCOM is that ubiquitous business application services are flexibly coordinated and pervasively provided to the mobile worker/user by intelligent agents in dynamically changing contexts of open, large-scale, and pervasive environments. Validation and testing of the CASCOM architecture will take place in a real-world healthcare DL setting.

The CASCOM architecture is divided into several layers (see Figure 1). The planned technological innovations at each layer can be summarized as follows. The main outcome



**Figure 1** Layered CASCOM architecture

of the network layer is a generic, secure, and open intelligent agent-based peer-to-peer (IP2P) network infrastructure taking into account varying quality-of-service (QoS) properties of wireless communication paths, limitations of resource-poor mobile devices, and contextual variability of nomadic environments. IP2P environments are extensions to conventional P2P architectures with components for mobile and *ad hoc* computing, wireless communications, and a broad range of pervasive devices. This means that heterogeneous and dynamic systems have to be linked together (e.g. smart phones, PDAs, computers), and that the infrastructure should be robust with no major point of failure while the deployment and maintenance effort should be minimal. Conceptually, the IP2P layer will be built to permit seamless mobility to the user. In contrast to existing wireless technologies, users are then able to roam through different physical network technologies like GSM, UMTS, and WLAN. One essential approach to achieve this is to build a network abstraction (overlay network) on top of the network infrastructure. The main outcomes of the service coordination layer are (1) flexible semantic web service discovery including adaptive service QoS-oriented service matching and usage of distributed semantic web service directories (DSD), (2) dynamic context-aware semantic web service composition including resource-efficient interaction between DSD and service composition planner, fault-tolerant interleaving of planning and service execution, and (3) secure service execution and monitoring providing service data consistency.

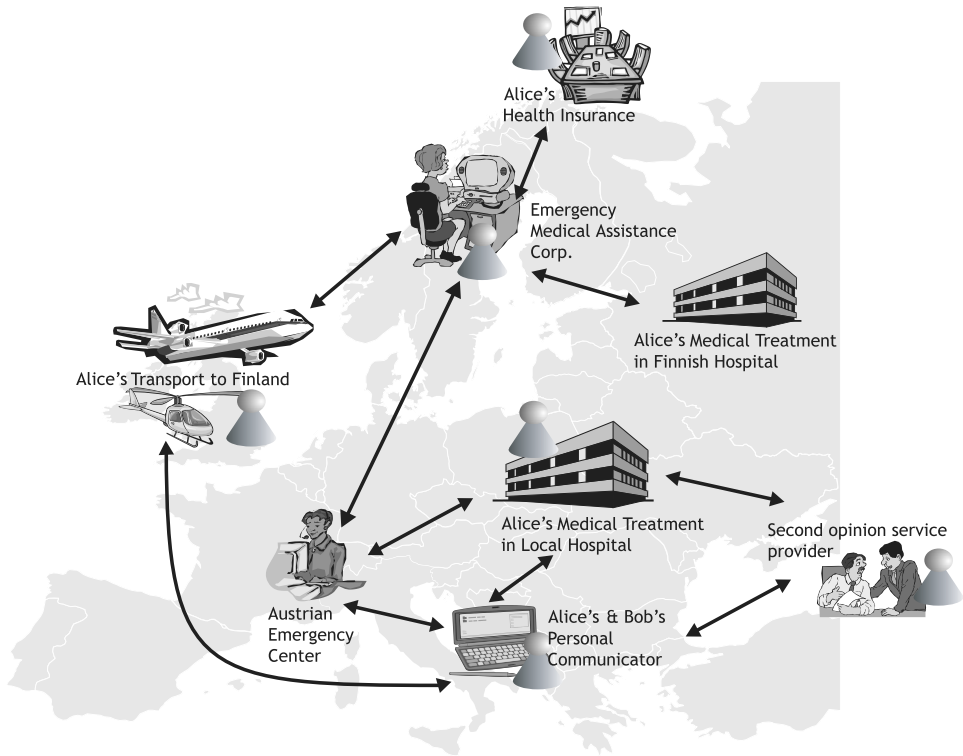
In this paper, we present the novel CASCOM agent-based approach to process generation and execution. Process execution is distributed among a set of cooperating service provider agents. Each agent works off its part of a process (i.e. locally invokes the required services) and then forwards control to the next agent, which is then in charge of continuing process execution. Processes are not defined statically. Rather, a dedicated planning component composes semantic services based on the particular goals of an application. In case of failure, the planner is reinvoked in order to define contingency execution strategies. Finally, instance matchmaking is done at run-time by choosing the most appropriate agent (according to pre-defined QoS constraints) among a set of agents qualifying for the execution of a particular semantic service.

The focus of this paper is the interaction of planning, matchmaking, and execution of processes consisting of invocations of semantic web services. The combination of these technologies will provide key support for highly flexible next-generation DL applications that make use of semantic service descriptions. In particular, we apply these technologies to semantic web service composition in a healthcare DL application (emergency assistance), which supports people travelling in foreign countries with the healthcare services they need when suddenly suffering from illnesses and needing medical treatment and care. This of course requires access to services and data in a healthcare DL.

The paper is organized as follows. In the next section we present an emergency assistance scenario which we focus on in CASCOM. Subsequent sections introduce the different technologies needed to support this scenario and show how these can be seamlessly integrated. A discussion of related work and our conclusions complete the paper.

## Sample healthcare application

In the following, a sample business application service scenario 'Emergency Assistance' is described (see Figure 2). The scenario is based on the fact that people on the move, e.g.



**Figure 2** Emergency assistance application

travelling in foreign countries for business or holidays, may get into situations where they need medical assistance because of a sudden disease or emergency. Currently, these sorts of episodes are neither tackled nor realized in this form in practice and no software system is presently in widespread use to address them.

Alice and Bob, tourists from Finland, are abroad on a countryside journey in Austria during their summer vacation. They carry a PDA, already equipped with the CASCUM mobile agent suite. Suddenly after some days, Alice is seriously suffering from unknown pain in the upper part of her body. For this reason, she wants to immediately call a hospital or physician. After activation of the PDA, the agent immediately finds out the contact information of a local healthcare institution near them.<sup>1</sup> Additionally, the agent gives them the contact information of the Finnish representative of the Emergency Medical Assistance (EMA) service centre that takes care of the remote support of the patient. Alice decides to immediately go to the local hospital. The agent on the PDA also supplies them with information on how to get there. This could be either a map showing their current location and the healthcare centre location, or a phone number for a local taxi, or instructions for a connection via public transportation. On arrival and check-in at the local hospital, Alice has to manually answer some questions about her personal data because the healthcare institution does not provide the infrastructure and services to plug her PDA into the local information system, i.e. to exchange initial data. During the first

examination by the local emergency physician it turns out that Alice had either a silent heart attack or angina pectoris, but the physician is not sure about the diagnosis and wants to obtain a second opinion. Even Bob and Alice are concerned about the doubtful situation. Bob now uses the PDA to access a second opinion service by forwarding all information available so far. If the hospital had had the CASCOM infrastructure installed, the local physician could have used a local PC or PDA to access the second opinion service. However, in this case the agent on the PDA finds out the contact information of a specialized cardiologist and establishes a connection. After assessment of the situation, the doctors decide that Alice should be transferred soon to a hospital with advanced cardiac life support to undertake thorough examination. Alice says that she wants to be transferred back to a hospital in her home country, Finland. As a result, the EMA service centre will be contacted to organize the transfer by using the PDA: remember that contact information was transferred before. Now, the EMA agent first automatically investigates possible travel arrangements (depending on the medical circumstances and the geographical distance, the agent may eventually come up with a decision on whether to use regular flights, a car, or some other form of transportation). Second, the agent informs all people that are involved during the transfer (doctors and escorts). Third, it contacts Alice's insurance company to make sure that her insurance will cover all possible transportation costs. In addition, the agent could possibly automatically contact the Finnish hospital (which participates in the CASCOM network) to make further arrangements. Back in Finland, Alice is treated at a sophisticated cardiac hospital. After 2 weeks of recovery she finally uses her PDA to send a 'thank you' to all the people involved with her medical case.

As can be seen in this scenario, people (patients) not only need medical treatment, but also need information as well as (sometimes) transportation assistance. Furthermore, assistance in the form of information is also required by the physicians, hospitals, and healthcare professionals involved. One straight implication of these complex requirements is the need for on-demand *initiation*, *composition*, *coordination*, and *supervision* of various activities represented mostly through non-human actors, like agents and services, but also through persons.

## Service coordination layer

The process of service coordination is usually considered to encompass all activities that are devoted not only to the description but also to the discovery, composition, and execution of services. In subsequent sections, we introduce the basic technologies needed for the requirements derived from the healthcare application scenario.

One outstanding requirement of the scenario is its demand for coverage of large areas, i.e. it is supposed to spread over many different countries. As a consequence, the number of users (service requesters) and service providers is expected to range from many thousands to millions. Due to this fact, it is impossible to know all services, their functionality, and their distribution beforehand. As an initial step, this requires appropriate methods to discover and register services (either centralized or decentralized). In particular, this demands matchmaking – the selection of appropriate web services using semantic similarity. For instance, in case of emergency, the user needs to find a hospital or

emergency centre near him or her. When several distinct but similar service providers exist which are all able to return this information, one has to be chosen. For this, quality-of-service criteria can be exploited (e.g. one service might be able to find emergency centres closer to the current location than another service). As a second step, the scenario requires service composition planning – the composition of several web services into processes. Typical usage cases within the scenario involve interaction with several service providers. For instance, transportation of a patient back to his or her home country may require interaction with different service providers to arrange the most suitable transportation. Finally, the service coordination layer must include execution, i.e. a runtime environment for compound services.

### ***Service matchmaking***

The service matchmaking functionality provides the means to compare the semantics specified for services, thus allowing the detection of semantically equivalent/similar services. Several approaches to the sophisticated semantic matchmaking of web services have been proposed that rely on ontology-based languages (e.g. OWL-S [2], WSMO, and Annotated WSDL [3]) and are grounded with formal semantics such as description logics [4]. The most important principle of semantic matchmaking is that semantics of words used in the description of web services are formally defined in ontologies. Those ontologies can be exploited by matchmaker agents to determine the degree of semantic matching of advertised services with a given service request.

For semantic matching of services specified in OWL-S, the OWLS-MX for hybrid matchmaking has been developed at DFKI. It takes any OWL-S service description as a query, and returns an ordered set of relevant services that match the query, each annotated with its individual degree of matching (DOM) and syntactic similarity value. The user may extend the query by specifying the desired DOM and a syntactic similarity threshold. OWLS-MX first classifies the service query I/O concepts into its local service I/O concept ontology. As usual, we assume that the type of computed terminological subsumption relation determines the degree of semantic relation between pairs of input and concepts. Attached to each concept in the concept hierarchy is auxiliary information on whether it is used as an input or output concept by any service that has been registered at the matchmaker. The corresponding I/O lists of unique service identifiers for input and output concepts are then used by the matchmaker to compute the set of relevant services that match the given query according to its five matching filters. In particular, OWLS-MX determines pairwise not only the degree of logical match but also the syntactic similarity between the terminological expressions built by unfolding each of the considered query and service input (output) concepts in the local matchmaker ontology. In this way, logical subsumption failures produced by the integrated description logic reasoner of OWLS-MX are tolerated, if the syntactic similarity value computed by means of a specific information retrieval similarity metric is sufficient (i.e. exceeds the given threshold).

### ***Service composition planning***

The service composition functionality supports the context dependent composition of compound, value-added services whenever no appropriate single service can be found

during matchmaking. In CASCOM we intend to use OWLS-Xplan [5]. OWLS-Xplan takes a set of available OWL-S services, related OWL ontologies, and a query as input, and returns a plan sequence of composed services that satisfies the query goal. For this purpose, it first converts the domain ontology and service descriptions in OWL and OWL-S, respectively, to equivalent problem and domain descriptions. The problem description contains the definition of all types, predicates and actions, whereas the domain description includes all objects, the initial state, and the goal state. Both descriptions are then used by the AI planner Xplan to create a composition plan that solves the given problem in the actual domain.

Xplan is a heuristic hybrid search planner based on the FF planner [6]. It combines a guided local search with graph planning and a simple form of hierarchical task networks to produce a plan sequence of actions that solves a given problem. This yields a higher degree of flexibility than pure hierarchical task-reduction planning (HTN), and the use of predefined workflows or methods improves the efficiency of the FF planner. In contrast to the general HTN planning approach, a graph-plan-based planner is guaranteed to always find a solution independent of whether the given set of decomposition rules for HTN planning would allow building a plan that contains only atomic actions (services). In fact, any graph-plan-based planner would test every combination of actions in the search space to satisfy the goal which, of course, can quickly become prohibitively expensive. Xplan combines the strengths of both approaches. It is a graph-plan-based planner with additional functionality to perform decomposition like an HTN planner.

The Xplan system consists of the XML parsing module, a pre-processing module, the planning core, and the replanning module. The latter is used to readjust outdated plans during execution time (see later). After the domain and problem definitions have been parsed, Xplan compiles the information into memory efficient data structures. A connectivity graph is then generated, which contains information about connections between facts and instantiated operators, as well as information about numerical expressions which can be connected to facts. This connectivity graph is maintained during the whole planning process and serves as a kind of efficient lookup table for the actual search.

### **Service execution**

The service execution system (SES) executes compound services as they are generated by the service composition planner agent (SCPA). For process execution, we first assume that a compound service contains an arbitrary number of service invocations whereby the composition structure is equal to an acyclic ordered graph, i.e. combined sequential and parallel flows together forming processes as denoted in [7]. Second, as a basis for correct process execution, each service invocation is assumed to be atomic and compensatable. This means that the effects of a service can be undone later. Otherwise, unwanted side effects of aborted or compensated executions may remain and an at-most-once execution semantic could not be guaranteed. For services which do not comply with the atomicity requirement, we assume that a wrapper can be built which adds this functionality [8]. Third, we assume that services are stateless, i.e. that they never have to remember anything beyond interaction. In our approach, process state (i.e. the intermediate results) is solely stored by the execution system. Finally, our approach considers the crash failure model, which means that components such as services and machines may fail by prematurely halting their execution.



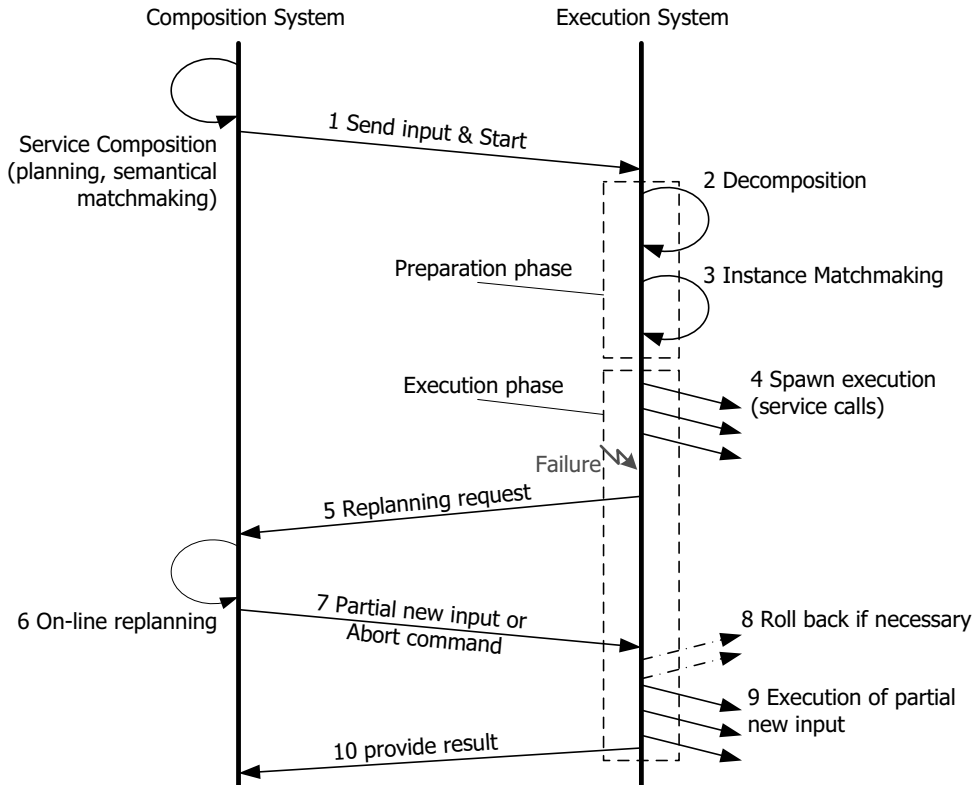
The execution system is based on the principles of the OSIRIS (Open Service Infrastructure for Reliable and Integrated process Support) process management system [9]. Within OSIRIS, aspects of agent-oriented systems were introduced to fit into the CASCOM infrastructure. In particular, the execution system consists of one or more federated execution agents organized in a peer-to-peer manner, meaning that no central execution coordinator is required. To accomplish this, every agent implements a process manager which locally invokes services and which coordinates execution basically by forwarding control and data to the next agent(s). Furthermore, we distinguish between two types of service execution agents (SEAs): service provider agents and standard agents. The difference is that the former is locally attached to one or more service instance(s) on the same machine (i.e. agent and service(s) run on the same device), whereas the latter may run on any computing device – especially mobile devices – and calls services remotely. Nevertheless, both implement execution functionality completely according to our execution requirements.

The execution first involves decomposing the process model into its atomic execution units. An execution unit contains a service invocation  $s$  and links to all the services that are the direct successors of  $s$ . In addition, for failure handling purposes, information on the predecessors of  $s$  is also needed. This is important in order to determine which services need to be compensated (i.e. which effects need to be undone) when a failure during process execution occurs. This means that for every service only *links* to adjacent services are of interest. All in all, the units provide execution agents with all the information they require to execute a service and to do forward navigation afterwards. The explicit distinction between control and data flow enables optimal interaction paths with as few communication efforts between execution agents as possible.

## Integration of service matchmaking, composition planning, and execution

As noted earlier, the SCPA acts as the client for SES. Consequently, our combined interaction and execution model consists of the following steps (see Figure 3 for the model and step numbers). Before actual execution starts, the SCPA creates a new process using a planning algorithm and semantic matchmaking to employ some of the services in the domain according to their service descriptions. It then sends input (the newly generated process type) to SES and orders execution *start* (1). Note that the process type contains all the necessary information for instantiation; just the individual service types still need to be bound to instances. Now, the execution preparation phase starts. Since a process instance is not suitable for execution on the physical layer, a detailed execution plan has to be created [10]. The most important part of this plan is the *decomposition* of the process into its execution units (2). The following step is called *instance matchmaking* at runtime (3), where a concrete service provider instance of a given type will be selected based on most current QoS criteria like average load or execution costs.<sup>2</sup>

The preparation phase is finished by distributing required information to the execution agents, such that they can forward control on their own during execution. Then SES spawns service *execution* on behalf of the input (4), i.e. the execution phase starts. During execution, failures might happen, for example service instances or other infrastructure components might crash. In such a situation, execution cannot terminate or at least



**Figure 3** Interaction and execution model

cannot continue without some recovery mechanism. In classical transactional systems, this would lead to an abort of the global transaction (i.e. all side effects created so far would be undone or compensated) and some external logic would have to decide what to do next. In our approach, a crash failure situation does not necessarily end in the abortion of process execution. After a failure, SES temporarily freezes process execution. In particular, if parallel execution paths exist, all of them will be frozen. Furthermore, SES knows about the current process state and the side effects created. Then, SES transmits this information to the SCPA and *requests* online contingency *replanning* (5); remember that the original process execution goal still holds. Starting from the stop point, the planner now tries to fix the problem by searching for an alternative path (6) – usually by employing semantically similar services. If SCPA succeeds in composing a partial new process of remaining activities, this *new process fragment* will be sent to SES (7). Otherwise, if it was not possible to find an alternative path, SCPA sends an *abort* command to SES. Consequently, SES is then obligated to *roll back* the process side effects completely (8) – which is possible according to our assumption of atomic, compensatable services. In the former case, SES can continue execution with the new process fragment. In order to be able to do this, it first has to replace the old remaining process fragment (which is now obsolete) with the new one. This is accomplished by starting a new sub-preparation phase, whereby decomposition, instance matchmaking, and distribution to the service provider agents

again take place (i.e. update of the execution plan). Afterwards, *execution* continues (9). Replanning is also required if the original goal for which the process has been generated is altered. If the new process fragment requires the partial undoing of side effects because of its changes (i.e. utilization of other services), this will be done immediately before continuation. Finally, when execution has finished, the *result* will be sent back to SCPA (10). In order not to block resources endlessly during online replanning, we use a timeout-based approach: when there is no reply from SCPA to SES until the timeout (because SCPA has crashed or connection has been lost), SES aborts the current process execution and tries to notify SCPA about that.

One aspect of our interaction model that is still open for discussion is whether we allow for indefinite replanning phases. By allowing indefinite replanning phases it is evident that execution theoretically might never terminate. On the other hand, and with the presented emergency assistance scenario in mind, the probability of high numbers of replanning cycles falls with the number of cycles. For emergency service providers it is crucial that their services are constantly available; if not, nobody would develop trust in such applications. However, because of different service providers, similar services (alternatives) are expected to exist. Thus, it is evident that either an alternative is available early, which eventually leads to success, or no alternative exists and execution stops entirely. A simple approach to address this issue is to fix a maximum replanning cycle count for the implementation. A more sophisticated approach would be the definition of *execution progress*. If there is no significant progress towards the execution goal even though both SCPA and SEA are not inactive (i.e. execution stagnates), its value converges to zero. Thus, it is possible to detect stagnated executions and abort them eventually. All in all, the decision about which policy should be used for replanning phases should not be made without taking the target application into account.

In the scenario presented earlier, Alice and Bob are first required to state the goal of the process they wish to be executed (e.g. transfer to a hospital, receive treatment from there while giving the local physician access to Alice's health record). Then, by combining matchmaking and planning, a process tailored to Alice's needs is generated and executed. In the case of failures or changes in the environment, planning is reinvoked and the process is changed (and executed) accordingly.

## Related work

Similar to CASCOM, the ARTEMIS project (Semantic Web Service-Based P2P Infrastructure for the Interoperability of Medical Information) [11] aims at supporting healthcare applications by means of dedicated semantic web services. However, ARTEMIS focuses on providing single semantic web services and addresses standards and interoperability issues of these services, while the goal of CASCOM is to provide value-added, composite services in order to support sophisticated *ad hoc* process-based healthcare applications in IP2P environments.

Issues of service composition (planning) and coordination are currently widely addressed in research, especially if extension to semantic description of web services comes into play. Some ontology-based approaches to semantic service matchmaking that have been proposed in the literature are LARKS [12], OWLS/UDDI [13], MATCHMAKER-Service [14], RACER [15], and HotBlu [16]. Other approaches are either process based (e.g.

High-Precision Service Retrieval Service [17]), peer based (e.g. Semantic Web Services P2P Discovery Service [18]) or hybrid (e.g. the Recursive Tree Matchmaker [19]). Alternative approaches include graph-based matching methods such as those presented in [20] and [16]. Furthermore, there are currently very few approaches and software tools available for OWL-S-based service composition planning, such as OWL-S Composition Planner using SHOP2 [21], logic-based DAML-S composition planning [22], the DAML-S workflow composer [23], and a Petri-net approach in which an OWL-S service description is automatically translated into Petri nets [24].

Finally, the issue of service execution is widely addressed in classical research domains like transactional information systems and process management. The OSIRIS infrastructure on which the SES is built provides a scalable distributed process navigation platform. To achieve this, it combines a rich set of aspects. Based on the hyperdatabase vision [25], ideas from process management, peer-to-peer networks, database technology, and Grid [26] infrastructures have been combined. Similar to OSIRIS, where processes are running within a peer-to-peer community that is established by the individual service providers, the MARCAs presented in [27] are service providers acting as peers. Finally, Pleisch and Schiper [28] provide a general overview on fault-tolerant agent-based (process) execution.

## Conclusions

In this paper, we have presented the CASCOM approach to providing access to functionality and data, encapsulated by semantic services, in a healthcare digital library. *Ad hoc* process-based applications are supported by seamlessly combining sophisticated service composition planning, service matchmaking, and agent-based distributed service execution. The binding of service types to service instances during runtime integrates well with the dynamic nature of the healthcare application domain, i.e. provides a high degree of flexibility.

The CASCOM infrastructure that is currently being built will be evaluated in detail in a real-world setting in cooperation with TILAK, the umbrella organization of the state hospitals of the Austrian state Tyrol. In future work, we aim – among other things – to address more sophisticated transaction and failure models, especially by considering malicious failures (i.e. Byzantine failures as they might appear in untrusted environments).

## Acknowledgements

This work is supported by the EU in the 6th Framework Programme within the STREP CASCOM (Context-Aware Business Application Service Coordination in Mobile Computing Environments), contract no. 511632. The work presented in this article has been done while Thorsten Möller and Heiko Schuldt have been affiliated with the Information & Software Engineering Group at the University for Health Sciences, Medical Information and Technology in Tyrol, Austria.

## Notes

- 1 Their location is found either by using the GSM network cell identifier or by GPS.
- 2 This is important when there is more than one service instance available with equal signatures.

## References

- 1 The CASCOM project. <http://www.ist-cascom.org>.
- 2 T.O.-S. Coalition. OWL-S 1.0 (Beta) Draft Release. Autonomous Agents and Multi-Agent Systems, 2003.
- 3 Patil A, Oundhakar S, Sheth A, Verma K. Meteor-s web service annotation framework. In *Proceedings of the World Wide Web Conference, 2004*.
- 4 Baader F, Nutt W. Basic description logics. Chapter in *The Description Logic Handbook: Theory, Implementation and Applications* 47–100. Cambridge: Cambridge University Press, 2003.
- 5 Klusch M, Gerber A, Schmidt M. Semantic web service composition planning with OWLS-Xplan. To appear in the First International Symposium on Agents and the Semantic Web, 2005.
- 6 Hoffmann J, Nebel B. The FF planning system: fast plan generation through heuristic search. *Journal of Artificial Intelligence Research* 2001; **14**: 253–302.
- 7 Schuldts H, Alonso G, Beeri C, Schek H-J. Atomicity and isolation for transactional processes. *ACM Transactions on Database Systems (TODS)* 2002; **27** (1); 63–116.
- 8 Schuldts H, Schek H-J, Alonso G. Transactional coordination agents for composite systems. In *Proceedings of the 3rd International Database Engineering and Applications Symposium (IDEAS 1999), August, Montréal* 321–31. IEEE Computer Society.
- 9 Schuler C, Weber R, Schuldts H, Schek H-J. Scalable peer-to-peer process management: the OSIRIS approach. In *Proceedings of the 2nd ICWS 2004, July, San Diego* 26–34. IEEE Computer Society.
- 10 Schuler C, Schuldts H, Türker C, Weber R, Schek H-J. Peer-to-peer execution of (transactional) processes. To appear in *International Journal of Cooperative Information Systems (IJCIS)* 2005.
- 11 The ARTEMIS project. <http://www.srdc.metu.edu.tr/webpage/projects/artemis>.
- 12 Sycara K, Widoff S, Klusch M, Lu J. *LARKS: Dynamic Matchmaking among Heterogeneous Software Agents in Cyberspace*. Boston: Kluwer, 2002.
- 13 Paolucci M, Kawamura T, Payne T R, Sycara K P. Semantic matching of web services capabilities. In *Proceedings of the First International Semantic Web Conference on the Semantic Web* 333–47. Springer, 2002.
- 14 Colucci S, Noia T D, Sciascio E D, Donini F, Mongiello M, Piscitelli G, Rossi G. An agency for semantic-based automatic discovery of web-services. In *Artificial Intelligence Applications and Innovations: Proceedings of IFIPWCC-04* 315–28. Boston: Kluwer, 2004.
- 15 Li L, Horrocks I. A software framework for matchmaking based on semantic web technology. In *Proceedings of the 12th International Conference on the World Wide Web* 331–9. ACM Press, 2003.
- 16 Constantinescu I, Faltings B. Efficient matchmaking and directory services. In *IEEE/WIC International Conference on Web Intelligence, 2003*.
- 17 Klein M, Bernstein A. Towards high-precision service retrieval. *IEEE Internet Computing* 2004; **8** (1); 30–6.
- 18 Banaei-Kashani F, Chen C, Shahabi C. WSPDS: Web Services Peer-to-Peer Discovery Service. In *ISWS 2004*.
- 19 Bansal S, Vidal J. Matchmaking of web services based on the DAML-S service model. In *AAMAS2003, Melbourne*.
- 20 Trastour D, Bartolini C, Gonzalez-Castillo J. A semantic web approach to service description for matchmaking of services. In *Proceedings of SWWS 2001*.
- 21 Wu D, Parsia B, Sirin J H E, Nau D. Automating DAML-S web services composition using SHOP2. In *Proceedings of 2nd ISWC2003, Sanibel Island, FL* 20–3.
- 22 Sheshagiri M, desJardins M, Finin T. A planner for composing services described in DAML-S. In *Proceedings of AAMAS 2003 Workshop on Web Services and Agent-Based Engineering*.
- 23 Tarkoma S, Laukkanen M. Adaptive agent-based service composition for wireless terminals. In Klusch M et al. eds *Proceedings of CIA VII, August 2003, Helsinki* 16–29. Berlin: Springer, 2003.
- 24 Hamadi R, Benatallah B. A Petri-net-based model for web service composition. In *Proceedings of the 14th Australasian Database Conference: Database Technologies* 191–200. ACM Press, 2003.
- 25 Schek H-J, Schuldts H, Schuler C, Weber R. Infrastructure for information spaces. In *Advances in Databases and Information Systems: Proceedings of the 6th East European Symposium, ADBIS 2002, September, Bratislava* 23–36.
- 26 Foster I, Kesselman C eds. *The Grid: Blueprint for a New Computing Infrastructure* 2nd edn. San Francisco: Morgan Kaufmann, 2004.

- 27** Dogac A, Tambag Y, Tumer A, Ezbiderli M, Tatbul N, Hamali N, Icdem C, Beeri C. A workflow system through cooperating agents for control and document flow over the internet. In *Proceedings of the 7th International Conference on Cooperative Information Systems (CoopIS 2000)*, September 2000, Eilat 138–43.
- 28** Pleisch S, Schiper A. Approaches to fault-tolerant and transactional mobile agent execution: an algorithmic view. *ACM Computing Surveys* 2004; **36** (3); 219–62.

**Correspondence to:** Thorsten Möller

**Thorsten Möller**

*University of Basel  
Department of Computer Science  
Database and Information Systems Group  
Bernoullistrasse 16, 4056 Basel,  
Switzerland  
E-mail: thorsten.moeller@unibas.ch*

**Andreas Gerber**

*German Research Center for Artificial  
Intelligence (DFKI)  
Stuhlsatzenhausweg 3, 66123 Saarbrücken,  
Germany  
E-mail: gerber@dfki.de*

**Matthias Klusch**

*DFKI, Germany  
E-mail: klusch@dfki.de*

**Heiko Schuldt**

*University of Basel  
Switzerland  
E-mail: heiko.schuldt@unibas.ch*

---

## KDEC: Secure Distributed Data Clustering

J. Costa da Silva, M. Klusch, S. Lodi, G. Moro: Privacy-preserving agent-based distributed data clustering. *Web Intelligence and Agent Systems*, 4(2):221 - 238, IOS Press, 2006.

# Privacy-preserving agent-based distributed data clustering

Josenildo Costa da Silva<sup>a</sup>, Matthias Klusch<sup>a</sup>, Stefano Lodi<sup>b,\*</sup> and Gianluca Moro<sup>c</sup>

<sup>a</sup>*Deduction and Multiagent Systems, German Research Center for Artificial Intelligence, Stuhlsatzenhausweg 3, 66123 Saarbrücken, Germany*

<sup>b</sup>*Department of Electronics, Computer Science, and Systems, University of Bologna, Viale Risorgimento 2, 40136 Bologna BO, Italy*

<sup>c</sup>*Department of Electronics, Computer Science and Systems, University of Bologna, Via Venezia 52, 47023 Cesena FC, Italy*

**Abstract.** A growing number of applications in distributed environment involve very large data sets that are inherently distributed among a large number of autonomous sources over a network. The demand to extend data mining technology to such distributed data sets has motivated the development of several approaches to distributed data mining and knowledge discovery, of which only a few make use of agents. We briefly review existing approaches and argue for the potential added value of using agent technology in the domain of knowledge discovery, discussing both issues and benefits. We also propose an approach to distributed data clustering, outline its agent-oriented implementation, and examine potential privacy violating attacks which agents may incur.

**Keywords:** Distributed data mining, data clustering, data security, Multi-Agent System, data privacy

## 1. Introduction

Mining information and knowledge from huge data sources such as weather databases, financial data portals, or emerging disease information systems has been recognized by industrial companies as an important area with an opportunity of major revenues from applications such as business data warehousing, process control, and personalised on-line customer services over the Internet and Web. *Knowledge discovery* (KD) is a process aiming at the extraction of previously unknown and implicit knowledge out of large databases which may potentially be of added value for some given application [7]. The automated extraction of unknown patterns, or *data mining* (DM), is a central element of the KD process. The large variety of DM techniques which have been developed over the past decade includes methods for pattern-based similarity search,

cluster analysis, decision-tree based classification, generalization taking the data cube or attribute-oriented induction approach, and mining of association rules [2]. One classical application of data mining is the market-based or basket analysis of customer transactions, via offline methods for partitioning, discovery of sequential patterns, including techniques to efficiently reduce the set of potential candidates for the selection of relevant items, such as hashing and sampling.

The increasing demand to scale up to massive data sets inherently distributed over a network, with limited bandwidth and available computational resources, motivated the development of methods for parallel (PKD) and distributed knowledge discovery (DKD) [17]. The related pattern extraction problem in DKD is referred to as *distributed data mining* (DDM). DDM is expected to perform partial analysis of data at individual sites and then to send the outcome as a partial result to other sites, where it is sometimes required to be aggregated to the global result. The principal problems any approach to DDM must cope with concern issues of autonomy, privacy, and scalability.

\*Corresponding author. Tel.: +39 051 2093560; Fax: +39 051 2093540; E-mail: stefano.lodi@unibo.it.



Most of the existing DM techniques were originally developed for centralized data and need to be modified for handling the distributed case. As a consequence, one of the most widely used approaches to DDM in business applications is to apply traditional DM techniques to data which have been retrieved from different sources and stored in a central *data warehouse*, i.e., a collection of integrated data from distributed data sources in a single repository [23]. However, despite its commercial success, such a solution may be impractical or even impossible for some business settings in distributed environments. For example, when data can be viewed at the data warehouse from many different perspectives and at different levels of abstraction, it may threaten the goal of protecting individual data and guarding against invasion of privacy. Requirements to respect strict, or a certain degree of, autonomy of given data sources, as well as privacy restrictions on individual data, may make monolithic DM unfeasible.

Another problem arises with the need to scale up to massive data sets which are distributed over a large number of sites. For example, the NASA Earth Observing System (EOS) is a data collector for satellites producing 1450 data sets of about 350GB per day and per pair of satellites at a very high rate which are stored and managed by different systems, geographically located all over the USA. Any online mining of such huge and distributed data sets in a central data warehouse may be prohibitively expensive in terms of costs for both communication and computation.

To date, most work on DDM and PDM use distributed processing and the decomposability of data mining problems to scale up to large data sources. One lesson from the recent research work on DDM is that cooperation among distributed DM processes may allow effective mining even without centralized control [16]. This in turn leads us to the question of whether there is any real added value in using concepts from agent technology [18,35] for the development of advanced DDM systems. A number of DDM solutions are available using various techniques such as distributed association rules, distributed clustering, Bayesian learning, classification (regression), and compression, but only a few of them make use of intelligent agents at all. In general, the inherent feature of software agents, as being autonomous and capable of adaptive and deliberative reasoning, seems to fit quite well with the requirements of coping with the above mentioned problems and challenges of DDM. An autonomous data mining agent, as a special kind of information agent [18], may perform various kinds of mining operations on behalf

of its user(s) or in collaboration with other agents. Systems of cooperative information agents for data mining tasks in distributed massive data environments, such as multidimensional peer-to-peer networks [22,25,28] and grid computing systems [9], appear to be quite a natural vision for the near future.

In this paper we briefly review and classify existing DDM systems and frameworks according to some criteria in Section 2. This is followed by a brief discussion on the benefits of using agents for DDM in Section 3. We introduce in Section 4 an agent-based, distributed data clustering scheme and discuss the threats to data privacy which potentially arise in its application. We conclude the paper in Section 6 with an outline of ongoing and future research work.

## 2. State of the art

In this section we provide a brief review of the most representative agent-based DDM systems to date, according to (a) the kind, type, and used means for security of data processed; (b) used DM techniques, implementation of the system and agents; and (c) the architecture with respect to the main coordination and control, execution of data processing, and transmission of agents, data, and models in due course of the DM tasks to be pursued by the system.

*BODHI* [17] has been designed according to a framework for collective DM tasks on heterogeneous data sites such as supervised inductive distributed function learning and regression. This framework guarantees a correct local and global data model with low network communication load. *BODHI* is implemented in Java; it offers message exchange and runtime environments (agent stations) for the execution of mobile agents at each local site. The mining process is distributed to the local agent stations and agents that are moving between them on demand each carrying its state, data and knowledge. A central facilitator agent is responsible for initializing and coordinating DM tasks to be pursued within the system by the agents and agent stations, as well as the communication and control flow between the agents. Inter-agent communication bases on KQML [8].

*PADMA* [16] deals with the problem of DDM from homogeneous data sites. Partial data cluster models are first computed by stationary agents locally at different sites. All local models are collected to a central site that performs a second-level clustering algorithm to generate the global cluster model. Individual agents perform

hierarchical clustering in text document classification, and web based information visualization.

*JAM* [32] is a Java-based multi-agent system designed to be used for meta-learning DDM. Different learning classifiers such as Ripper, CART, ID3, C4.5, Bayes, and WPEBLS can be executed on heterogeneous (relational) databases by any JAM agent that is either residing on one site or is being imported from other peer sites in the system. Each site agent builds a classification model and different agents build classifiers using different techniques. JAM also provides a set of meta-learning agents for combining multiple models, learnt at different sites, into a meta-classifier that in many cases improves the overall predictive accuracy. Once the combined classifiers are computed, the central JAM system coordinates the execution of these modules to classify data sets of interest at all data sites simultaneously and independently.

*Papyrus* [1] is a Java-based system addressing wide-area DDM over clusters of heterogeneous data sites and meta-clusters. It supports different task and predictive model strategies including C4.5. Mobile DM agents move data, intermediate results, and models between clusters to perform all computation locally and reduce network load, or from local sites to a central root which produces the final result. Each cluster has one distinguished node which acts as its cluster access and control point for the agents. Coordination of the overall clustering task is either done by a central root site or distributed to the (peer-to-peer) network of cluster access points. Papyrus supports various methods for combining and exchanging the locally mined predictive models and metadata required to describe them by using a special markup language.

Common to all approaches, is that they aim at integrating the knowledge discovered from data at different geographically distributed network sites, with a minimum amount of network communication and a maximum of local computation.

### 3. Why agents for DDM?

Looking at the state of the art of agent-based DDM systems presented in the previous section we may identify the following arguments in favor or against the use of intelligent agents for distributed data mining.

**Autonomy of data sources.** A DM agent may be considered as a modular extension of a data management system to deliberately handle the access to the data source in accordance with constraints on the re-

quired autonomy of the system, data and model. This is in full compliance with the paradigm of cooperative information systems [26].

**Interactive DDM.** Pro-actively assisting agents may drastically limit the amount a human user has to supervise and interfere with the running data mining process [39]. For example, DM agents may anticipate the individual limits of the potentially large search space and proper intermediate results particularly driven by their individual users' preferences with respect to the particular type of DM task at hand.

**Dynamic selection of sources and data gathering.** One challenge for intelligent DM agents acting in open distributed data environments in which, for example, the DM tasks to pursue, the availability of data sites and their content may change at any time, is to discover and select relevant sources. In such settings DM agents may be applied to adaptively select data sources according to given criteria such as the expected amount, type and quality of data at the considered source, actual network and DM server load [30]. Such DM agents may be used, for example, to dynamically control and manage the process of data gathering to support any OLAP (online analytical processing) and business data warehouse application.

**Scalability of DM to massive distributed data.** One option to reduce network and DM application server load may be to let DM agents migrate to each of the local data sites in a DDM system on which they may perform mining tasks locally, and then either return with or send relevant pre-selected data to their originating server for further processing. Experiments in using mobile information filtering agents in distributed data environments are encouraging [34].

**Multi-strategy DDM.** For some complex application settings an appropriate combination of multiple data mining techniques may be more beneficial than applying just one particular one. DM agents may learn in the due course of their deliberative actions which one to choose, depending on the type of data retrieved from the different sites and the mining tasks to be pursued. The learning process of the multi-strategy selection of DM methods is similar to the adaptive selection of coordination strategies in a multi-agent system as proposed, for example, in [27].

**Collaborative DM.** DM agents may operate independently on data they have gathered at local sites, and then combine their respective models. Or they may agree to share potential knowledge as it is discovered, in order to benefit from the additional opinions of other DM agents. Meta-learning techniques may be

used to mine homogeneous, distributed data. However, naive approaches to local data analysis may produce an ambiguous and incorrect global data model if different heterogeneous data sites are involved which store data for different sets of features, possibly with some common features among the sites. Collaborative DM agents may negotiate among each other and jointly plan a solution for the above mentioned problems at hand. The need for DM agents to collaborate is prominent, for example, in cases where credit card frauds have to be detected by scanning, analysing, and partially integrating world-wide distributed data records in different, autonomous sources. Other applications of potential added value include the pro-active re-collection of geographically distributed patient records and mining of the corresponding data space on demand to infer implicit knowledge to support an advanced treatment of patients, regardless of which, and how many hospitals they have been taken into in the past. However, frameworks for agent-based collective data mining, such as BODHI, are still more than rare to date.

**Security and trustworthiness.** In fact, this may be an argument against the use of agents for DDM. Of course, any agent-based DDM system has to cope with the problem of ensuring data security and privacy. However, any failure to implement the minimal privilege at a data source, which means endowing subjects with only enough permission to discharge their duties, could give any mining agent unsolicited access to sensitive data. Moreover, any mining operation performed by agents of a DDM system lacking a sound security architecture could be subject to eavesdropping, data tampering, or denial of service attacks. Agent code and data integrity is a crucial issue in secure DDM: Subverting or hijacking a DM agent places a trusted piece of (mobile) software – thus any sensitive data carried or transmitted by the agent – under the control of an intruder. In cases where DM agents are even allowed to migrate to remote computing environments of the distributed data sites of the DDM system methods to ensure confidentiality and integrity of a mobile agent have to be applied. Regarding agent availability there is certainly no way to prevent malicious hosts from simply blocking or destroying the temporarily residing DM agents but selective replication in a fault tolerant DDM agent architecture may help. In addition, data integration or aggregation in a DDM process introduces concern regarding inference attacks as a potential security threat. Data mining agents may infer sensitive information even from partial integration to a certain extent and with some probability. This problem,

known as the so called inference problem, occurs especially in settings where agents may access data sources across trust boundaries which enable them to integrate implicit knowledge from different sources using commonly held rules of thumb. None of the existing DDM systems, agent-based or not, is capable of coping with this inference problem in the domain of secure DDM.

In the following sections we investigate how agents may be used to perform a special kind of distributed data mining, that is clustering of data at different homogeneous data sites. For this purpose, we present an approach to distributed cluster analysis based on density estimation, and then briefly discuss issues of implementing the resulting scheme for distributed data clustering in an agent-based DDM system, including data privacy and trustworthiness.

#### 4. A scheme for distributed data clustering

##### 4.1. Density estimation based clustering

*Cluster analysis* is a descriptive data mining task which aims at partitioning a data set into groups such that the data objects in one group are similar to each other and are as different as possible from those in other groups. As dense regions of the data space are more likely to be populated by similar data objects, one popular clustering technique is based on reducing the search for clusters to the search for such regions. In *density estimation* (DE) based clustering the search for densely populated regions is accomplished by estimating a probability density function from which the given data set is assumed to have arisen [5,15,31]. One important family of methods requires the computation of a non-parametric density estimate known as *kernel estimator*.

Let us assume a set  $D = \{x_i | i = 1, \dots, N\} \subseteq \mathbb{R}^d$  of data objects. Kernel estimators originate from the intuition that the higher the number of neighbouring data objects  $x_i$  of some given space object  $x \in \mathbb{R}^d$ , the higher the density at this object  $x$ . However, there can be many ways of computing the influence of neighbouring data objects. Kernel estimators use a so called *kernel function*  $K(x) : \mathbb{R}^d \rightarrow \mathbb{R}$  which integrates to unity over  $\mathbb{R}^d$ . A *kernel estimator*  $\hat{\varphi}_{K,h}[D](\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}_+$  is defined as the sum over all data objects  $x_i$  in  $D$  of the differences between  $x$  and  $x_i$ , scaled by a factor  $h$ , called *window width*, and weighted by the kernel function  $K$ :

$$\hat{\varphi}_{K,h}[D](\mathbf{x}) = \frac{1}{Nh^d} \sum_{i=1}^N K\left(\frac{1}{h}(\mathbf{x} - \mathbf{x}_i)\right). \quad (1)$$

A kernel estimator can be considered as a sum of “bumps” placed at the observations. The window width  $h$  controls the smoothness of the estimate, whereas  $K$  determines the shape of the bumps.

Usually,  $K$  is radially symmetric and non-negative with a unique maximum in the origin; in this case,  $K$  formalizes the decay of the influence of a data object according to its distance. In the following, we will consider only kernels of this kind.

Prominent examples of kernel functions are the standard multivariate normal density  $(2\pi)^{-d/2} \exp(-\frac{1}{2}\mathbf{x}^T \mathbf{x})$ , the multivariate uniform kernel  $w(\mathbf{x})$ , defined by

$$w(\mathbf{x}) = \begin{cases} c_d^{-1} & \text{if } \mathbf{x}^T \mathbf{x} < 1, \\ 0 & \text{otherwise,} \end{cases}$$

and the multivariate Epanechnikov kernel  $K_e(\mathbf{x})$ , defined by

$$K_e(\mathbf{x}) = \begin{cases} \frac{1}{2}c_d^{-1}(d+2) & \text{if } \mathbf{x}^T \mathbf{x} < 1, \\ 0 & \text{otherwise,} \end{cases}$$

where  $c_d$  is the volume of the unit  $d$ -dimensional sphere. An example of kernel estimate in  $\mathbb{R}$  showing the normal component kernels is shown in Fig. 1.

When  $d \geq 2$ , it may be advisable to linearly transform the data, in order to avoid large differences of spread in the dimensions, and transform inversely after applying the estimate [11]. In an equivalent manner, this is accomplished by the estimate:

$$\hat{\varphi}_{K,h}[D](\mathbf{x}) = \frac{(\det \mathbf{S})^{-1/2}}{Nh^d} \sum_{i=1}^N k\left(\frac{1}{h^2}(\mathbf{x} - \mathbf{x}_i)^T \mathbf{S}^{-1}(\mathbf{x} - \mathbf{x}_i)\right) \quad (2)$$

where  $\mathbf{S}$  is the sample covariance matrix of  $D$ , and  $k$  satisfies  $k(\mathbf{x}^T \mathbf{x}) = K(\mathbf{x})$ .

The criteria to optimally choose  $K$  and  $h$  have been extensively dealt with in the literature on non-parametric density estimates, where  $K$  and  $h$  are deemed optimal if they minimize the expected value of the integrated squared pointwise difference between the estimate and the true density  $\varphi$  of the data, or *mean integrated square error* (MISE). It has been shown that the performance of the Epanechnikov kernel, measured by the MISE criterion, is optimal; however, the performances of commonly used kernels do not differ substantially from the performance of an optimal kernel.

Therefore, the choice of a kernel may be based on computational or differentiability properties, rather than on its MISE. The optimal value of the window width  $h$  can be approximated as

$$h_{\text{opt}} = A(K)N^{-1/(d+4)} \quad (3)$$

where  $A(K)$  depends also on the unknown true density  $\varphi$ . The value of  $A(K)$  has been tabulated for commonly used kernels when  $\varphi$  is a unit multivariate normal density [31]. The resulting values of  $h_{\text{opt}}$  can be used directly in Eq. (2). Alternatively, if Eq. (1) is used, the value of  $h$  can be defined by

$$h = \left(d^{-1} \sum_i s_{ii}\right)^{1/2} h_{\text{opt}} \quad (4)$$

taking thus into account the average variance of the data over all dimensions [31].

In DE-clustering, the kernel estimate of a data set has been used for the discovery of many types of density-based clusters [5,15,31]. One simple type is the so-called *center-defined* cluster: every local maximum of  $\hat{\varphi}$  corresponds to a cluster including all data objects which can be connected to the maximum by a continuous, uphill path in the graph of  $\hat{\varphi}$ . It is apparent that, for every data object, an uphill climbing procedure driven by the kernel estimate will find the local maximum representing the object's cluster [15,19,31].

#### 4.2. KDE-based distributed data clustering

We define the problem of *homogeneous distributed data clustering* (homogeneous DDC) as follows. Let  $D = \{\mathbf{x}_i | i = 1, \dots, N\} \subseteq \mathbb{R}^d$  be a data set of objects. Let  $L_j, j = 1, \dots, M$  be a finite set of *sites*. Each site  $L_j$  stores one data set  $D_j$  of size  $N_j$ . It will be assumed that  $D = \bigcup_{j=1}^M D_j$ . Let  $\mathcal{C} = \{C_k\} \subseteq 2^D$  be a clustering of  $D$ , whose elements are pairwise disjoint. The homogeneous DDC problem is to find for  $j = 1, \dots, M$ , a site clustering  $\mathcal{C}$  residing in the data space of  $L_j$ , such that  $\mathcal{C}_j = \{C_k \cap D_j | k = 1, \dots, |\mathcal{C}|\}$  (*correctness requirement*), time and communications costs are minimized (*efficiency requirement*), and, at the end of the computation, the size of the subset of  $D$  which has been transferred out of the data space of any site  $L_j$  is minimized (*privacy requirement*). The traditional solution to the homogeneous DDC problem is to simply collect all the distributed data sets  $D_j$  into one centralized repository where the clustering of their union is computed and transmitted to the sites. Such an approach, however, does not satisfy our problem's requirements both in terms of privacy

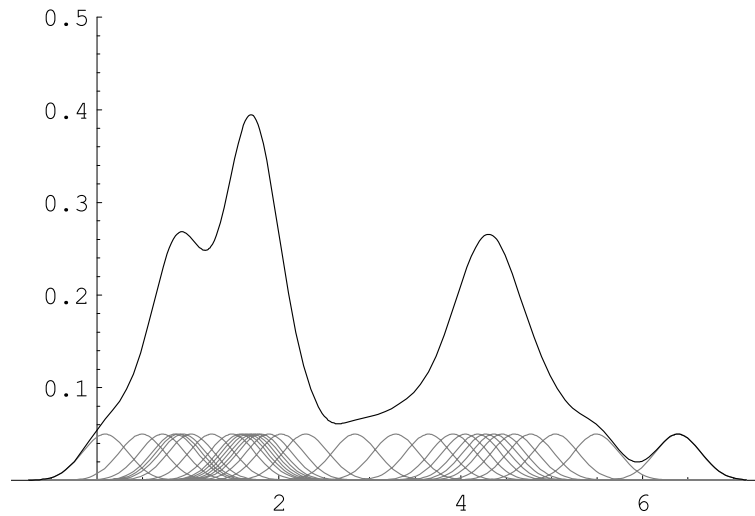


Fig. 1. Kernel estimate and its normal component kernels ( $h = 0.25$ ,  $N = 32$ ).

and efficiency. Therefore, in [19] a different approach has been proposed yielding a kernel density estimation based clustering scheme, called KDEC, which may be implemented by appropriately designed DM agents of an agent-based DDM system. Before examining the issues and benefits of an agent-based implementation, we briefly review the KDEC scheme.

The KDEC scheme is based on three simple observations: density estimates are (i) additive for homogeneous distributed data sets, (ii) sufficient for computing DE-clustering, and (iii) provide a more compact representation of the data set for the purpose of transmission. In the sequel, we tacitly assume that all sites  $L_j$  agree on using a global kernel function  $K$  and a global window width  $h$ . We will therefore omit  $K$  and  $h$  from our notation, and write  $\hat{\varphi}[D](\mathbf{x})$  for  $\hat{\varphi}_{K,h}[D](\mathbf{x})$ .

The global density estimate  $\hat{\varphi}[D](\mathbf{x})$  can be written as the sum of the site density estimates, one estimate for every data set  $D_j$ :

$$\begin{aligned} \hat{\varphi}[D](\mathbf{x}) &= \sum_{j=1}^M \sum_{\mathbf{x} \in D_j} K\left(\frac{1}{h}(\mathbf{x} - \mathbf{x}_i)\right) \\ &= \sum_{j=1}^M \hat{\varphi}[D_j](\mathbf{x}). \end{aligned} \quad (5)$$

Thus, the local density estimates can be transmitted to and summed up at a distinguished helper site yielding the global estimate which can be returned to all sites. Each site  $L_j$  then may apply to its local data space a hill-climbing technique to assign clusters to its local data objects. Note however that Eq. (1) explic-

itly refers to the data objects  $\mathbf{x}_i$ . Hence, transmitting a naive coding of the estimate entails transmitting the data objects which contradicts the privacy requirement. Multi-dimensional sampling provides an alternative extensional representation of the estimate which makes no explicit reference to the data objects.

For  $\mathbf{x} \in \mathbb{R}^d$ , let  $x_1, \dots, x_d$  be its components. Let  $\boldsymbol{\tau} = [\tau_1, \dots, \tau_d]^T \in \mathbb{R}^d$  be a vector of *sampling periods*, and let  $\mathbf{z} \bullet \boldsymbol{\tau}$  denote  $[z_1\tau_1, \dots, z_d\tau_d]^T$ , where  $\mathbf{z} \in \mathbb{Z}^d$ . A function  $f: \mathbb{R}^d \rightarrow \mathbb{R}$  is *band-limited* to a bounded  $B \subset \mathbb{R}^d$  if and only if the support of its Fourier transform is contained in  $B$ . If  $B$  is a subset of a rectangle  $[-\pi/\tau_1, \pi/\tau_1] \times \dots \times [-\pi/\tau_d, \pi/\tau_d]$ , it is well-known that the *sampling series*

$$\begin{aligned} &\sum_{\mathbf{z} \in \mathbb{Z}^d} f(\mathbf{z} \bullet \boldsymbol{\tau}) \operatorname{sinc}\left(\frac{x_1}{\tau_1} - z_1\right) \\ &\dots \operatorname{sinc}\left(\frac{x_d}{\tau_d} - z_d\right), \end{aligned} \quad (6)$$

where

$$\operatorname{sinc}(x) = \begin{cases} 1 & \text{if } x = 0, \\ \frac{\sin \pi x}{\pi x} & \text{otherwise,} \end{cases}$$

converges to  $f$  under mild conditions on  $f$  (see, e.g. [14, p.155]). If we let  $f(\cdot) = \hat{\varphi}[D_j](\cdot)$  in Eq. (6), and truncate the series to a finite  $d$ -dimensional rectangle  $R(\mathbf{z}_1, \mathbf{z}_2)$  having diagonal  $(\mathbf{z}_1, \mathbf{z}_2)$  we obtain an interpolation formula:

$$\sum_{\mathbf{z} \in R(\mathbf{z}_1, \mathbf{z}_2)} \sum_{j=1}^M \hat{\varphi}[D_j](\mathbf{z} \bullet \boldsymbol{\tau}) \operatorname{sinc}\left(\frac{x_1}{\tau_1} - z_1\right)$$

$$\dots \text{sinc} \left( \frac{x_d}{\tau_d} - z_d \right) \quad (7)$$

Notice that the function represented by Eq. (7) is not extensionally equal to the kernel global estimate  $\hat{\varphi}[D](\mathbf{x})$  both because kernel estimates are not band-limited on any region, and because of the truncation in the series. However, as argued in [19], the approximation introduces only a small error. In fact, both a density estimate and its Fourier transform vanish rapidly when the norm of the argument  $\rightarrow \infty$ ; therefore, we may take  $\tau$  in such a way that the transform is negligible in  $\mathbb{R}^d \setminus [-\pi/\tau_1, \pi/\tau_1] \times \dots \times [-\pi/\tau_d, \pi/\tau_d]$ , and by selecting  $(z_1, z_2)$  so that the estimate is negligible in  $\mathbb{R}^d \setminus R(z_1, z_2)$ .

Therefore, Eq. (7) gives an approximation of the global density estimate that can be exploited to devise a distributed clustering scheme: For  $j = 1, \dots, M$ , the samples  $\{\hat{\varphi}[D_j](z \bullet \tau) : z \in R(z_1, z_2)\}$  of the  $j$ -th local density estimate are transmitted to and summed up at a distinguished helper site yielding the samples of the global estimate which can be returned to all sites, which then use Eq. (7) as the global density estimate to which the hill-climbing technique is applied.

**Algorithm 1** KDEC: distributed clustering based on density estimation

```

func Interpolate ( $x, \tau, z_1, z_2, Sam[]$ )  $\equiv$ 
  foreach  $z \in R(z_1, z_2)$  do
     $r := r + Sam[z] \Pi_{i=1}^d Sinc \left( \frac{x_i}{\tau_i} - z_i \right)$ 
  od;
   $r$ .
proc DataOwner ( $D[], H, Clus[]$ )  $\equiv$ 
  Negotiate ( $H, \tau, z_1, z_2, K, h$ );
  Send ( $Sample (D, \tau, z_1, z_2, K, h)$ );
   $Sam := Receive (H)$ 
  for  $i := 1$  to Length ( $D$ ) do
     $Clus [i] := Nearest($ 
      FindLocalMax ( $x_i, \tau, z_1, z_2, Sam,$ 
         $\nabla Interpolate( )$ );
  od.
proc Helper ( $L[]$ )  $\equiv$ 
  Negotiate ( $L$ ); SetZero( $Sam$ );
  for  $j := 1$  to Length ( $L$ ) do
     $Sam := Sam + Receive (L[j])$ 
  od;
  for  $j := 1$  to Length ( $L$ ) do
    Send( $Sam, L[j]$ )
  od.

```

A distributed implementation of the KDEC scheme is sketched as Algorithm 1. Local sites run *DataOwner*, whereas the helper site runs *Helper*, where  $H$ ,  $D[]$ ,  $L[]$ , reference the helper, the local data set, a list of local sites, respectively, and  $Clus[]$  is the result (an object-cluster look-up table). *Negotiate* sets up a fo-

rum where the local sites can reach an agreement on  $\tau$ ,  $R(z_1, z_2)$ , the kernel  $K$  and the window width  $h$ . Local sites send the samples of the local estimate of  $D[]$  to  $H$  which sums them orderly. Finally, each local site receives the global samples and uses them in procedure *Interpolate* to compute the values of the global density estimate and applies the gradient-driven, hill-climbing procedure *FindLocalMax* to compute the corresponding local data clusters (see [19] for more details).

*Example.* Figure 2 shows a dataset of 100 2-dimensional objects containing two clusters, which were randomly generated with a bivariate normal distribution, and the contour plot of its kernel estimate, computed for a normal kernel with  $h = h_{opt} = 1/\sqrt{5}$  by means of Eq. (2). Figure 3 illustrates the clusters computed by the KDEC scheme on the dataset, divided between two sites in such a way that each local dataset spans across both clusters. Objects plotted with equal symbol shapes belong to the same cluster and viceversa. The figure also shows the contours of the sampling series of the whole dataset, computed with  $h = 1/\sqrt{5}$ ,  $\tau = [h, h]^T$ . As the local maxima of the estimate are preserved, the clusters are correctly recovered. Figure 4 shows the output of KDEC and the contours of the series on the same datasets, setting  $h = 1/\sqrt{5}$ ,  $\tau = [5h, 5h]^T$ . In this case, the sampling series is clearly a poor approximation of the estimate. Nevertheless, the local maxima are preserved, and KDEC returns the correct clusters.

Algorithm 1 is abstract, in that it only specifies the flow of information between processes, and it ignores actual running locations in the network and the technology of distributed computation. In the following section we will describe various options to make the KDEC scheme more concrete, and some pitfalls to data privacy and security.

#### 4.3. Agents for KDEC-based homogeneous DDC

The KDEC scheme may be modified in several ways to generate protocols for stationary or mobile agents. We assume a DDM system consisting of a set  $\{L_n | 1 \leq n \leq M, M > 1\}$  of networked homogeneous data sites and a set  $\{L_n | M + 1 \leq n \leq M'\}$  of helpers. Both data sites and helpers provide support for local and external, visiting agents. Each site respects its local autonomy by individually granting read-only access to external data mining agents.

$M'$  data mining agents  $A_n$  have been designed and deployed at the data sites and helpers. Agent  $A_n$  is assumed to be associated to site  $L_n$ , for  $1 \leq n \leq M'$

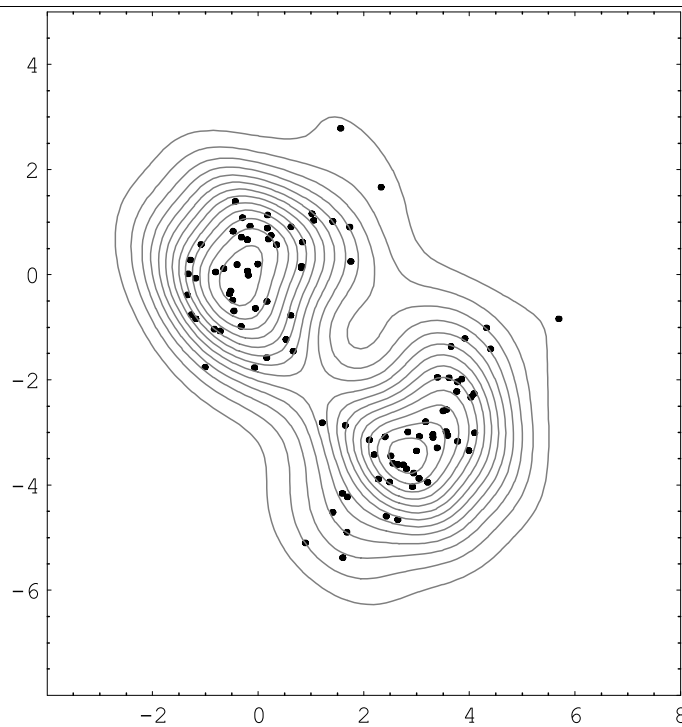


Fig. 2. Scatter plot of a 2-dimensional dataset and contour plot of its kernel estimate,  $h = 1/\sqrt{5}$ .

The data site agent interested in starting a data mining initiative, named the *initiator*, collects a list of possible participants by using search techniques commonly adopted in peer-to-peer systems, whose complexity is logarithmic in the number of peers in the network. The initiator agent searches for all peers able to interact according to a given data clustering service, which corresponds in this case to our algorithm, executed with a protocol specified by the initiator itself. When the list has been fixed, the initiator selects a *master* helper and, possibly, a set of auxiliary helpers, depending on the protocol. The initiator sends the master helper the list of peer data sites, the list of auxiliary helpers, and the protocol specifications. The helper takes care of arranging the data sites and the auxiliary helpers according to the topology specified in the protocol, and starts the mining activity. For  $n \leq M'$ ,  $A_n$  carries out the initial negotiations of the protocol on behalf of  $L_n$  as stationary agent. Later,  $A_n$ ,  $n > M$ , may proceed as either stationary or mobile agent, depending on the protocol.

To illustrate the negotiation and the protocols we introduce the following scenario.

*Example.* It has been announced that an international insurance company will soon enter the national

market. As a consequence of the announcement, eight national insurers decide to combine and analyse their policy data by clustering. Their goal is to achieve a better understanding of the national market, and to plan strategies for keeping their share. Therefore, each insurer creates a view with the following dimensions: *Latitude*, *Longitude*, *Indemnity*, where *Latitude* and *Longitude* are computed from the address of the policyholder.

For simplicity we assume the datasets of the eight insurers follow three distribution patterns. Distribution 1 (sites 1–3): The policies have high indemnity and are held by inhabitants of large cities in the north-east; distribution 2 (site 4–6): The policies have very high indemnity, their holders live in the south; distribution 3 (sites 7 and 8): The policies have small indemnity and the holders live in the west. Figures 5(a), 5(b), and 5(c) show the scatter plots of the dataset, in the hyperplanes (*Longitude*, *Latitude*), (*Longitude*, *Indemnity*), and (*Latitude*, *Indemnity*), respectively. Objects belonging to the first, second, and third distribution have been plotted in light gray, dark gray and black, respectively.

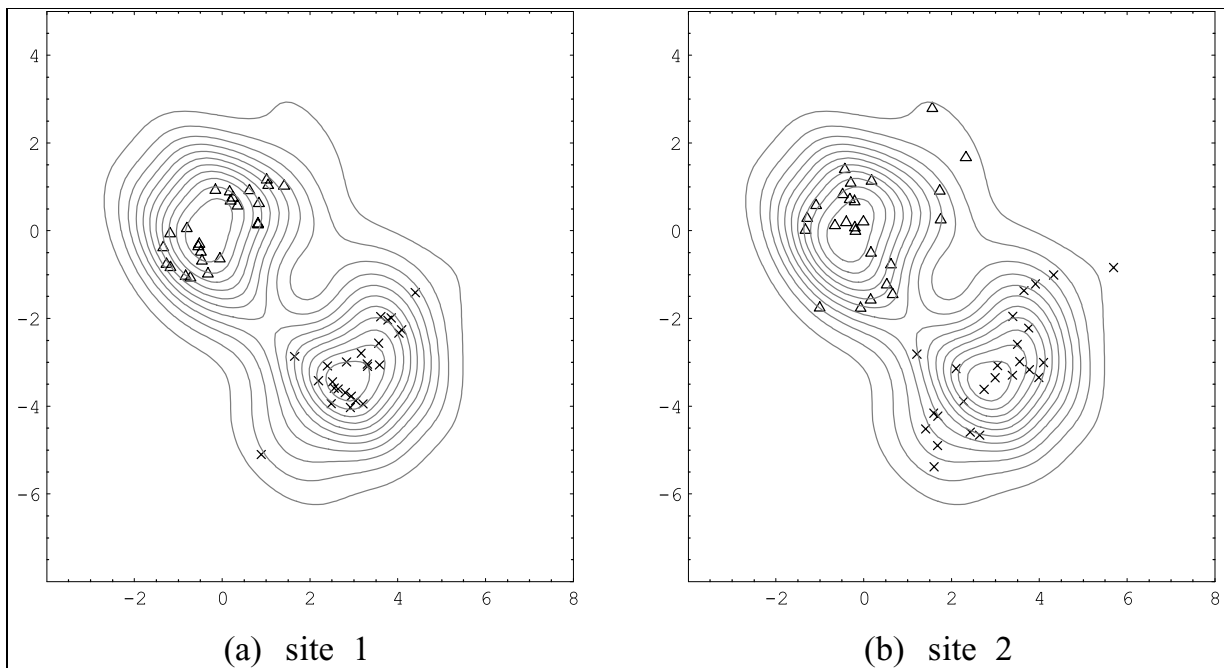


Fig. 3. Clusters computed by KDEEC for  $h = 1/\sqrt{5}$ ,  $\tau = [h, h]^T$ .

#### 4.3.1. Negotiation of parameters

According to the KDEEC scheme, the master helper engages the other site agents in a negotiation to agree what kernel function to use for computing the local density estimate samples, an optimal value of the parameters  $h$  and  $\tau$ , and the data space hyper-rectangle delimiting the data clustering area of interest.

The negotiation procedure is made up of three phases which can be implemented according to the communicative acts of ACL FIPA [10]. The interactions are based on the primitives of such a standard (e.g. *call-for-proposal*, *accept*, *inform*, ...) which give a first level of semantics to messages exchanged among agents.

The helper agent in the first phase asks each participant, using the *query-ref* ACL primitive, for the number of its local objects, the linear sums and sums of squares of all its local objects along each dimension (i.e. object attribute), and finally the corners of the smallest hyper-rectangle covering its data space domain.

In the second phase, when all participant agents have replied with the requested data (using the primitive *inform*), the helper sends them a *call-for-proposal* containing (1) a set of possible kernel functions from which each participant should select its preferred one, (2) the corners  $\mathbf{l}, \mathbf{r} \in \mathbb{R}^d$  of a hyper-rectangle containing the union of all hyper-rectangular data spaces, (3) for every kernel function  $K$ , a recommended value  $h_r$  for  $h$ , and

the parameter  $\beta_K$ , which will be explained below, and (4) a possible value for  $h^{-1} \tau$ .

The helper computes the recommended values of the window width  $h$  for each kernel function using Eq. (4), the marginal variances of the whole dataset, and the values of  $h_{\text{opt}}$ . The marginal variances of the data can be easily computed by the helper from the total number of objects in the dataset, their vector sum, the sums of squares of dimensions, which in turn can be computed as the sum of their local values at each site. The values of  $h_{\text{opt}}$  for each kernel function are given by Eq. (3). Although more sophisticated and accurate ways to choose  $h_{\text{opt}}$  automatically have been studied in the literature, the value given by Eq. (3) will be sufficiently accurate as a starting point for the negotiation. Moreover, it can be computed by the helper from the total number of objects only. The helper sets  $\mathbf{l}, \mathbf{r}$  to the corners  $\mathbf{l}', \mathbf{r}'$  of the smallest hyper-rectangle containing the union of all hyper-rectangular data spaces, randomly extended in order to prevent the disclosure of the coordinates of extremal data objects. That is,  $\mathbf{l} = \mathbf{l}' + \mathbf{v}_l$ ,  $\mathbf{r} = \mathbf{r}' + \mathbf{v}_r$ , where the components of  $\mathbf{v}_l, \mathbf{v}_r$  are uniformly distributed in  $[-h_r, 0]$  and  $[0, h_r]$ , respectively. The parameter  $\beta_K$  is used to delimit how far beyond the border of the data space hyper-rectangle the estimate is not negligible and must be sampled. We assume  $\beta_K$  to be equal to the smallest multiple of  $10^{-2}$  greater than  $\sup\{\|\mathbf{x}\| : K(\mathbf{x}) > 10^{-2}\}$ .



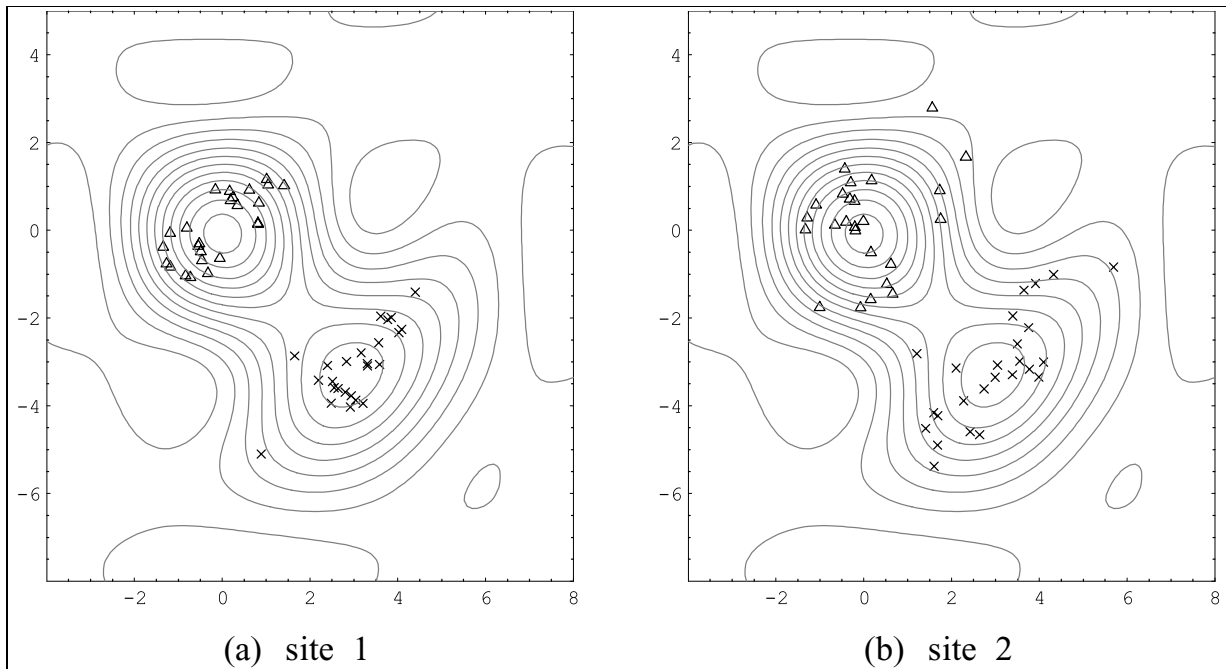


Fig. 4. Clusters computed by KDE for  $h = 1/\sqrt{5}$ ,  $\tau = [5h, 5h]^T$ .

With the exception of the normal kernel, popular kernels have bounded support and their value is zero for  $\|\mathbf{x}\| \geq 1$ ; thus  $\beta_K = 1$ . For the normal kernel, we have  $\beta_K = 3.04$ .

Each proposal returned to the master helper shall include a kernel function, an interval for  $h$ , and an interval for  $h^{-1}\tau$ . The choice of both the kernel and  $h$  have an impact on data privacy. The regularity of the kernel increases privacy (see Section 4.4.1). Isolated maxima centered at the data objects tend to become more and more evident in the density estimate as  $h$  decreases. On the other hand, choosing too large a value for  $h$  may result in an oversmoothed estimate and merging of some natural cluster. Each participant proposes an interval for  $h^{-1}\tau$  such that the resulting time and transmission complexities are compatible with its application constraints. To this purpose, each agent assumes that the estimate is negligible outside the hyper-rectangle of corners  $\mathbf{l} + [-\beta h_r, \dots, -\beta h_r]^T$ ,  $\mathbf{r} + [\beta h_r, \dots, \beta h_r]^T$ . The parameter  $\beta$  is equal to the largest  $\beta_K$  over all kernels  $K$  contained in the *call-for-proposal*.

In the third phase the helper site, after collecting all proposals from the interested participants, defines the final proposal including the most selected kernel function by participants, the best  $h$  and  $\tau$  with respect to all possible counterproposal values and the definitive hyper-rectangular data space. The helper sends the

final proposal and only agents who accept it will be involved in the distributed data clustering initiative.

All communications between the helper and participants are carried out using the digital sign mechanism, moreover each agent involved knows the digital identity of the helper, therefore no one may alter a message or impersonate the helper.

We illustrate the negotiation of parameters with our example.

*Example.* After sending a *query-ref*, the helper is informed of the aggregate statistics of the sites, computes the total count of data objects, the marginal variances of the data, and the corners of the hyper-rectangle covering the data space  $\mathbf{l}' = [-29.11, -16.07, 1.89]^T$ ,  $\mathbf{r}' = [105.49, 97.14, 161.99]^T$ . The eligible kernels are the normal kernel and the Epanechnikov kernel, corresponding to  $h_{\text{opt}} = 0.36$ ,  $h_{\text{opt}} = 1.09$ . The average marginal variance is  $\sigma = 29.83$ . As all sites are concerned about the potential disclosure of their data, their agents choose the most regular kernel, i.e., the normal kernel corresponding to a recommended value  $h_r = \sigma h_{\text{opt}} = 10.80$ . Sites  $L_1, L_2, L_3$  propose the interval  $[h_r/2, h_r]$ , whereas the remaining sites  $L_4, \dots, L_8$  propose  $[h_r, 2h_r]$ .  $L_1, L_2, L_3$  propose a smaller lower bound because the maximum distance between data objects in their datasets is smaller. Thus, for  $h = h_r/2$ , the density estimate has no isolated “bump” which would reveal the location of an

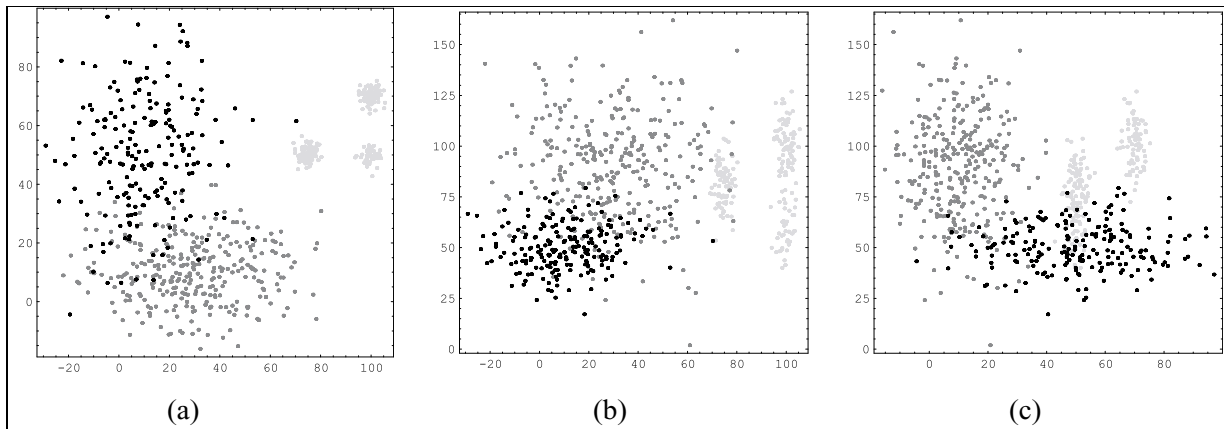


Fig. 5. Scatter plots of example data in the hyperplanes  $(Longitude, Latitude)$ ,  $(Longitude, Indemnity)$ , and  $(Latitude, Indemnity)$ .

object, whereas the structure of the three clusters corresponding to their combined local datasets is visible. In contrast, for  $h < h_r$ , in the region where *Indemnity* is high, some maxima corresponding to objects stored at sites  $L_4$  or  $L_5$  are evident. The insurers owning  $L_4$  and  $L_5$  would be deeply concerned about the possibility that geographical information and high indemnity could be exploited to narrow the search for the identities of the policyholders. Figure 6(a) shows the marginal estimate in the hyperplane  $(Longitude, Indemnity)$  for  $h = h_r/2$ , and Fig. 6(b) is an inset showing the local maxima. Assuming  $\beta = 3.04$  (only the normal kernel and the Epanechnikov kernel are contained in the *call-for-proposal*), the number of required samples is about 5000 for  $\tau = h_r/2$ , which is modest and acceptable for all agents. Thus, all participants propose  $h^{-1}\tau = 1/2$ . The final proposal of the helper contains the normal kernel,  $h = h_r$  and the hyper-rectangle  $l + 3.04[-h_r, -h_r, -h_r]^T, r + 3.04[h_r, h_r, h_r]^T$ .

#### 4.3.2. Protocols

We classify potential implementations and variations of the remaining steps of Algorithm 1 according to two directions: The path of samples, or partial sums of samples, in the network of sites, and the protocol to compute the sums. We consider the following three basic partial orders of the network of sites.

**Sequence** The data sites are arranged into a sequence  $\{L_{\pi(n)}\}_{1 \leq n \leq M}$ . For all  $z \in R(z_1, z_2)$ , the value  $s_{n-1}(z) = \sum_{k=1}^{n-1} \hat{\varphi}[D_{\pi(k)}](z \bullet \tau)$  is known at site  $L_{\pi(n)}$ , where  $s_n(z) = s_{n-1}(z) + \hat{\varphi}[D_{\pi(n)}](z \bullet \tau)$  is computed and moved to  $L_{\pi(n+1)}$ .

**Star** For all  $z \in R(z_1, z_2)$ , at site  $L_n$ , the value  $\hat{\varphi}[D_n](z \bullet \tau)$  is computed and moved to a single helper.

**Tree** The master helper, the auxiliary helpers and the data sites are organized into a tree of order  $b$ , satisfying the following: the master helper is the root of the tree, the  $M$  data sites are exactly the leaves of the tree, the depths of any two leaves differ at most by one, and all internal nodes have exactly  $b$  children, except possibly one internal node of maximal depth, which has at least 2 and at most  $b - 1$  children. A tree satisfying the properties above can be computed straightforwardly in linear time in  $M$ : Create a perfect tree of depth  $\lceil \log_b M \rceil$ , consisting of the master helper as root,  $\lceil (M - 1)/(b - 1) \rceil - 1$  auxiliary helpers, and  $\lfloor (b^{\lceil \log_b M \rceil + 1} - M)/(b - 1) \rfloor$  data sites at depth  $\lceil \log_b M \rceil$ ; then, add the remaining data sites (if any) as children of the auxiliary helpers at depth  $\lceil \log_b M \rceil$ , leaving at most one helper with less than  $b$  children. The minimum value of  $b$  is chosen by the initiator, which therefore knows in advance the maximum number of auxiliary helpers needed. At data site  $L_n$ ,  $1 \leq n \leq M$ , for all  $z \in R(z_1, z_2)$ , the value  $\hat{\varphi}[D_n](z \bullet \tau)$  is computed and moved to the parent. At any helper site (except the tree root)  $L_n$ ,  $n > M$ , the value

$$s_n(z) = \sum_{\substack{1 \leq k \leq M \\ L_k \in \text{subtree}(L_n)}} \hat{\varphi}[D_k](z \bullet \tau)$$

is computed.

In any of the partial orders above, the actual protocol among the sites can be implemented by stationary or mobile agents. In the following, assume  $A_{M+1}$  is the master helper's agent.

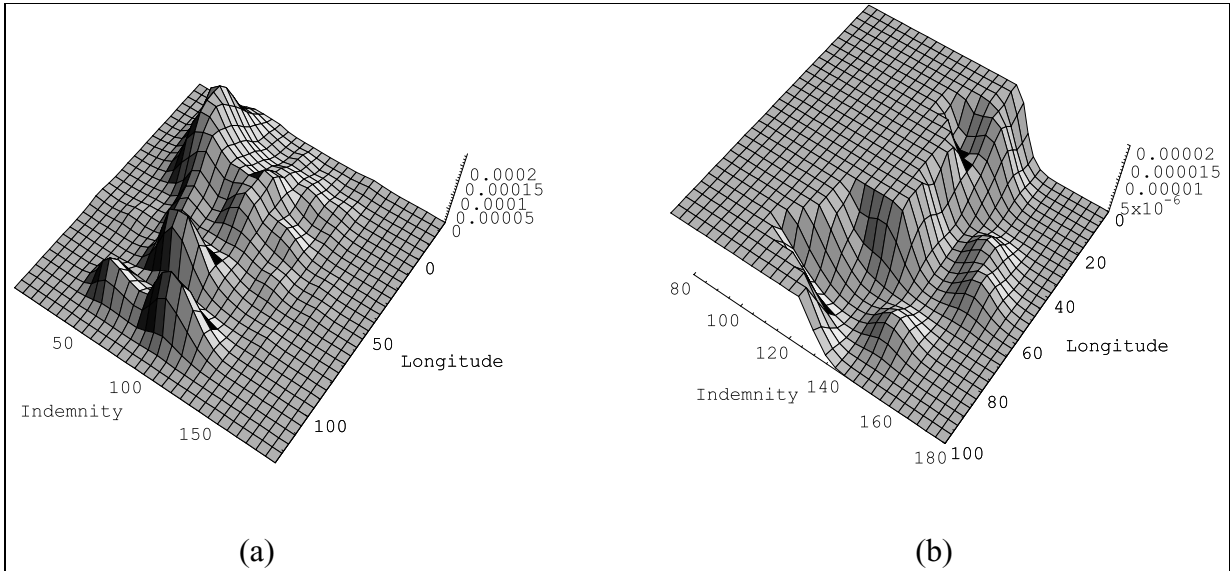


Fig. 6. Marginal kernel estimate ( $h = h_r/2$ ) in the hyperplane *Longitude*, *Indemnity* and inset showing isolated local maxima.

**Sequence**  $A_{M+1}$  selects an arrangement  $\pi$  such that  $\pi(1), \dots, \pi(M)$  is a random permutation of  $1, \dots, M$ , and  $\pi(M+1) = M+1$ .

**Stationary agents** For  $2 \leq n \leq M$ ,  $A_{M+1}$  sends  $A_{\pi(n)}$  the addresses of sites  $L_{\pi(n-1)}$ ,  $L_{\pi(n+1)}$ . To  $A_{\pi(1)}$ , the master sends its own address, the address of  $L_{\pi(2)}$ , and an empty list of samples.  $A_{\pi(1)}$  waits for a list of samples from the master, adds its local list of samples to it and sends the result to  $A_{\pi(2)}$ . Each  $A_{\pi(n)}$  waits for a list of samples from  $A_{\pi(n-1)}$ ,  $2 \leq n \leq M$ , adds its local list of samples to it and sends the result to  $A_{\pi(n+1)}$ . The master waits for the list of samples from  $A_{\pi(M)}$ , and sends it to  $A_n$ , for  $1 \leq n \leq M$ .

**Mobile agents** Agent  $A_{M+1}$  initializes an empty list of samples and moves to  $L_{\pi(1)}$ . When residing at site  $L_{\pi(n)}$  ( $1 \leq n \leq M$ ),  $A_{M+1}$  requests the local agent  $A_{\pi(n)}$  to locally send the list of samples, adds it to the sum in its data space, and moves to  $L_{\pi(n+1)}$ . Finally  $A_{M+1}$  sends the sum to all agents  $A_n$ ,  $1 \leq n \leq M$ .

**Star** Assume agent  $A_{M+1}$  was designated as master.

**Stationary agents**  $A_{M+1}$  waits for the list of local samples from each of the  $A_n$ ,  $1 \leq n \leq M$ , adds the lists and sends the result to  $A_n$ , for  $1 \leq n \leq M$ .

**Mobile agents** For  $n = 1, \dots, M$  agent  $A_{M+1}$  moves to site  $L_n$ , requests the local agent

$A_n$  to locally send the list of samples, adds it to the sum in its data space, and moves back to  $L_{M+1}$ .

**Tree** Assume site agents  $A_{M+2}, \dots, A_{M+H}$  were designated as auxiliary helpers.  $A_{M+1}$  receives the list  $L_{M+2}, \dots, L_{M+H}$  of helper sites and arranges it into a tree, as described earlier in this section. To minimize collusions,  $A_{M+1}$  sends each helper agent  $A_{M+k}$  only the references to its parent and its children. If the agents are mobile,  $A_{M+1}$  also sends each helper agent  $A_{M+k}$  the references to its siblings.

**Stationary agents** Every agent  $A_{M+k}$ ,  $1 \leq k \leq H$ , waits for the list of local samples from all  $A_j$  residing at the children of  $L_{M+k}$ , and adds the lists. Then  $A_{M+k}$  sends the result to its parent, if  $k > 1$ ; the master  $A_{M+1}$  sends the list to  $A_n$ , for  $1 \leq n \leq M$ .

**Mobile agents** For  $k = 1, \dots, H$ , for every child  $L_j$  of  $L_{M+k}$ , agent  $A_{M+k}$  moves to  $L_j$  and requests the local agent  $A_j$  to locally send the list of samples. It adds the list to the sum in its data space, and moves back to  $L_{M+k}$ .

The transmission complexity of KDEC under any of the protocols above is  $O(hops \cdot |\{z \in \mathbb{Z}^n \cap R(z_1, z_2)\}|)$ , where  $hops$  is the total number of hops of the network.

#### 4.4. Issues of data security in KDEC

The various KDEC protocols transmit only density estimates outside the physical data space of a site. However, avoiding network data transmission does not guarantee that data cannot be disclosed by appropriate techniques. In fact, a large body of research in statistical databases has been motivated by the observation that many types of aggregates contain implicit information that allows for data reconstruction, possibly obtained by posing carefully designed sequences of aggregate queries [6]. To the best of our knowledge, the problem of data reconstruction from kernel density estimates has not been investigated. Moreover, sites in open environments can be guaranteed neither to run provably correct code nor to be secure, and they can release information the protocol has been designed to hide. Finally, unsecured mobile agents are vulnerable to malicious servers which attempt to tamper with their code and data to their advantage [29,36]. Therefore, the security of the approach must be carefully evaluated. In particular, the following questions must be answered:

1. Are kernel density estimates secure, i.e., is it possible to disclose the original dataset from a kernel density estimate computed on that dataset?
2. Can a site improve the accuracy of the data disclosure by exploiting knowledge available to all sites taking part in the protocol?
3. Can a subset of the sites participating to an execution of the protocol improve the accuracy of their disclosure attempt by forming a coalition?

In the following, we will try to answer the questions above and discuss the strengths and weaknesses of potential countermeasures. Discussing security issues in agent platforms is outside the scope of this work. We assume authorization and authentication mechanisms to run mobile code are secure and authenticated and secure communication protocols between sites are used. Consequently, a site always knows the sender or receiver of any message.

##### 4.4.1. Inference attacks on kernel density estimates

In any of the proposed protocol, all parties learn the global kernel density estimate. Even if all parties strictly follow the protocol, the privacy of data is not guaranteed unless it can be argued that density estimates do not contain enough implicit information to allow for reconstructing the data. Two techniques may apply: Iterative disclosure and non-linear system solving.

**Iterative disclosure** A simple form of attack consists in searching the density estimate or its derivatives for discontinuities. For example, if  $K(\mathbf{x})$  equals  $w(\mathbf{x})$ , then the distance between discontinuities of the estimate on the same axis equals the distance between data objects on that axis. Therefore the relative positions of objects are known. If the window width  $h$  is known, then all objects are known since every discontinuity is determined by an object lying at distance  $h$  from the discontinuity, on the side where the estimate is greater. Kernels whose derivative is not continuous, such as the Epanechnikov kernel, allow for similar inferences. For the general case, in [4] a simple algorithm has been presented, which partially discloses the data by reconstructing one object at the time.

**Non-linear system solving** Let  $g : \mathbb{R}^d \rightarrow \mathbb{R}$  be extensionally equal to a kernel estimate  $\hat{\varphi}[D](\mathbf{x})$ , that is,  $(\forall \mathbf{x} \in \mathbb{R}^d) g(\mathbf{x}) = \hat{\varphi}[D](\mathbf{x})$  holds, and let  $K$  and  $h$  be known. The problem is to compute  $\mathbf{x}_i, i = 1, \dots, N$ . In an inference attack to a KDEC-based clustering,  $g(\mathbf{x})$  will be a reconstructed estimate effectively computed by the interpolation formula Eq. (7).

An attacker can select  $Nd$  space objects  $\mathbf{y}_j$  and attempt to solve a system of equations

$$\sum_{i=1}^N K \left( \frac{1}{h} (\mathbf{y}_j - \mathbf{x}_i)^T (\mathbf{y}_j - \mathbf{x}_i) \right) = g(\mathbf{y}_j), \quad (8)$$

$$j = 1, \dots, Nd.$$

Although the resulting system of equations is non-linear (assuming the normal kernel) and contains a large number of unknowns, several efficient methods to solve non-linear systems of equations have been proposed in the literature [24]; therefore solving system Eq. (8) is not necessarily unfeasible even for large datasets. On the other hand, the accuracy and speed of convergence of such methods still depend on the structure of the problem at hand. Preliminary investigations have shown that an attacker is likely to incur slow speed of convergence or a large number of spurious solutions when trying to solve systems like Eq. (8).

##### 4.4.2. Protocol attacks

Even if one of the agents could disclose all data objects, by inference alone it could not discover at which site a given reconstructed object resides. However, in each of the protocols of Section 4.3, semi-honest [12] or malicious behaviours by one agent or a coalition of agents could substantially reduce the uncertainty about the location of disclosed objects. In the following we

describe potential attack scenarios in each of the protocols.

**Sequence protocol** In the sequence protocol with stationary agents, the  $n$ -th agent in the arrangement  $A_{\pi(n)}$  knows the density estimate of the data at sites  $L_{\pi(1)}, \dots, L_{\pi(n-1)}$  and, by difference, the density estimate of the data at  $L_{\pi(n+1)}, \dots, L_{\pi(M)}$ . Therefore a semi-honest agent can infer the data objects but the amount of information that is learned by  $A_{\pi(n)}$  on the location of the objects depends on the position of  $L_{\pi(n)}$  in the arrangement. In particular, if  $\pi(n) = M - 1$  or  $\pi(n) = 2$ , objects can be assigned to  $L_{\pi(M)}$  or  $L_{\pi(1)}$ . To refine their knowledge, two malicious agents  $A_{\pi(n-p)}, A_{\pi(n+1)}$ ,  $0 < p < n < M$  can collude and, by difference, calculate the estimate of the union of the datasets at  $L_{\pi(n-p+1)}, \dots, L_{\pi(n)}$ . If  $p = 1$ , the reconstructed objects can be assigned to  $L_{\pi(n)}$ .

In the mobile agent case, the protocol is secure if agents are assumed semi-honest, as the partial density estimates are stored in the mobile agent's data space only. If local agents are potentially malicious, then data and code of the mobile agent can be tampered with and virtually all information that is learned by the mobile agent at any site could be learned by some malicious agent. For example, the malicious agent, say  $A_{\pi(e)}$ , could read the summation of samples of already visited sites  $s_{e-1}(\mathbf{z}) = \sum_{1 \leq k < e} \hat{\varphi}[D_{\pi(k)}](\mathbf{z} \bullet \boldsymbol{\tau})$  for all  $\mathbf{z} \in R(\mathbf{z}_1, \mathbf{z}_2)$ . Worse, the agent's code could be modified to forward to  $A_{\pi(e)}$  any local density estimate obtained after visiting its site.

*Example.* Suppose the sequence protocol with mobile agents is used. The master helper's agent  $A_9$  randomly selects the permutation 5, 1, 4, 2, 6, 8, 3, 7 and starts its navigation in the sequence. Assume  $A_1, A_2, A_3$  are malicious, i.e., they actively attempt to read the data space of  $A_9$  and tamper with its code, exploiting privileged accounts in their respective sites.  $A_1$  reads the samples of  $L_5$  from  $A_9$ 's data. Then, it forwards the sum of its samples and  $L_5$  to  $A_2$ .  $A_2$  reads the cumulative samples of sites  $L_5, L_1, L_4$  from  $A_9$ 's data, and, by difference, learns the samples of  $L_4$ . Then,  $A_2$  forwards to  $A_3$  the sum of samples of  $L_5, L_1, L_4, L_2$ , and tampers with the code of the mobile agent, reprogramming it to keep a separate copy of the current sum of samples, before adding the samples of  $L_8$ .  $A_3$  reads the sum of samples of preceding sites from  $A_9$  and the separate copy, and learns the samples of  $L_6$  and  $L_8$ . When the global sum is transmitted to all parties,  $A_3$  learns the samples of  $L_7$ .

**Star protocol** In the stationary case, the helper has complete knowledge of the global density estimate and

the local density estimates. One or more data sites could collude with the helper and have it send them the local density estimates of each of the remaining sites, from which local data could be inferred and correctly assigned. Without the collusion of the helper, the malicious sites can only obtain the density estimate of the remaining sites and infer the data without any assignment (unless there are only two data sites).

In the mobile agent case, code tampering is useless as the agent code is loaded once and returns to the originator immediately after collecting the samples. As communications are assumed authenticated, a malicious data site cannot impersonate the helper and send the agent to another data site to collect the samples of its local estimate. The attack scenario is therefore similar to the stationary case.

**Tree protocol** In the stationary case, a helper site and its children data sites form a star and the collusion scenario of the previous paragraph applies. To infer and assign data objects outside the local star, the malicious agents must collude with other helper agents. For instance, the malicious agents could collude with the master helper to learn the structure of the tree, which is not published, and thus attempt to collude with helpers having data sites as children of their sites. Alternatively, the malicious agents could attempt to set up a path of colluded agents until a helper agent having data sites as children of its site is encountered, by iteratively asking the last colluded agent a reference to the parent or a child. The scenarios in the mobile case are similar (see the paragraph on the Star protocol).

*Example.* Assume the initiator has set the minimum tree order  $b$  to 3, and, therefore, has collected 3 auxiliary helpers:  $L_{10}, L_{11}$ , and  $L_{12}$ . The master helper's agent  $A_9$  computes the tree and finds out that one of the auxiliary helpers has exactly two children. As  $b = 3$ , reassigning any data site is useless. Therefore,  $A_9$  sets the tree order  $b$  to 4 and recomputes the tree. The tree has now the following edges:  $(L_9, L_{10}), (L_9, L_{11}), (L_9, L_3), (L_9, L_7), (L_{10}, L_5), (L_{10}, L_1), (L_{10}, L_4), (L_{10}, L_2), (L_{11}, L_6), (L_{11}, L_8)$ . Again, an auxiliary helper,  $L_{11}$ , has exactly two children. In this case, however, the parent of a data site, other than  $L_6, L_8$ , can be changed to  $L_{11}$ . Assume  $(L_{10}, L_2)$  is changed to  $(L_{11}, L_2)$ . Let us suppose  $A_1, A_2, A_3$  are malicious. If  $A_1$  and  $A_2$  collude with  $A_{10}$  and  $A_{11}$ , they learn the sample sums  $\sum_{k \in \{1,4,5\}} \hat{\varphi}[D_k](\mathbf{z} \bullet \boldsymbol{\tau})$ ,  $\sum_{k \in \{2,6,8\}} \hat{\varphi}[D_k](\mathbf{z} \bullet \boldsymbol{\tau})$ . By difference, the sample sums  $\sum_{k \in \{4,5\}} \hat{\varphi}(\mathbf{z} \bullet \boldsymbol{\tau})$ ,  $\sum_{k \in \{6,8\}} \hat{\varphi}(\mathbf{z} \bullet \boldsymbol{\tau})$  are learned by malicious agents  $A_1$  and  $A_2$ , respectively. When the global sum of samples is returned to every agent, mali-

icious agent  $A_3$  learns by difference the samples of  $L_7$ , i.e.,  $\hat{\varphi}[D_7](z \bullet \tau)$ , for all  $z \in R(z_1, z_2)$ . Therefore, if the data objects can be reconstructed from the sample sums above,  $D_7$ ,  $D_4 \cup D_5$ , and  $D_6 \cup D_8$  are known to the malicious agents. The agents cannot however decide whether a reconstructed data object in  $D_4 \cup D_5$  belongs to  $L_4$  or  $L_5$ , and whether a reconstructed data object in  $D_6 \cup D_8$  belongs to  $L_6$  or  $L_8$ .

#### 4.4.3. Countermeasures

The attacks in Sections 4.4.1 and 4.4.2 exploit two types of vulnerabilities: (i) given a kernel density estimate, data objects are reconstructible and (ii) partial density estimates are not secured against semi-honest or malicious agents. Countermeasures to these attacks which may apply have been investigated in the literature on both Secure Multiparty Computation and Mobile Cryptography.

**Secure multiparty computation** The goal of *Secure Multiparty Computation* (SMC) [12] is to allow two or more parties to compute the value of a function on an input which is distributed among the parties in such a way that at the end of the computation each party knows nothing except the value of the function and its own input. A simple SMC technique which can be applied to KDEC with the sequential protocol is *Secure Sum* [3]: For all  $z \in R(z_1, z_2)$ , the helper agent  $A_{M+1}$  generates randomly  $r(z) \in [0, 1]$  and sends  $r(z)$  to  $A_{\pi(1)}$ . For  $1 \leq n \leq M$ , agent  $A_{\pi(n)}$  receives  $v_{n-1} = (r(z) + s_{n-1}(z)) \bmod 1$  and sends to  $A_{\pi(n+1)}$  the value  $v_n = (r(z) + s_n(z)) \bmod 1 = (v_{n-1} + \hat{\varphi}[D_{\pi(n)}](z \bullet \tau)) \bmod 1$ . Finally,  $A_{M+1}$  sends to  $A_{\pi(n)}$ ,  $1 \leq n \leq M$ , the difference  $v_M - r(z)$ . Both in the stationary and mobile protocol,  $A_{\pi(n)}$ ,  $1 \leq n \leq M + 1$ , learns nothing about the density estimate of the other sites or group of sites, even if agents are malicious, since the samples of partial estimates that are moved between sites are uniformly distributed in  $[0, 1]$ . (The list of samples of the global density estimate is the result of the computation and is known by all sites, independent of the way it is computed.) Consequently, in the stationary case a semi-honest agent cannot assign reconstructed objects, independent of its position in the arrangement.

The secure sum is vulnerable to collusion attacks. Agents can easily learn their relative positions in the arrangement by comparing successors and predecessors, and colluded agents  $A_{\pi(n-1)}$  and  $A_{\pi(n+1)}$  can easily compute  $\hat{\varphi}[D_{\pi(n)}](z \bullet \tau)$  by comparing  $v_n$  and  $v_{n-1}$ . Dividing each  $r(z)$  into shares and permuting the arrangement for each share prevents malicious

agents from having the same neighbour [3]. Obviously, since the crucial secrets  $r(z)$  are generated by the master helper's agent, any security failure of the master helper or collusion of its agent makes the Secure Sum approach useless.

In the mobile agent case, the above type of collusion is not possible as long as the mobile agent keeps its path secret. However, when the mobile agent moves to a new site, the destination address must be shared with the host.

**Mobile cryptography** In [29] Sander and Tschudin introduce the term "mobile cryptography" to denote fully software based cryptographic solutions to the problem of designing secure mobile programs, and define two basic scenarios: Computing with Encrypted Data (CED) and Computing with Encrypted Functions (CEF). In the CED scenario, Alice wants to know the output of Bob's private algorithm for function  $f$  on her private input  $x$ ; nothing else must be learned by Alice or Bob. In the CEF scenario, Alice wants to know the output of her private algorithm for function  $f$  at Bob's private input  $x$ ; nothing else must be learned by Alice or Bob. It turns out that, when  $f$  is a polynomial, both CED and CEF can be implemented using a *Homomorphic Encryption Scheme* (HES). An algebraic HES is an encryption function  $E : R \rightarrow S$ , where  $R$  and  $S$  are rings, such that  $E(x \diamond y)$  can be efficiently computed from  $E(x)$  and  $E(y)$ , where  $\diamond$  is a ring operation.

As density estimate samples are summed, in the sequel we assume  $E$  to be only additively homomorphic, that is, there exists an efficient algorithm `PLUS` to compute  $E(x + y)$  from  $E(x)$  and  $E(y)$ . A CED approach in KDEC applies naturally both to the sequential protocol with mobile agents and the star protocol with stationary agents, as follows. Agent  $A_n$ ,  $1 \leq n \leq M$ , sends  $E(\hat{\varphi}[D_n](z \bullet \tau))$  to the agent which computes the partial sum of samples, i.e.  $A_{M+1}$ . Algorithm `PLUS` must be made known to  $A_{M+1}$  in advance, e.g. by  $A_1$ . The helper  $A_{M+1}$  uses `PLUS` to sum encrypted samples:  $E(s_M(z)) = E(\sum_{k=1}^M \hat{\varphi}[D_k](z \bullet \tau)) = \text{PLUS}(E(\hat{\varphi}[D_1](z \bullet \tau)), \text{PLUS}(\dots, E(\hat{\varphi}[D_M](z \bullet \tau)))$ . Finally  $A_{M+1}$  sends  $E(s_M(z))$  to all agents  $A_n$ ,  $1 \leq n \leq M$ . Such implementation of CED in KDEC effectively hides the samples from the helper agent, however it is not effective against collusions between the helper and malicious agents, as  $E$  must be known by all  $A_n$ , which could decrypt any set of samples maliciously forwarded by  $A_{M+1}$ .

#### 4.4.4. Untrustworthy helpers

Recently, we have witnessed increasing interest towards trustworthiness and referrals as a means to ascer-

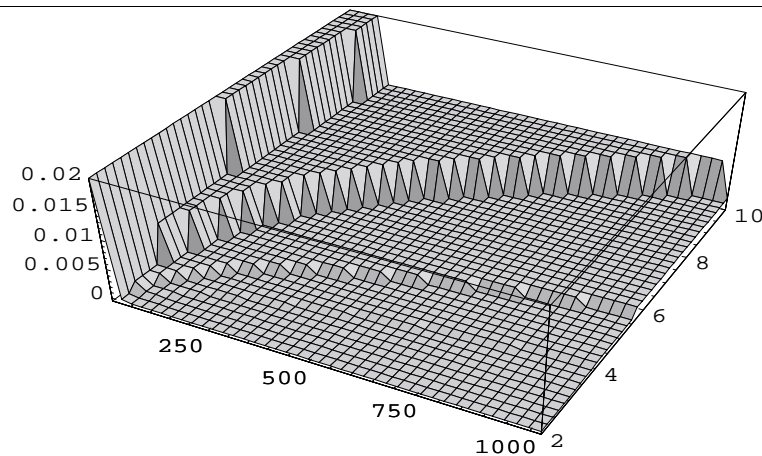


Fig. 7. Graph of the upper bound on  $P(m, b)$  ( $p = 0.2$ ,  $M = 1024$ ).

tain the degree of trustworthiness of an agent [37,38]. In the following we assume the agent community supports referrals about an agent's reputation as a helper. We model a helper's *reputation* as a binary random variable with probability  $p$  and  $1 - p$ , where  $p$  is the probability that the helper will behave as untrustworthy in the forthcoming interaction. We assume that  $p$  can be derived from referrals during the initial negotiation phase.

One way to measure the risk of data privacy infringement in KDEC is the probability  $P(m)$  that an agent's local data are not protected against at least  $m$  other participating agents. If only one helper is used, it is apparent that  $P(m) = p$ , for every  $m$ . If the helpers and the site agents are arranged to form a complete  $b$ -ary tree, it is not difficult to see that  $P(m, b) \leq p^{\lceil \log_b(m+1) \rceil}$ . Such an upper bound decreases with  $m$  and increases with  $b$  according to intuition (see Fig. 7), however, it is worth noting that the lower  $b$ , the higher the chance that an agent could incur coalition attacks. Notably, the best performance against untrustworthiness is obtained by a binary tree, which should always be rejected by any site agent since it gives complete information to each member of any pair of siblings in the tree about the other member's density estimate. Techniques to find a trade-off are under investigation.

## 5. Related work

Only a few approaches to solve the problem of homogeneous distributed data clustering are available to date.

In [21] a solution to the problem of homogeneous distributed clustering under an information theoretic privacy constraint is proposed.

A global probabilistic model of the data is built by combining the parameters of the models computed at the different sources. The approach differs from ours in that a particular parametric family of models must be assumed, whereas the KDEC scheme is non-parametric.

In [33] the  $k$ -windows approach to data clustering is extended to handle the distributed case. Initially, the  $k$ -windows algorithm is executed locally at every data site. Then the generated windows are sent to a central node which is responsible for the final merging of the windows and the construction of global clusters.

In [19] the abstract KDEC scheme is described in more detail. However, its agent implementations and their security issues are not discussed. A shorter version of the present paper is contained in [20]. Possible agent implementations of the KDEC scheme are briefly sketched and the possibility of inference attacks on kernel estimates is suggested. The impact of untrustworthy helpers is also considered. However, no countermeasures have been proposed. An algorithm to perform an inference attack on a kernel estimate is presented in [4]. It has been experimentally proven that the algorithm effectively recovers objects located in the tails of a kernel estimate.

## 6. Conclusion and future work

The ever growing amount of data that are stored in distributed form over networks of heterogeneous

and autonomous sources poses several problems to research in knowledge discovery and data mining, such as communication minimization, autonomy preservation, scalability, and privacy protection. In this paper, we have reviewed prominent approaches in the literature and discussed the benefits that agent-based data mining architectures provide in coping with such problems, and the related issues of data security and trustworthiness. We have presented a scheme for agent-based distributed data clustering based on density estimation, which exploits information theoretic sampling to minimize communications between sites and protect data privacy by transmitting density estimation samples instead of data values outside the site of origin. Potential privacy violations due to inference and coalition attacks and issues of trustworthiness have been discussed. Ongoing research will focus in particular on the investigation of inference attacks on kernel density estimates exploiting recent advances in numerical methods for the solution of non-linear systems of equations, and the analysis of risks of security and privacy violations in DDM environments. Finally, it is planned to implement a multiagent system for KDEC-based homogeneous DDC able to work in peer-to-peer networks and grid computing systems.

## References

- [1] S. Bailey, R. Grossman, H. Sivakumar and A. Turinsky, *Pa-pyrus: a system for data mining over local and wide area clusters and super-clusters*, In Proc. Conference on Supercomputing, 63, ACM Press, 1999.
- [2] M.-S. Chen, J. Han and P.S. Yu, Data mining: an overview from a database perspective, *IEEE Trans. On Knowledge And Data Engineering* 8 (1996), 866–883.
- [3] C. Clifton, M. Kantarcioglu, J. Vaidya, X. Lin and M.Y. Zhu, Tools for privacy preserving distributed data mining, *ACM SIGKDD Explorations Newsletter* 4(2) (2002), 28–34.
- [4] J. Costa da Silva, M. Klusch, S. Lodi and G. Moro, Inference attacks in peer-to-peer homogeneous distributed data mining, in: *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI 2004)*, R. López de Mántaras and L. Saitta, eds, Valencia, Spain, 22–27 August 2004, pp. 450–454, IOS Press.
- [5] M. Ester, H.-P. Kriegel, J. Sander and X. Xu, *A density-based algorithm for discovering clusters in large spatial databases with noise*, in Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), Portland, OR, 1996, 226–231.
- [6] C. Farkas and S. Jajodia, The inference problem: A survey, *ACM SIGKDD Explorations Newsletter* 4(2) (2002), 6–11.
- [7] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth and R. Uthurusamy, *Advances in Knowledge Discovery and Data Mining*, AAAI Press/MIT Press, 1996.
- [8] T. Finin, Y. Labrou and J. Mayfield, KQML as an agent communication language, in: *Software Agents*, J. Bradshaw, ed., MIT Press, 1997, pp. 291–316.
- [9] I.T. Foster, N.R. Jennings and C. Kesselman, *Brain meets brawn: Why grid and agents need each other*, In 3rd International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2004), 19–23 August 2004, 8–15, New York, NY, USA, 2004, IEEE Computer Society.
- [10] Foundation for Intelligent Physical Agents. FIPA Communicative Act Library Specification, Aug. 2001. Published on August 10th, 2001, <http://www.fipa.org>.
- [11] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, New York, NY, USA, 1972.
- [12] O. Goldreich, Secure multi-party computation. <http://www.wisdom.weizmann.ac.il/~oded/pp.html> October 2002, *Final (incomplete) Draft*, version 1.4.
- [13] D. Heckerman, H. Mannila, D. Pregibon and R. Uthurusamy, eds, *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, Newport Beach, California, USA, 1997, AAAI Press.
- [14] J.R. Higgins, *Sampling Theory in Fourier and Signal Analysis*, Clarendon Press, Oxford, 1996.
- [15] A. Hinneburg and D.A. Keim, *An efficient approach to clustering in large multimedia databases with noise*, In Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98), 58–65, New York City, New York, USA, 1998, AAAI Press.
- [16] H. Kargupta, I. Hamzaoglu and B. Stafford, *Scalable, distributed data mining using an agent-based architecture*, In Heckerman et al. [13], 211–214.
- [17] H. Kargupta, B. Park, D. Hershberger and E. Johnson, *Advances in Distributed and Parallel Knowledge Discovery*, chapter 5, Collective Data Mining: A New Perspective Toward Distributed Data Mining, 131–178. AAAI/MIT Press, 2000.
- [18] M. Klusch, Information agent technology for the internet: A survey. Data and Knowledge Engineering, Special Issue on Intelligent Information Integration, *Elsevier Science* 36(3) (2001), 337–372.
- [19] M. Klusch, S. Lodi and G. Moro, *Distributed clustering based on sampling local density estimates*, In Proceedings of the 19th International Joint Conference on Artificial Intelligence, IJCAI-03, 485–490, Acapulco, Mexico, August 2003. AAAI Press.
- [20] M. Klusch, S. Lodi and G. Moro, *The role of agents in distributed data mining: Issues and benefits*, In Proceedings of the 2003 IEEE/WIC International Conference on Intelligent Agent Technology (IAT 2003), 211–217, Halifax, Canada, October 2003. IEEE Computer Society -Web Intelligence Consortium (WIC), IEEE Computer Society Press.
- [21] S. Merugu and J. Ghosh, Privacy-preserving distributed clustering using generative models, In Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM 2003), 19–22 December 2003, Melbourne, Florida, USA, *IEEE Computer Society* (2003).
- [22] G. Moro, G. Monti and A.M. Ouksel, Merging G-Grid P2P systems while preserving their autonomy, in: *Proceedings of the MobiQuitous '04 Workshop on Peer-to-Peer Knowledge Management (P2PKM 2004)*, I. Zaihrayeu and M. Bonifacio, eds, August 22, 2004, volume 108 of CEUR Workshop Proceedings, Boston, MA, USA, 2004. CEUR-WS. org.
- [23] G. Moro and C. Sartori, Incremental maintenance of multi-source views, in: *Proceedings of 12th Australasian Database Conference, ADC 2001, Brisbane, Queensland, M.E. Orłowska and J. Roddick, eds, Australia, IEEE Computer Society, February 2001, pp. 13–20.*
- [24] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer 2000.



- [25] A.M. Ouksel and G. Moro, G-Grid: A class of scalable and self-organizing data structures for multi-dimensional querying and content routing in P2P networks, in: *Agents and Peer-to-Peer Computing, Second International Workshop, AP2PC 2003*, G. Moro, C. Sartori and M.P. Singh, eds, July 14, 2003, Revised and Invited Papers, volume 2872 of Lecture Notes in Computer Science, Melbourne, Australia, 2004. Springer, pp. 123–137.
- [26] M.P. Papazoglou and G. Schlageter, *Cooperative Information Systems – Trends and Directions*, Academic Press Ltd, London, UK, 1998.
- [27] M. Prasad and V. Lesser, *Learning situation-specific coordinating in cooperative multi-agent systems*, Autonomous Agents and Multi-Agent Systems, 1999.
- [28] S. Ratnasamy, P. Francis, M. Handley, R. Karp and S. Schenker, *A scalable content-addressable network*, In SIGCOMM '01: Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, New York, NY, USA, 2001. ACM Press, 161–172.
- [29] T. Sander and C.F. Tschudin, Protecting mobile agents against malicious hosts, in: *Mobile Agents and Security*, G. Vigna, ed., volume 1419 of Lecture Notes in Computer Science, Springer-Verlag, Berlin Heidelberg, 1998, pp. 44–60.
- [30] S. Sen, A. Biswas and S. Gosh, *Adaptive choice of information sources*, In Proc. 3rd International workshop on Cooperative Information Agents. Springer, 1999.
- [31] B.W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman and Hall, London, 1986.
- [32] S.J. Stolfo, A.L. Prodromidis, S. Tselepis, W. Lee, D.W. Fan and P.K. Chan, *JAM: Java agents for meta-learning over distributed databases*, In Heckerman et al. [13], 74–81.
- [33] D.K. Tasoulis and M.N. Vrahatis, *Unsupervised distributed clustering*, In Proceedings of the IASTED International Conference on Parallel and Distributed Computing and Networks, Innsbruck, Austria, 2004.
- [34] W. Theilmann and K. Rothermel, *Disseminating mobile agents for distributed information filtering*, In Proc. of 1st International Sympos. on Mobile Agents, 152–161, IEEE Press, 1999.
- [35] M. Wooldridge, Intelligent agents: The key concepts, in: *Multi-Agent-Systems and Applications*, (vol. 2322 of LNCS) V. Marík, O. Stepánková, H. Krautwurmova and M. Luck, eds, Springer-Verlag, 2002, pp. 3–43.
- [36] B.S. Yee, *A sanctuary for mobile agents*, In Secure Internet Programming, 1999, 261–273.
- [37] P. Yolum and M.P. Singh, An agent-based approach for trustworthy service location, in: *Agents and Peer-to-Peer Computing, First International Workshop, AP2PC 2002, July, 2002, Revised and Invited Papers*, (vol. 2530 of Lecture Notes in Computer Science), G. Moro and M. Koubarakis, editors, Bologna, Italy, 2003. Springer, pp. 45–56.
- [38] B. Yu and M.P. Singh, *Emergence of agent-based referral networks*, In Proc. AAMAS 2002, ACM Press, 2002, 1268–1269.
- [39] N. Zhong, Y. Matsui, T. Okuno and C. Liu, Framework of a multi-agent kdd system, in: *Proc. of Intelligent Data Engineering and Automated Learning – IDEAL 2002*, H. Yin, N.M. Allinson, R. Freeman, J.A. Keane and S.J. Hubbard, eds, Third International Conference, Manchester, UK, (vol. 2412 of Lecture Notes in Computer Science), Springer-Verlag, 2002, pp. 337–346.



## Quantum Agent-Based Service Coordination



---

## Introduction

One vision of agent-based service coordination draws upon the potential evolution of the Internet into the so-called Quantum Internet<sup>1</sup> by extending the former with sophisticated quantum computing and communication devices until 2020 and beyond. Regarding the development of an impressive variety and number of agent-based business applications and services for the Internet and Web since the 1980s (cf. part four), we further envision the same to take place for the future Quantum Internet. One key challenge of realizing this vision is to enable intelligent agents and multiagent systems making appropriate use of quantum computing and communication means to pursue their tasks on networked quantum and digital computers on behalf of their users, and other agents. The cross-disciplinary research on theory, practice, and application of quantum computational agents world wide has just begun.

In this final part of the work, we first introduce the reader to the basics of quantum computing and communication in very brief, and then present our pioneering work on quantum agents and multi-agent systems we started at DFKI in 2004 (cf. chapter 18), with special focus on quantum agent-based service matchmaking.

### Quantum Computing in Brief

Quantum computing technology builds upon quantum physics<sup>2</sup> and promises to eliminate some of the problems associated with the rapidly approaching

---

<sup>1</sup> The term "Quantum Internet" has been introduced by Seth Lloyd at MIT and his colleagues in 2004 [240]. It captures the extension of the Internet with quantum computing devices. Communication between hybrid quantum computing devices in the Quantum Internet could base on both classical TCP/IP and non-classical quantum communication protocols depending on the technical infrastructure in place.

<sup>2</sup> The standard mathematical framework for quantum mechanics that describes quantum physical effects is essentially due to Paul Dirac (1927) and John von Neumann (1932)[374]. It bases on the seminal work in the field by quantum

ultimate limits to classical computers imposed by the fundamental law of thermodynamics. In fact, according to Gordon Moores first law on the growth rate of classical computing power, and the current advances in silicon technology, it is commonly expected that these limits will be reached around 2020. By then, the size of microchip components will be on the scale of molecules and atoms such that quantum physical effects will dominate, hence irrevocably require effective means of quantum computation, and quantum software engineering as well.

### *Origins of quantum computing*

The idea of quantum computing in its own rights dates back to the observation made by nobel laureate and quantum physicist Richard Feynman in 1982 (Feynman, 1982)[117] that an "universal simulator" could mimic the behaviour of any finite physical object down to the quantum level of atoms and photons.<sup>3</sup> A few years later, David Deutsch claimed that all the computational capabilities of any finite machine obeying the laws of quantum computation are contained in a quantum Turing machine (Deutsch, 1985)[97]. This led to the quantum version of the Church-Turing thesis of classical computing: There exists, or can be built a universal quantum computer that can be programmed to perform any computational task which can be performed by any physical object. In other words, classical computing can be simulated by quantum computing. Whether it can essentially go beyond we do not know yet.

### *Quantum bits, registers and measurement*

Quantum computers are devices that store and process information on any physical system with quantum states, so-called quantum systems, such as spins, photons, and atoms. Quantum computation provides a paradigm for information processing that differs fundamentally from ordinary digital computation and is built on the concept of the quantum bit or *qubit*. Any isolated physical 2-state quantum system is appropriate to realize a single qubit  $\psi$  associated to its state space, a complex 2-dimensional Hilbert space  $H_2 = \text{span}\{|0\rangle, |1\rangle\}$  with orthonormal computational standard basis. A quantum state  $|\psi\rangle$  of qubit  $\psi$  is described by a *coherent superposition*, that is  $|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$  with (probability) amplitudes  $\alpha_0, \alpha_1$ . Each squared absolute amplitude value denotes the probability  $p(|\psi\rangle = |i\rangle) = |\alpha_i|^2$  that the system will be in one of the basis states with normalization condition  $|\alpha_0|^2 + |\alpha_1|^2 = 1$ . In contrast to classical probabilities, the amplitudes may

---

physicists Max Planck, Niels Bohr, Albert Einstein, Werner Heisenberg, Erwin Schrödinger, Max Born, Wolfgang Pauli, and others early last century.

<sup>3</sup> The term "quantum" refers to discrete units of physical quantities of an atom (observables) such as its position, spin and momentum. Quantum effects such as quantum state entanglement have been demonstrated not only on the scale of photons and atoms but molecules (e.g. Quantum teleportation of so-called fullerene molecules).

also take negative values which accounts for destructive (quantum) interference.<sup>4</sup>

The state space  $H_2^{\otimes n}$  of a quantum system composed of  $n$  single qubits  $\psi_i$  is the  $n$ -folded *tensor product* of the state spaces of its  $n$  constituting qubits  $H_2^{\otimes n} = \bigotimes^n H_2$ . Such system can be regarded as a *n-qubit register*  $\Psi$  with  $2^n$  computational basis states. If a state  $|\Psi\rangle$  of a  $n$ -qubit register can be written as a product of its constituting qubits in the form  $|\Psi\rangle = \bigotimes_i (\sum_j \alpha_{i,j} |j\rangle)$ , then  $|\Psi\rangle$  is called separable.

A *projective measurement* of  $\Psi$  is described by a set of pairwise orthogonal subspaces  $W_1, \dots, W_m$  satisfying  $H_2^{\otimes n} = \bigoplus_{k=1}^m W_k$  and results in  $j \in \{1, \dots, m\}$ . Let  $\{|\Phi_i^j\rangle\}$  define an orthonormal basis of subspace  $W_j$ , then the operator  $P_j = \sum_{i=1}^{\dim(W_j)} |\Phi_i^j\rangle\langle\Phi_i^j|$  projects  $\Psi$  on the subspace  $W_j$ . The probability of measuring  $j \in \{1, \dots, m\}$  is then given by  $\langle\Psi|P_j|\Psi\rangle$ . Time evolution of a  $n$ -qubit register  $\Psi$  is described by *unitary*, hence deterministic, linear transformations in its state space  $H_2^{\otimes n}$ . Quantum measurement is the only non-deterministic and irreversible operation of quantum computing.

### *Quantum parallelism and memory*

The linearity of quantum mechanics allows for quantum parallel computation, that is the simultaneous computation of some function  $f$  on  $2^n$  superposed classical states with  $n$  qubits. If combined with quantum (state) interference, it is possible to compute a property of  $f$  by just one evaluation of a combination of interfering alternative values of  $f$ . Classical computing can only evaluate different but mutually excluding alternative values of  $f$  with equal probability. Furthermore, in contrast to classical memory, quantum memory increases exponentially in the size of the number of qubits stored in a quantum register: The (joint) state space of any  $n$ -qubit ( $n+m$ ) register takes  $2^n$  ( $2^{n+m}$ ) amplitudes.

### *Principles of no-cloning and uncertainty of measurement*

Unlike classical bits, unknown qubit states cannot be perfectly copied. This principle of no-cloning (Wootters et al., 1982) is closely related to Heisenberg's

<sup>4</sup> In general, a complex wave-function  $\psi$  (equivalently, the quantum state vector  $\boldsymbol{\psi} = (\alpha_1, \dots, \alpha_n)$  in  $H_n$ ) completely describes the behavior of any isolated physical system in a given observable state space (e.g., the position, momentum or spin state space of an electron) with state  $|\psi\rangle = \sum_{i=1}^n \alpha_i |\phi_i\rangle$  of finitely many orthonormal basis states  $|\phi_i\rangle$ , and unitary state evolution over time following the Schrödinger equation. For example, the quantum state  $|\psi\rangle = \alpha_1 |\uparrow_z\rangle + \alpha_2 |\downarrow_z\rangle$  of a spin- $\frac{1}{2}$  particle like electron, proton, neutron, neutrino, and quark in the 2-dimensional spin state space is described by the values of its spin along any given x-, y-, or z-direction such as spin-up  $|\uparrow_z\rangle$  and spin-down  $|\downarrow_z\rangle$  along z-direction: Their linear combinations represent all possible states of the particle's spin. The amplitude values  $|\alpha_1|^2$  and  $|\alpha_2|^2$  denote the probability of obtaining one of the basis states after measuring the spin, that is measuring the state  $|\psi\rangle$  in the spin state basis  $\{|\uparrow_z\rangle, |\downarrow_z\rangle\}$ .

so-called uncertainty relation according to which a perfect measurement of unknown qubit states is impossible. For example, one can find an electron in a particular region around the nucleus of its atom at a particular time with certain probability, but the uncertainty principle of quantum mechanics quantifies the inability to measure both its position and momentum at the same time with arbitrary precision.

### *Entanglement*

Non-separable composite qubit states are called *entangled states*. Entanglement causes the non-locality principle of quantum mechanics: The states of spatially separated but entangled qubits can instantaneously change in a correlated manner by measuring either of them. That is, the result of a measurement performed on one part *A* of a quantum system has a non-local effect on the physical reality of another distant part *B*, in the sense that quantum mechanics can predict outcomes of some measurements carried out at *B*. This non-local correlation of quantum systems is impossible to realize in classical physics with its assumed local realism<sup>5</sup>, and has been particularly objected by Einstein, Rosen and Podolski (1935) in their EPR thought experiment. However, the non-local effect of quantum entanglement was experimentally verified by Alain Aspect in 1982 based on the respective theorem of John Bell (1964) for testing it.

### *Standard Copenhagen and many-world interpretation*

According to the standard Copenhagen interpretation of quantum mechanics, first advocated by Bohr and Heisenberg, observing a physical quantum system in a superposed state by means of measurement causes its respective wave-function to collapse into one of the (classical) basis states with a certain probability. This interpretation rejects the local realism of classical physics by assuming that the wave-function has no physical interpretation of reality (but used as mere mathematical tool to calculate the probabilities of state measurement outcomes). As one consequence, the collapse of the wave-function for entangled states does not imply faster-than-light-speed transmission of information retaining the principle of causality of classical physics (the cause has to precede its effect), but not locality in general. Besides, counterfactual definiteness (CFD) is excluded by Heisenberg's uncertainty principle of quantum mechanics<sup>6</sup>

---

<sup>5</sup> Local realism states that information about the state of a physical system is mediated only by interactions in its immediate (local) surroundings, and the state is real, i.e., it exists and is well-defined before any measurement.

<sup>6</sup> Counterfactual definiteness (CFD) states that every possible measurement, even if not performed, yields a unique, fixed but possibly unknown result. The complementarity (wave-particle duality) of quantum mechanics states that physical reality is determined and defined by complementary pairs of properties of physical entities (like the complementary properties of the position and the momentum



Alternatively, in the non-standard so-called many-worlds or relative-states interpretation of quantum mechanics introduced by Hugh Everett (1952) both realism and locality are retained but CFD is not. The actual (observer-dependent) ad-hoc collapse of the wave-function is replaced by the dynamic (observer-independent) process of quantum decoherence<sup>7</sup>. Roughly, each measurement of a correlated state of a combined (2-state) object-observer system (S+M) causes it to split such that both possible measurement outcomes are simultaneously realized in separate (non-interfering) so-called parallel or alternative worlds. Repeated measurement yields a many-branched event tree where every single branch or complete path represents a complete alternative state measurement history of system S, a world that exists independent from the many other possible ones not accessible to the observer in this world, with certain joint (product) probability. In this sense, the wave-function of S does not collapse but exists and evolves across parallel worlds. This interpretation would resolve many of the paradoxons of quantum mechanics like Schrödinger's cat and EPR but cannot be experimentally verified.

### Quantum vs. Classical computing

Non-measuring (unitary) quantum operations are reversible since any unitary transformation  $U$  has an inverse. As a consequence, quantum computing is equivalent to reversible computing, hence can simulate classical computing in principle: The constraint of unitary evolution of qubit states yields a generalization of the restriction of classical (Turing machine or logic circuit based) models of computation to unitary, hence reversible computation. Quantum computing (BQP, bounded error quantum polynomial time) is as powerful as classical computation (PSPACE):  $P \subseteq BQP \subseteq PSPACE$ .<sup>8</sup>

In particular, quantum computing simulates randomized computing (RP, random polynomial; BPP, bounded error probabilistic polynomial time) since it allows to generate purely (not only pseudo-) randomized bits through non-

---

of an electron) which manifestations are inverse proportionally limited in their precision by the uncertainty principle.

<sup>7</sup> Decoherence occurs when a physical system (object) interacts with its environment (observer, subject) in an irreversible way such that elements of the assumed superposition of the combined object-observer wave-function can no longer interfere with each other, as if the wave-function collapsed.

<sup>8</sup> As mentioned above, this led to the modified Church-Turing thesis of computability theory that "every function which would naturally be regarded as (physically) computable is computable by a mechanical device equivalent to a (quantum) Turing machine". In other words, the theory of computation is now the quantum theory of computation as a quantum Turing machine (QTM) is claimed to be able to simulate any physical system - in this sense being universal for any kind of computation (Deutsch, 1985)[97]. Since the models of (quantum) physical systems to be simulated by the QTM still may change in quantum physics, this claim is debatable.

deterministic measurement of qubits:  $(P \subseteq) RP \subseteq BPP \subseteq BQP$ . It differs from randomized (probabilistic) computing in that non-classical physical effects of entanglement and destructive interference can be exploited for computation. Apart from being significantly faster than classical computers ever can be, it is still not known whether quantum computers can also be more powerful in problem solving, that is whether  $NP \subseteq BQP$  or  $NP \neq BQP$  holds. It is widely believed that this is not the case. Though, there are polynomial quantum solutions of the conjectured NP-hard problems of integer prime factoring and computation of discrete logarithms (Shor, 1997)[333]. Besides, quantum search provides a quadratic speed up of solving the NP-complete Hamiltonian cycle problem and is optimal. However, an exponential speed up of search in logarithmic time would have been required to allow quantum computers to solve all intractable problems in NP ( $NP \subseteq BQP$ ).

### **Quantum vs. Classical Communication**

Quantum entanglement together with the impossibility of perfect qubit copying guarantees perfect communication security against interception by means of quantum cryptography or quantum key distribution (QKD), in particular quantum teleportation (Ekert, 1991). This is not possible to accomplish with means of classical cryptography. Meanwhile, QKD tools are commercially available and have been also applied to, for example, Vickrey (second-price-sealed-bid) auctions in order to ensure perfect privacy of bids against possibly deceiving auctioneers.

While quantum teleportation allows to send  $n$  entangled qubits at the cost of  $2n$  bits with perfect privacy, so-called quantum superdense coding allows one to encode  $n$  bits with  $n/2$  qubits. This is possible by exploiting additional  $n/2$  entangled qubits that are shared among sender and receiver for this purpose in advance. That is in line with the so-called Holevo-bound of quantum information according to which the amount of information carried by one qubit and one classical bit over respective classical and quantum communication channels is the same (Holevo, 1973).

### **Quantum Computing and AI**

What are the implications of quantum physics and quantum computing to the world of AI? Apart from faster computation, perfectly private communication and the promise of leveraging the computational power of machines beyond the boundaries of classical computing, we do not know yet. Cross-disciplinary research on the interaction between AI and quantum physics just started a few years ago. One prominent forum for this research is the annual AAAI Spring Symposium on Quantum Interaction which was initially held in 2007 at Stanford University[50].

### *Selected early results*

From the perspective of agent-based coordination, (Hogg & Huberman, 2004; Chen, Hogg & Huberman, 2007)[168, 65] exploit the effect of quantum entanglement for the development of quantum game-theoretic cooperation and coordination protocols, as well as quantum auction protocols for resource allocation in multiagent systems. Notably, quantum games with entangled state shared among players can be played more efficient in the sense that less information needs to be exchanged, and a greater number of strategies can be chosen from than in classical games. However, it is known that quantum games can ultimately be represented by a more complex classical game, hence are not conceptually different from their classical counterpart. Other recent contributions to the interaction between both fields as reported in, for example, [50] include the use of quantum mechanical concepts for information retrieval and natural language processing.

Our own basic research contributed a generic architecture, classification and programming of quantum agents and multiagent systems for networked hybrid quantum computers. Besides, several quantum agent-based service matchmaking protocols have been developed and simulated by use of open-source quantum simulators (cf. chapters 18 - 20). We also developed and simulated quantum encoded versions of (a) Sandholm's contract net protocol TRACONET for competitive agents, named TRACONET-Q, and (b) the game-theoretic coalition protocol for rational quantum agents, named KCA-Q. Each of these quantum agent coordination protocols allows for both reduced computation and communication complexity in a way not possible with classical computing (Nesbitt, 2008)[273].

### *Quantum mind theories*

However, it is largely disbelieved that quantum computing can contribute to cracking one essential problem of AI, that is the simulation of human intelligent behaviour. The debate about whether a computer will ever be able to be truly artificially intelligent has been going on for decades. It is largely based on philosophical arguments, and has been accompanied by several related shifts of computing paradigms ranging from symbolic logical reasoning, to subsymbolic computing with neural networks, fuzzy and genetic computing, to multiagent-based computing and swarm intelligence, to so-called embodied intelligence without a brain, and artificial life. Even the uniqueness of creativity, emotion, and consciousness to humans are questioned in principle by several AI researchers like Marvin Minsky, Raymond Kurzweil, and Hans Moravec - mostly in favor of regarding the brain as a kind of computing machinery. This point of view is partially supported by research advances in neurophysiology and cognitive science on how the human brain might actually work in the past decade.

A few contributions to this debate have been made by quantum theorists in terms of so-called quantum mind theories and universal quantum computing.

For example, Seth Lloyd at MIT conjectured that every physical object, from the tiniest rock to the brain to the universe as a whole can be regarded as a quantum computer and any detectable physical process including consciousness a quantum computation (Lloyd, 2000)[238]. One prominent quantum mind theory of Sir Roger Penrose and Stuart Hameroff (1997) attempts to explain human consciousness as the result of quantum computational activities and quantum gravity effects in microtubules of the axons of nerve cells in the human brain. However, this theory has been quite vividly debated and eventually refuted in practice: Max Tegmark's experiment in 2000[358] showed that temperatures in the brain in average are too high (not working near absolute zero) for microtubules and neurons to remain in superposed or entangled quantum states long enough to fire before they actually decohere - making it reasonably unlikely that the brain evolves any quantum behavior[358]. Since the late 1960s, researchers of so-called quantum brain dynamics such as Stapp, Marshall, Umezawa, and Vitiello developed other quantum mind theories which are, however, not experimentally verified yet.

### **Will Quantum Computers Ever be Built?**

Since the mid 1980s, tremendous research and development efforts have been made world wide to realize basic quantum computing and communication devices. Remarkable achievements include the physical implementation of multiple-qubit processors, quantum repeaters, and one-way quantum computers exploiting different means of cold ion traps, nuclear magnetic resonance, cavity quantum electrodynamics, or solid state technology like quantum dots. In particular, the first experimental realization of quantum teleportation using photon polarization states has been demonstrated by the research group of Anton Zeilinger in Innsbruck in 1997 [47]. Quantum teleportation-based communication devices are already part of commercially distributed quantum cryptographic products such as those delivered by the company MagiQ Technologies.

The US National Institute of Standards and Technology (NIST)<sup>9</sup> is building a 10-qubit processor, the IBM Almaden Research Center in California together with IBM Thomas Watson Research Center in New York developed a NMR-based quantum computer that factored the number 15, and the Canadian company D-Wave Systems presented the first 16-qubit processor "Orion" in 2007<sup>10</sup> and upgraded it to a 28-qubit processor in November, 2007. In this sense, the main question is not whether but when sophisticated quantum computing devices become eventually affordable for the common user of the future Quantum Internet, or will quantum computers be only used for niche applications?

---

<sup>9</sup> [qubit.nist.gov](http://qubit.nist.gov)

<sup>10</sup> [www.computerbase.de/news/allgemein/forschung/2007/februar/weltweit-quantenprozessor/](http://www.computerbase.de/news/allgemein/forschung/2007/februar/weltweit-quantenprozessor/)

All known quantum algorithms require the determinism and reliability of classical control for the execution of suitable quantum circuits consisting of a finite sequence of quantum gates and measurement operations. This requires a hybrid quantum computer architecture in which classical signals and processing of a classical machine are used to control the timing and sequence of quantum operations carried out in a quantum machine. Therefore, the required total isolation of quantum computing devices from the environment to avoid decoherence of qubit states cannot be guaranteed. Quantum error correction codes and recent progress in nanotechnology hold promise to overcome this problem.

Another main barrier of taking up quantum computing technology in practice may be our own imagination of the potential of quantum software engineering. In general, quantum algorithms appear to be best at problems that rely on promises or oracle settings, hence use some hidden structure of a problem to find an answer that can be easily verified through, for example, means of amplitude amplification. One challenge is not only to design such quantum algorithms for quantum computing machinery that can drastically outperform their counterparts on a classical von-Neumann computer but create hybrid quantum computer interfaces that are also simple and reliable enough for mere mortal users to understand and to use in practice.

### **Practical Applications of Quantum Computing and Communication?**

Only very few practical applications of quantum computing and communication have been developed so far. Prominent examples are Lov Grover's family of quantum search algorithms offering a quadratic speed-up of searching large data sets, and quantum cryptography. Regarding the potential of quantum computers to crack codes much more quickly than any ordinary classical computer could (based on Shor's quantum prime factorization), it comes for no surprise that quantum cryptographic solutions were developed already into commercial products by, for example, IdQuantique<sup>11</sup>, MagiQ Technologies<sup>12</sup>, and D-Wave Systems<sup>13</sup>. Other potential application areas are quantum-based optimization, multiparty computation, controlled quantum teleportation for secure and efficient interaction between agents in space satellites and ground station, and quantum cryptographic conferencing for day-to-day business in the Quantum Internet.

---

<sup>11</sup> [www.idquantique.com](http://www.idquantique.com)

<sup>12</sup> [www.magiqtech.com](http://www.magiqtech.com)

<sup>13</sup> [www.dwavesys.com](http://www.dwavesys.com)

## Further Readings

For a more comprehensive introduction to theory, practice, and application of quantum computing and communication, we refer the interested reader to, for example, the "bible" of the field written by Nielsen and Chuang (2000)[275], and other excellent readings on the topic such as (Bouwmeester et al., 2000)[48], (Stolze & Sutter, 2004) [346], Arroyo-Camejo (2006) [14], and (Bruss & Leuchs, 2007)[49]. Other accessible readings on the subject with in-depth analysis and philosophical treatment of the interpretation of quantum mechanics and their implications include Penrose (1997)[293], Davies and Brown (1986)[85], Polkinghorne (2002)[298], Zeilinger (2003)[392], and the popular science account on alternative universes by Michio Kaku (2005). [239] provides a particularly enjoyable discussion between David Deutsch and Seth Lloyd in 1997 about the many-world interpretation of quantum mechanics. A short account of the history of proposals and discussions about quantum effects in the human brain is given, for example, in (Suppes & de Barros, 2007)[352] and (Marcer & Rowlands, 2007)[248].

Besides, the electronic archive arXive.org/quant-phys hosted by the Cornell University Library serves as a solid source of publications related to all topics of quantum physics, quantum computing, and quantum communication. The first chapter of this part provides a condensed introduction to quantum computing for computer scientists without knowledge on quantum physics.

## The Contributions

In the final chapters of this thesis, I present my contributions to theory and practice of quantum agents and multi-agent systems in terms of a classification and architecture of quantum agents for hybrid quantum computers, several quantum matchmaking algorithms, and first simulation results of quantum pattern matching-based search agents via quantum simulators on a classical machine.

*Chapter 18: Quantum computing and agents: A manifesto.* In this chapter, we provide some first thoughts on, and preliminary answers to the question how intelligent software agents could take most advantage of the potential of quantum computation and communication, once practical quantum computers become available. After introducing the reader to the principles of quantum computing and communication from the perspective of a computer scientist rather than a quantum physicist, we sketch a potential hybrid quantum computer architecture that is derived from different architecture proposals and implementation schemes for quantum computing units and communication reported elsewhere. Based on these precursors, we then present a classification of quantum computational agents and multi-agent systems, and show that their principled capability to perform certain computational tasks more efficient than classically computing agents comes at the cost of limited self-

autonomy, due to non-local effects of quantum entanglement.

*Chapter 19: Programming of quantum search agents.* In this chapter we propose an extension of the classical hybrid agent architecture InteRRap for quantum computing agents, called QuantumInterrap, and demonstrate its principles by means of a type-I quantum search agent for service selection. We show the feasibility of its realization on a hybrid quantum computer by its programming and simulation using three quantum simulators running on a classical computer. This has been joint work with my Master student Rene Schubotz.

*Chapter 20: Quantum matchmaker agents.* Based on the insights of the previous chapters, we provide first quantum-based solutions to the classical coordination problem of service matchmaking in more detail. Each of these quantum matchmaking algorithms performs under certain conditions more efficient and secure than its classical counterpart. Building on these results the implementation of distributed quantum service matchmaker agents in networks of hybrid quantum computers is relatively straightforward. However, it is not known how to conduct agent-based service composition planning or negotiation as outlined, for example, in the previous chapters by means of quantum computing and communication.

### Open problems

Selected open problems of quantum agent-based service coordination are as follows.

- Development of efficient means of quantum coding and quantum processing of logic-based semantics of resources. Relevant work include results on quantum logics [67], and reasoning about quantum knowledge [80].
- Development of quantum versions of classical coordination schemes for multi-agent systems like the extended contract net protocol, and secure distributed quantum brokerage in quantum computer networks with cascading, instantaneous propagation of coordination tokens. Relevant work include the design and physical implementation of distributed quantum computing and networks [331, 79]. My research team started work in this direction with a quantum encoded version of Sandholm's contract net protocol TRACONET for competitive agents (Nesbigall, 2008)[273].
- Development of quantum protocols for rational coalition forming among quantum computational service provider agents. Related works include the whole new field of n-player quantum games [28], in particular, 2-agent quantum cooperative coordination games [168], and quantum coalition leader election procedures [357]. My research team started work in this direction by the development of a quantum encoded version of a game-theoretic coalition algorithm KCA (Nesbigall, 2008)[273].

I admit that whether quantum computers, quantum agents and quantum agent-based service coordination will ever become true in practice still remains speculative at this time. However, not only the tremendous advance of quantum computing technology over the past decade, and the ongoing race for who actually builds the first sophisticated quantum computer world wide, but the just recently started research on quantum computing multiagent systems at the University of Waterloo (Canada), the HP Labs in Palo Alto (USA), and the University of Southampton (UK)[50] makes me strongly believe in the vision of quantum agent-based service coordination in the Quantum Internet.



---

## Quantum Computing and Agents: A Manifesto

M. Klusch: Toward Quantum Computational Agents. In: Computational Autonomy and Agents. M Nickles, M Rovatsos, G Weiss (eds.), Lecture Notes in Artificial Intelligence, 2969, pages 170 - 186, Springer, 2004.

# Toward Quantum Computational Agents

Matthias Klusch

German Research Center for Artificial Intelligence, Deduction and Multiagent Systems, Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany.  
e-mail: klusch@dfki.de

**Abstract.** In this chapter, we provide some first thoughts on, and preliminary answers to the question how intelligent software agents could take most advantage of the potential of quantum computation and communication, once practical quantum computers become available in foreseeable future. In particular, we discuss the question whether the adoption of quantum computational and communication means will affect the autonomy of individual and systems of agents. We show that the ability of quantum computing agents to perform certain computational tasks more efficient than classically computing agents is at the cost of limited self-autonomy, due to non-local effects of quantum entanglement.

## 1 Introduction

Quantum computing technology based on quantum physics promises to eliminate some of the problems associated with the rapidly approaching ultimate limits to classical computers imposed by the fundamental law of thermodynamics. According to Gordon Moore's first law on the growth rate of classical computing power, and the current advances in silicon technology, it is commonly expected that these limits will be reached around 2020. By then, the size of microchip components will be on the scale of molecules and atoms such that quantum physical effects will dominate, hence irrevocably require effective means of quantum computation.

Quantum physics has been developed in the early 1920's by physicists and Nobel laureates such as Max Planck, Niels Bohr, Richard Feynman, Albert Einstein, Werner Heisenberg, and Erwin Schrödinger. It uses quantum mechanics as a mathematical language to explain nature at the atomic scale. In quantum mechanics, quantum objects including neutrons, protons, quarks, and light particles such as photons can display both wave-like and particle-like properties that are considered as complementary. In contrast to macroscopic objects of classical physics, any quantum object can be in a superposition of many different states at the same time that enables for quantum parallelism. In particular, it can exhibit interference effects during the course of its unitary evolution, and can be entangled with other spatially separated quantum objects such that operations on one of them may cause non-local effects that are impossible to realize by means of classical physics.

It has been proven that quantum computing can simulate classical computing. However, the fundamental *raison d'être* of quantum computation is the fact that quantum physics appears to allow one to transgress the classical boundary between polynomial and exponential computations [25]. Though there is some evidence for that proposition, only very few practical applications of quantum computing and communication have been proposed so far including quantum cryptography [5].

Quantum computing devices have been physically implemented since the late 1990's by use of, for example, nuclear magnetic resonance [43], and solid state technologies such as that of neighbouring quantum dots implanted in regions of silicon based semiconductor on the nanometer scale [27]. As things are now, they work for up to several tens of qubits. Whether large-scale fault-tolerant and networked quantum computers with millions of qubits will ever be built remains purely speculative at this point. Though, rapid progress and current trends in nanoscale molecular engineering, as well as quantum computing research carried out at research labs across the globe could make it happen to let us see increasingly sophisticated quantum computing devices in the era 2020 to 2050. This leads, in particular, to the question how intelligent software agents [46, 45] could take most advantage of the potential of quantum computation and communication, once practical quantum computers are available. Will quantum computational agents be able to outperform their counterparts on classical von-Neumann computers? What kinds of architectures and programming languages are required to implement them? Does the adoption of quantum computational and communication means affect the autonomy of individual and systems of agents? This chapter provides some first thoughts on, and preliminary answers to these questions based on known fundamental and recent results of research in quantum computing and communication. It is intended to help bridging the gap between the agent and quantum research community for interdisciplinary research on quantum computational intelligent agents.

In sections 2 and 3, we briefly introduce the reader to the basics of quantum information, computation and communication in terms of quantum mechanics. For more comprehensive and in-depth introductions to quantum physics, and quantum computation we refer the interested reader to, for example, [12], respectively, [33, 23, 42, 1]. [17] provides a well-readable discussion of alternative interpretations of quantum mechanics. Readers who are familiar with the subjects can skip these sections. In section 4, we outline an architecture for a hybrid quantum computer, and propose a conceptual architecture and examples of quantum computational agents for such computers in section 5. Issues of quantum computational agent autonomy are discussed in section 6.

## 2 Quantum Information

Quantum computation is the extension of classical computation to the processing of quantum information based on physical two-state quantum systems such as

photons, electrons, atoms, or molecules. The unit of quantum information is the quantum bit, the analogous concept of the bit in classical computation.

## 2.1 Quantum Bit

Any physical two-state quantum system such as a polarized photon can be used to realize a single *quantum bit* (qubit). According to the postulates of quantum mechanics, the state space of a qubit  $\psi$  is the 2-dimensional complex Hilbert space  $H_2 = \mathbb{C}^2$  with given orthonormal computational basis in which the state  $|\psi\rangle$  is observed or measured<sup>1</sup>. The standard basis of qubit state spaces is  $\{|0\rangle, |1\rangle\}$  with coordinate representation  $|0\rangle = (1, 0)^t$ , and  $|1\rangle = (0, 1)^t$ . Any *quantum state*  $|\psi\rangle$  of a qubit  $\psi$  is a coherent *superposition* of its basis states

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle \quad (1)$$

where the probability *amplitudes*  $\alpha_1, \alpha_2 \in \mathbb{C}$  satisfy the normalization requirement  $|\alpha_0|^2 + |\alpha_1|^2 = 1$  for classical probabilities  $p(|\psi\rangle = |0\rangle) \equiv p(0) = |\alpha_0|^2$ , respectively,  $p(|\psi\rangle = |1\rangle) \equiv p(1) = |\alpha_1|^2$  of the occurrence of alternative basis states<sup>2</sup>. The decision of the physical quantum system realizing the qubit on one of the alternatives is made non-deterministically upon irreversible measurement in the standard basis. It reduces the superposed qubit state to the bit states '0' and '1' in classical computing. This transition from the quantum to the observable macroscopic world is called *quantum decoherence*.

## 2.2 Quantum Bit Register

A *n-qubit register*  $\psi = \psi_1 \dots \psi_n$  of  $n$  qubits  $\psi_i, i \in \{1, \dots, n\}$  is an ordered, composite  $n$ -quantum system. According to quantum mechanics, its state space is the  $n$ -

folded *tensor (Kronecker) product*  $H_2^{\otimes n} = \overbrace{H_2 \otimes \dots \otimes H_2}^n$  of the (inner product) state spaces  $H_2$  of its  $n$  component qubits. Each of the  $2^n$   $n$ -qubit basis states  $|x_i\rangle, x_i \in \{0, 1\}^n$  of the register can be viewed as the binary representation of a number  $k$  between 0 and  $2^n - 1$ . Any *composite state* of a  $n$ -qubit register is in a superposition of its basis states

$$|\psi\rangle = |\psi_1 \psi_2 \dots \psi_n\rangle = \sum_{k=0}^{2^n-1} \alpha_k |k\rangle, \quad \sum_{k=0}^{2^n-1} |\alpha_k|^2 = 1 \quad (2)$$

<sup>1</sup> Paul Dirac's bra-ket notation  $\langle \psi | = (\alpha_1, \dots, \alpha_k)^T$  (bra) and  $|\psi\rangle = (\alpha_1^*, \dots, \alpha_k^*)$  (ket) with complex conjugates  $\alpha_i^*, i \in \{1, \dots, k\}$  is the standard notation for system states in quantum mechanics. The inner product of quantum state vectors in  $H_k$  is defined as  $\langle \psi_1 | \psi_2 \rangle = (\alpha_i^*)_{i \in \{1, \dots, k\}} \otimes (\beta_i)_{i \in \{1, \dots, k\}} = \sum_{i=1}^k \alpha_i^* \beta_i$ . The orthonormal basis of  $H_k$  can be chosen freely, but if fixed refers to one physical observable of the quantum system  $\psi$  such as position, momentum, velocity, or spin orientation of a polarized photon, that can take  $k$  values.

<sup>2</sup> In contrast to physical probabilistic systems, a quantum system can destructively interfere with itself which can be described by negative amplitude values.

As the state of any  $n$ - and  $m$ -qubit register can be described by  $2^n$ , respectively,  $2^m$  amplitudes, any distribution on the joint state space of the  $n + m$ -qubit register takes  $2^{n+m}$  amplitudes. Hence, in contrast to classical memory, quantum memory increases exponentially in the size of the number of qubits stored in a quantum register. It can be doubled by adding just one qubit.

### 2.3 Measurement of Qubits

Measurement of a  $n$ -qubit register  $\psi$  in the standard basis yields a  $n$ -bit post-measurement quantum state  $|\psi_k\rangle$  with probability  $|\alpha_k|^2$ . Measurement of the first  $z < n$  qubits corresponds to the orthogonal measurement with  $2^z$  projectors  $M_i = |i\rangle\langle i| \otimes I_{2^{n-z}}$ ,  $i \in \{0, 1\}^z$  which collapses it into a probabilistic classical bit vector, yielding a single state randomly selected from the exponential set of possible states<sup>3</sup>. Measurement of the individual qubit  $\psi_m$  of a  $n$ -qubit register  $\psi = \psi_1 \dots \psi_m \dots \psi_n$ ,  $n \geq m$  in compound state  $|\psi\rangle = \sum_{i=0}^{2^n-1} c_i |i_1 \dots i_n\rangle$  with measurement operator  $M_m$  will give the classical outcome  $x_m \in \{0, 1\}$  with probability  $p(x_m) = \sum_{i_1 \dots i_n} |c_{i_1 \dots i_{m-1} x_m i_{m+1} \dots i_n}|^2 = \langle \psi | M_m^* M_m | \psi \rangle$ , and post-measurement state is

$$|\psi\rangle' = \frac{1}{\sqrt{p(x_m)}} \sum_{i_1 \dots i_{m-1} i_{m+1} \dots i_n} c_{i_1 \dots i_{m-1} x_m i_{m+1} \dots i_n} |i_1 \dots i_{m-1} x_m i_{m+1} \dots i_n\rangle$$

where  $c_{i_1 \dots i_{m-1} x_m i_{m+1} \dots i_n}$  denote the amplitudes of those  $2^n$  alternatives for which  $x$  could be observed as state value of the  $m$ -th qubit of  $\psi$  upon measurement. In general, the post-measurement quantum state  $|\psi_k\rangle'$  of  $|\psi_k\rangle$  is  $\frac{M_m |\psi_k\rangle}{\sqrt{\langle \psi | M_m^* M_m | \psi \rangle}}$ .

### 2.4 Unitary Evolution of Quantum States

According to the postulates of quantum mechanics, the time evolution of any  $n$ -qubit register,  $n \geq 1$ , is determined by any linear, unitary<sup>4</sup> operator  $U$  in the  $2^n$ -dimensional Hilbert space  $H_2^{\otimes n}$ . The size of the unitary matrix of a  $n$ -qubit operator is  $2^n \times 2^n$ , hence exponential in the physical size of the system. Since any unitary transformation  $U$  has an inverse  $U^{-1} = U^*$ , any non-measuring quantum operation is reversible, its action can always be undone. Measurement of a qubit  $\psi$  is an irreversible operation since we cannot reconstruct its state  $|\psi\rangle$  from the observed classical state after measurement.

<sup>3</sup> According to the *standard interpretation of quantum mechanics* it is meaningfully to attribute a definite state to a qubit only *after* a precisely defined measurement has been made. Due to Heisenberg's *uncertainty principle* complementary observables such as position and momentum cannot be exactly determined at the same time.

<sup>4</sup> *Unitarity* preserves the inner product ( $\langle \phi | U^* U | \psi \rangle$ ), similar to a rotation of the Hilbert space that preserves angles between state vectors during computation.

## 2.5 Entangled Qubits

Entangled  $n$ -qubit register states cannot be described as a tensor product of its component qubit states. Central to entanglement is the fact that measuring one of the entangled qubits can affect the probability amplitudes of the other entangled qubits no matter how far they are spatially separated. Such kind of non-local or holistic correlations between qubits captures the essence of the *non-locality principle of quantum mechanics* which has been experimentally verified by John Bell in 1964 [3] but is impossible to realize in classical physics.

### Example 2.1: *Entangled qubits*

Prominent examples of entangled 2-qubit are the *Bell states*

$$\begin{aligned} |\psi^+\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), |\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \\ |\psi^-\rangle &= \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle), |\phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \end{aligned}$$

The Bell state  $|\phi^+\rangle = (\frac{1}{\sqrt{2}}, 0, 0, \frac{1}{\sqrt{2}})$  is not decomposable. Otherwise we could find amplitudes of a 2-qubit product state  $(\alpha_{11}|0\rangle + \alpha_{12}|1\rangle)(\alpha_{21}|0\rangle + \alpha_{22}|1\rangle) = \alpha_{11}\alpha_{21}|00\rangle + \alpha_{11}\alpha_{22}|01\rangle + \alpha_{12}\alpha_{21}|10\rangle + \alpha_{12}\alpha_{22}|11\rangle$  such that  $\alpha_{11}\alpha_{21} = \frac{1}{\sqrt{2}}$ ,  $\alpha_{11}\alpha_{22} = 0$ ,  $\alpha_{12}\alpha_{21} = 0$  and  $\alpha_{12}\alpha_{22} = \frac{1}{\sqrt{2}}$  which is impossible. We cannot reconstruct the total state of the register from the measurement outcomes of its component qubits.  $|\phi^+\rangle$  can be produced by applying the conditioned-not 2-qubit operator  $M_{cnot} = ((1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 0, 1), (0, 0, 1, 0))$  to the separable register state  $|\psi_1\psi_2\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |10\rangle)$ .

Suppose we have measured 0 as definite state value of the second qubit in state  $|\phi^+\rangle \equiv a|00\rangle + b|01\rangle + c|10\rangle + d|11\rangle \equiv (a, b, c, d)$  with amplitudes normalized to 1. The corresponding measurement operator is the self-adjoint, non-unitary projector  $M_{2:0} = ((1, 0, 0, 0), (0, 0, 0, 0), (0, 1, 0, 0), (0, 0, 0, 0))$ , which yields the outcome 0 or 1 with equal probability, for example,  $p(0) = \langle \phi^+ | M_{2:0}^* M_{2:0} | \phi^+ \rangle = (\frac{1}{\sqrt{2}}, 0, 0, \frac{1}{\sqrt{2}})(\frac{1}{\sqrt{2}}, 0, 0, 0)^t = \frac{1}{2}$ , and the post-measurement state  $|\phi^+\rangle' = \frac{M_{2:0}|\phi^+\rangle}{\sqrt{\langle \phi^+ | M_{2:0} | \phi^+ \rangle}} = \frac{(\frac{1}{\sqrt{2}}, 0, 0, 0)}{\sqrt{1/2}} = \sqrt{2}(\frac{1}{\sqrt{2}}, 0, 0, 0) = (1, 0, 0, 0) = |00\rangle \neq a|00\rangle + c|10\rangle$ .

That means, measurement of the second qubit caused also the entangled first qubit to instantaneously assume a classical state without having operated on it.

o

Pairs of entangled qubits are called *EPR pairs* with reference to the associated Einstein-Podolsky-Rosen (EPR) thought experiment [20]. The *non-local effect of instantaneous state changes* between spatially separated but entangled quantum states upon measurement belongs to the most controversial issue and debated phenomenon of quantum physics, and caused interesting attempts of developing a quantum theory of the human mind and brain [39, 38]. Entanglement links information across qubits, but does not create more of it [22], nor does it allow to communicate any classical information faster than light.

Entangled qubits can be physically created either by having an EPR pair of entangled particles emerge from a common source, or by allowing direct interaction between the particles, or by projecting the state of two particles each

from different EPR pairs onto an entangled state without any interaction between them (*entanglement swapping*) [12]. Entanglement of qubits is considered as one essential feature of, and resource for quantum computation and quantum communication [25, 11].

### 3 Quantum Computation and Communication

The quantum Turing machine model [37], and the quantum circuit model [18] are equivalent models of quantum computation. In this paper, we adopt the latter model.

#### 3.1 Quantum Logic Gates and Circuits

A  $n$ -qubit gate is a unitary mapping in  $H_2^{\otimes n}$  which operates on a fixed number of qubits (independent of  $n$ ) given  $n$  input qubits. Most quantum algorithms to date are described through a *quantum circuit* that is represented as a finite sequence of concatenated quantum gates. Basic quantum gates are the 1-qubit *Hadamard* (H) and *Pauli* (X, Y, Z) gates, and the 2-qubit XOR, called *conditioned not* (CNOT), gate. These operators are defined by unitary matrices as follows

$$M_H = \frac{1}{\sqrt{2}}((1, 1), (1, -1))$$

$$M_{CNOT} = ((1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 0, 1), (0, 0, 1, 0))$$

$$M_X = ((0, 1), (1, 0)), M_Z = ((1, 0), (0, -1)), M_Y = ((0, -i), (i, 0))$$

The Hadamard gate creates a superposed qubit state for standard basis states, demonstrates *destructive quantum interference* if applied to superposed quantum states ( $M_H(\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)) = |0\rangle$ ), and can be physically realized, for example, by a 50/50-beamsplitter in a Mach-Zehnder interferometer [12]. The CNOT gate flips the second (target) qubit if and only if the first (control) qubit is in state  $|1\rangle$ . The quantum circuit consisting of a Hadamard gate followed by a CNOT gate creates an entangled *Bell state* for each computational basis state. The X gate is analogous to the classical bit-flip NOT gate, and the Z gate flips the phase (amplitude sign) of the basis state  $|1\rangle$  in superposition. Other common basic qubit gates include the NOP, S, and T gates for quantum operations of identity, phase rotation by  $\pi/4$ , respectively  $\pi/8$ . The set {H, X, Z, CNOT, T} is universal [33].

#### 3.2 Quantum Vs. Classical Computation

The constraint of unitary evolution of qubit states yields a generalization of the restriction of classical (Turing machine or logic circuit based) models of computation to unitary, hence reversible computation [4]. It has been shown that each classical algorithm computing a function  $f$  can be converted into an equivalent quantum operator  $U_f$  with the same order of efficiency [49, 1], which

means that quantum systems can imitate all classical computations. However, the fundamental *raison d'être* of quantum computation is the expectation that quantum physics allows one to do even better than that.

The linearity of quantum mechanics gives rise to *quantum parallelism* that allows a quantum computer to simultaneously evaluate a given function  $f(x)$  for all inputs  $x$  by applying its unitary transformation  $U_f : |x\rangle |0\rangle \mapsto |x\rangle |0 \oplus f(x)\rangle = |x\rangle |f(x)\rangle$  to a suitable superposition of these inputs such that

$$U_f \left( \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle |0\rangle \right) = \frac{1}{\sqrt{2^n}} \sum_{x \in \{0,1\}^n} |x\rangle |f(x)\rangle \quad (3)$$

Though this provides, in essence, not more than classical randomization, if combined with the effects of *quantum interference* such as in the Deutsch-Josza algorithm ([33], p.36) and/or quantum entanglement [11] it becomes a fundamental feature of many quantum algorithms for speeding-up computations. The basic idea is to compute some global property of  $f$  by just one evaluation based on a combination of interfered alternative values of  $f$ , whereas classical probabilistic computers only can evaluate different but forever mutually excluding alternative values of  $f$  with equal probability.

In general, *quantum algorithms* appear to be best at problems that rely on promises or *oracle* settings, hence use some hidden structure in a problem to find an answer that can be easily verified through, for example, means of amplitude amplification. Prominent examples include the quantum search developed by Grover (1996) for searching sets of  $n$  unordered data items [21], and the quantum prime factorization of  $n$ -bit integers developed by Shor (1994) [41] with complexity of  $O(\sqrt{n})$ , respectively,  $O(n^3)$  time, which is a quadratic and exponential speed-up compared to the corresponding classical case. It is not known to date whether quantum computers are in general more powerful than their classical counterparts<sup>5</sup>. However, it is widely believed that the existence of an efficient solution of the NP-hard problem of integer prime factoring using the quantum computation model [41], as well as the quadratically speed up of classical solutions of some NP-complete problems such as the Hamiltonian cycle problem by quantum search ([33], p.264), provides evidence in favor of this proposition.

### 3.3 Quantum Communication Models

In this paper, we consider the following models of quantum based communication between two quantum computational agents  $A$  and  $B$ .

---

<sup>5</sup> In terms of the computational complexity classes  $P$ ,  $BPP$ ,  $NP$ , and  $PSPACE$  with  $P \subseteq NP \subseteq PSPACE$ , it is known that  $P \subseteq QP$ ,  $BPP \subseteq BQP$ , and  $BQP \subseteq PSPACE$  [10].  $QP$  and  $BQP$  denote the class of computational problems that can be solved efficiently in polynomial time with success probability of 1 (exact), or at least 2/3 (bounded probability of error), respectively, on uniformly polynomial quantum circuits.



1. **QCOMM-1.** Agents  $A$  and  $B$  share entangled qubits and use a classical channel to communicate.
2. **QCOMM-2.** Agents  $A$  and  $B$  share entangled qubits and use a quantum channel to communicate.
3. **QCOMM-3.** Agents  $A$  and  $B$  share no entangled qubits and use a quantum channel to communicate.

*QCOMM-1: Quantum teleportation of  $n$  qubits with  $2n$  bits.* The standard process of teleporting a qubit  $\phi$  from agent  $A$  to agent  $B$  based on a shared EPR pair  $\psi_1\psi_2$  and classical channel works as follows [7]. Suppose agent  $A$  ( $B$ ) keeps qubit  $\psi_1$  ( $\psi_2$ ).  $A$  entangles  $\phi$  with  $\psi_1$  by applying the CNOT, and the Hadamard gate to the 2-qubit register  $[\phi\psi_1]$  into one of four Bell states  $|\phi\psi_1\rangle$ . It then sends the measurement outcome (00, 10, 01, or 11) to agent  $B$  through a classical communication channel at the cost of two classical bits. Only upon receipt of  $A$ 's 2-bit notification message, agent  $B$  is able to create  $|\phi\rangle$  by applying the identity or Pauli operator gates to its qubit  $\psi_2$  depending on the content of the message (00: I, 01: X; 10: Z; 11: XZ) <sup>6</sup>.

*QCOMM-2: Quantum dense coding of  $n$ -bit strings in  $n/2$  qubits.* Agent  $A$  dense codes each of consecutive pairs of bits  $b_1b_2$  at the cost of one qubit as follows [8]. Suppose agent  $A$  ( $B$ ) keeps qubit  $\psi_1$  ( $\psi_2$ ) of shared EPR pair in entangled Bell state  $|\psi\rangle = |\psi_1\psi_2\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ . According to prior coding agreement with  $B$ , agent  $A$  applies the identity or Pauli operators to its qubit depending on the 2-bit message to be communicated (for example, 00:  $I \otimes I|\psi\rangle$ , 01:  $X \otimes I|\psi\rangle$ , 11:  $Z \otimes I|\psi\rangle$ , 10:  $(XZ)^t \otimes I|\psi\rangle$ ) which results in one of four Bell states  $|\psi\rangle'$  and physically transmits the qubit  $\psi_1$  to  $B$ . Upon receipt of  $\psi_1$ , agent  $B$  performs  $M_{CNOT}|\psi\rangle'$  yielding separable state  $|\gamma_0\gamma_1\rangle$ , applies the Hadamard operation to the first qubit  $M_H|\gamma_0\rangle = |\delta_0\rangle$  and decodes the classical 2-bit message depending on measured states of  $\delta_0\gamma_1$  (e.g.,  $\delta_0\gamma_1 = 00$ : 00, 01: 01, 11: 10, 10: 11).

A fundamental result in quantum information theory by Holevo (1973) [24] implies that by sending  $n$  qubits one cannot convey more than  $n$  classical bits of information. However, for every classical (probabilistic) communication problem [48] where agents exchange classical bits according to their individual inputs and then decide on an answer which must be correct (with some probability), quantum protocols where agents exchange qubits of communication are at least as powerful [31].

<sup>6</sup> Due to (Bell state) measurement of  $|\phi\psi_1\rangle$  agent  $A$  lost the original state  $|\phi\rangle$  to be communicated. However, since  $\psi_1$  and  $\psi_2$  were entangled, this measurement instantaneously affected the state of  $B$ 's qubit  $\psi_2$  (cf. Ex. 2.1) such that  $B$  can recover  $|\phi\rangle$  from  $|\psi_2\rangle$ .

## 4 Quantum Computers

All known quantum algorithms require the determinism and reliability of classical control for the execution of suitable quantum circuits consisting of a finite sequence of quantum gates and measurement operations. Figure 1 shows a master-slave architecture of a *hybrid quantum computer* based on proposals in [35] and [9] in which classical signals and processing of a classical machine (CM) are used to control the timing and sequence of quantum operations carried out in a *quantum machine* (QM).

The QM consists of *quantum memory*, *quantum processing unit* (QPU) with error correction, *quantum bus*, and *quantum device controller* (QDC) with interface to the classical machine (CM). The classical machine consists of a CPU for high-level dynamic control and scheduling of the QM components, and memory that can be addressed by both classical and quantum addressing schemes (e.g., [9] p.20, [33] p.268). Quantum memory can be implemented as a lattice of static physical qubits, which state is factorized in tensor states over its nodes<sup>7</sup>. Qubit states can be transported within the QM along point-to-point quantum wires either via teleportation (cf. section 3.3), or chained quantum swapping and repeaters [36]<sup>8</sup>.

A few *quantum programming languages* (QPL) for hybrid quantum computers exist, such as the procedural QCL [34], and QL [9], and the functional qpl [40]. A QPL program contains high-level primitives for logical quantum operations, interleaved with classical work-flow statements. The QPL primitives are compiled by the CPU into low-level instructions for qubit operators that are passed to and then translated by the QDC to physical qubit (register) operations which are executed by the QPU. The QPU performs scheduled sequences of measurement and basic qubit operations from a universal set of 1- and 2-qubit quantum gates (cf. section 3.1) with error correction<sup>9</sup> to minimize quantum decoherence caused by imperfect control over qubit operations, measurement errors, number of entangled qubits, and the physical limits of the quantum systems such as nuclear spins used to realize qubits [19]. The QM returns only the results of quantum measurements to the CM.

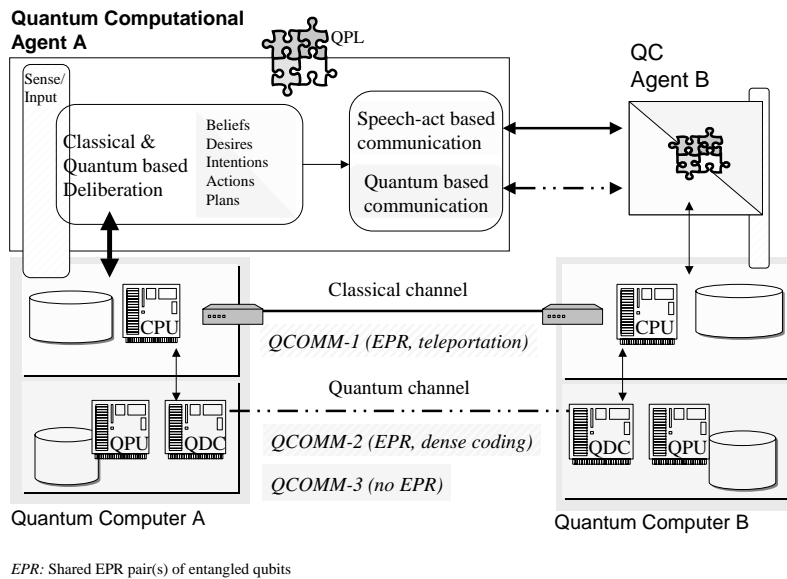
---

<sup>7</sup> According to the *no-cloning theorem* of quantum computing [47], a qubit state cannot be perfectly copied unless it is known upon measurement. Thus, no backup copies of quantum data can be created in due course of quantum computation.

<sup>8</sup> In short *quantum wires* a qubit state can be progressively swapped between pairs of qubits in a line, where each qubit is represented, for example, by the nuclear spin of a phosphorus atom implanted in silicon (quantum dot). Each swap operation along this line of atoms is realized by three back-to-back CNOT gates.

<sup>9</sup> According to the *threshold theorem* of quantum computing [28, 2], scalable quantum computers with faulty components can be built by using quantum error correction codes as long as the probability of error of each quantum operation is less than  $10^{-4}$ .





**Fig. 2.** Conceptual scheme of a quantum computational agent.

*Local quantum based search.* Suppose a QCIA has to search its local unstructured classical database  $LDB$  with  $N = 2^n$   $l$ -bit data entries  $d_x$  each of which is indexed by value  $x = 0 \dots N - 1$  for given  $l$ -bit input  $s$  and search oracle  $O$  with  $1 \leq M \leq N$  solutions. The oracle is implemented by an appropriate quantum circuit  $U_f$  that checks whether the input is a solution to the search problem ( $f(x) = 1$  if  $d_x = s$ , else  $f(x) = 0$ ). No further structure to the problem is given. Any classical search would take an average of  $O(N/M)$  oracle calls to find a solution. Using Grover's quantum search algorithm [21] the QCIA can do the same in  $O(\sqrt{N/M})$  time. The basic idea is that (a) the search is performed on a  $\log N$ -qubit index register  $|x\rangle$  which state is in superposition of all  $N = 2^n$  index values  $x^{10}$ , and (b) the oracle  $O$  marks the  $M$  solutions ( $|x\rangle \rightarrow (-1)^{f(x)}|x\rangle$  with  $f(x) = 1$  if  $d_x = s$ , 0 else) which are amplified to increase the probability that they will be found upon measurement of the index register after  $O(\sqrt{N/M})$  iterations. Type and cost of each oracle call (matching operation  $U_f$ ) depends on the application. Implementation of the search uses  $n$ -qubit index,  $l$ -qubit data and input, and 1-qubit oracle register of the QPU. Like in the classical search, we need a quantum addressing scheme ([33] p.268) with  $O(\log N)$  per operation to access, load, and restore indexed data  $d_x$  to the data register, and recreate respectively measured index states  $|x\rangle$  for further processing.

<sup>10</sup> The initial superposed index state  $|x\rangle$  is created by  $n$ -folded Hadamard operation ( $H^{\otimes n}|0\rangle = \frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} |i\rangle$ ).

*Local quantum based matchmaking* is a special case of local quantum search for binary coded service descriptions. The type of service matching depends on the implemented search oracle. We assume that both service requests and service ads are encoded in the same way to allow for meaningful comparison by a *quantum computational matchmaker agent*. Componentwise quantum search with bounded rather than exact success probability for partial matching could be used for syntactic but not semantic service matching such as in LARKS [44].

## 5.2 QC Multi-Agent Systems

A *quantum computational multi-agent system* (QCMAS) is a multi-agent system that consists of both classical and quantum computing agents which can interact to jointly accomplish their goals. A *pure QCMAS* consists of QC agents only. QCMAS which members cannot interact with each other using a quantum communication model (cf. section 3.3) are called *type-I QCMAS*, and *type-II QCMAS* otherwise. QC agents of type-I or type-II QCMAS are called *type-I QC agents*, respectively, *type-II QC agents*.

Inter-agent communication between type-I QC agents bases on the use of classical channels without sharing any EPR pairs. None of the quantum communication models is applicable. As a consequence, quantum computation is performed locally at each individual agent. In addition, type-II QC agents can use quantum communication models which cannot be simulated in any type-I QCMAS. It is assumed that type-II QC agents share a sufficient number of EPR pairs, and have prior knowledge on used quantum coding operations for this purpose. Any QC agent communicates appropriate speech-act based messages via classical channels to synchronize its actions, if required. Messages related to quantum communication between type-II QC agents concern, for example, the notification in quantum teleportation (QCOMM-1), the prior agreement on the order of operations in quantum dense coding (QCOMM-2), and the semantics of qubits (QCOMM-3).

*What are the main benefits of QCMAS?* In certain cases, QCMAS can be computationally more powerful than any MAS by means of properly designed and integrated QC agents. The main challenge is the development of application-specific quantum algorithms that can do better than any classical algorithm.

Quantum-based communication between type-II QC agents is inherently secure. Standard quantum teleportation (QCOMM-1) ensures data integrity, since it is impossible to deduce the original qubit state from eavesdropped 2-bit notification messages of the sender without possessing the respective entangled qubit of the receiver. Quantum dense coding (QCOMM-2) is secure, since any quantum operation on the physically transmitted qubit in any of the four Bell states takes the same value. The physical transmission of qubits via a quantum channel (QCOMM-3) is secure due to the no-cloning theorem of quantum mechanics, a fact that is also used in quantum key distribution [5]. Any attempt of eavesdropping will reckognizably interfere with the physical quantum states of transmitted qubits.

Finally, certain communication problems [48], that is the joint computation of some boolean function  $f$  minimizing the number of qubits to communicate for this purpose, can be solved more efficiently by type-II QC agents. In general, it has been proven in [15] that the gap between bounded-error (zero-error) classical and (exact) quantum communication complexity is near quadratic (exponential), and that each quantum communication model is at least as powerful than a classical one for every communication problem on  $n$ -bit inputs [31]. More interesting, they can do even better for certain communication problems such as the computation of inner product, equality, and disjointness of boolean functions  $f(x), g(x)$  according to individual  $n$ -bit inputs  $x \in \{0, 1\}^n$ . Quantum based solutions to latter problems can be applied to quantum based collaborative search, and matchmaking [30], with respective quadratic or exponential reduction of communication complexity.

### 5.3 Examples of Type-I and Type-II QCMAS

*Quantum based collaborative search in type-I QCMAS.* Upon receipt of a multicasted  $l$ -bit request  $s$  from QCIA  $A_1$ , each agent  $A_j, j = 2..n$  locally computes  $M_j \geq 1$  solutions to the given search problem  $LQS(s, O, LDB_j)$  in  $O(\sqrt{N_j/M_j})$  time, instead of  $O(N)$  in the classical case, and returns the found data items to  $A_1$ . Due to non-quantum based interaction, both requests and replies have to be binary coded for transmission via classical channel, and binary requests are directly quantum coded prior to quantum search (cf. section 5.1).

*Quantum based collaborative search in type-II QCMAS.* Suppose two QCIA  $A_1, A_2$  want to figure out whether a  $n$ -bit request  $s$  matches with data item  $s' \in LDB_2 (N = 1)$  with the promise that their Hamming distance is  $h(s, s') = 0$  else  $n/2$ . In this case, it suffices to solve the corresponding equality problem with  $O(\log n)$  qubits of communication, instead of  $O(n)$  in the classical case [15]. Basic idea is that  $A_1$  prepares its  $n$ -bit  $s$  in a superposition of  $\log n + 1$  qubits such that  $A_2$  can test, upon receipt of  $s$ , whether  $s_i \oplus s'_i = 0, i = 1..n$  by applying the known oracle-based Deutsch-Josza quantum algorithm ([33], p.34) to  $|s\rangle = |o \oplus s \oplus s'\rangle$ , followed by Hadamard operations ( $H^{\otimes(\log n + 1)}$ ), and measurement of the final state yields the desired result.

*QC matchmaking in type-I and type-II QCMAS.* As in the classical case, quantum based service matchmaking can be directly performed by pairs of QC agents in both types of QCMAS. In fact, it is a special case of the collaborative search scenario where two QC agents can both advertise and request a set of  $N (N')$   $l$ -bit services from each other. For example, QC service agents  $A$  of a type-II QCMAS can physically send a set of (QCOMM-2: dense coded)  $n$ -bit service request each of size  $n/2$  qubits to a QC matchmaker  $A^*$  via a quantum channel (QCOMM-3). In cases where only classical channels are available (QCOMM-1),  $A$  can teleport the qubit request to  $A^*$  at the cost of  $2n$  bits. In any case, upon receipt of the request,  $A^*$  quantum searches its classical database of  $N$  service

ads, and returns those that matches it according to the given search ("matching") oracle. Using quantum search, the disjointness of sets of quantum coded service descriptions interpreted as ads and/or requests can be decided with just  $\log N + 1$  qubits of communication [15], instead of at least  $N$  bits in the classical randomized setting [26].

## 6 Autonomy of QC Agents

Following the classification of different types of agent autonomy in [16], we define a QC agent  $A$  autonomous from QC agent  $B$  for given autonomy object  $o$  in the context  $c$ , if, in  $c$ , its behaviour regarding  $o$  is not imposed by  $B$ . The ability of an individual QC agent in type-I QCMAS to exhibit autonomous behaviour is not affected by its local quantum computation, since non-local effects are restricted to local quantum machine components. Hence, the self-autonomy of individual type-I QC agents in terms of the ability to autonomously reason about sets of goals, plans, and motivations for decision-making remains intact. That is independent from the fact that the computational complexity of deliberative actions could possibly be reduced by, for example, quantum searching of complex plan libraries. Regarding user autonomy, any external physical interaction with the quantum machine by the user will cause massive quantum decoherence which puts the success of any quantum computational process and associated accomplishment of tasks and goals of individual type-I QC agents at risk.

A type-II QC agent shall be able to adjust its behaviour to the current quantum computing context of the overall task or goal to accomplish. It can freely decide on whether and with which agents to share a sufficient number of EPR pairs, or to make prior coding agreements according to the used quantum communication model. However, both its adjustable interaction and computational autonomy, turn out to be limited to the extent of entanglement based joint computation and communication with other type-II QC agents. Any type-II QC agent can change the state of non-local qubits that are entangled with its own qubits by local Bell state measurements. This way, if malevolent, it can misuse its holistic correlations with other type-II QC agents to corrupt their computations by manipulating their respective entangled quantum data. Even worse, there is no way for these agents to avoid such kind of influence.

For example, suppose that agents  $A$  and  $B$  share EPR pairs to interact using quantum teleportation (QCOMM-1). Since the change of  $B$ 's entangled qubits caused by  $A$ 's local Bell state measurements is instantaneous,  $B$  cannot avoid it at all.  $B$  does not even know that such changes occurred until it receives  $A$ 's 2-bit notification messages (cf. section 3.3).  $B$  is not able to clone its entangled qubits, and measuring their state prior to  $A$ 's notification would let communication fail completely. The same situation occurs when entanglement swapping is used to teleport qubit states along a path of correlated QC agents in a type-II QCMAS; in fact, it holds for any kind of entanglement based computation in general.

To summarize, the use of entanglement as a resource for computation and communication requires type-II QC agents to strictly trust each other. The ability of individual type-II QC agents to influence other type-II QC agents is inherently coupled with the risk of being influenced in turn by exactly the same agents in the same way. Though, for an individual agent the degree of its influence can be quantified based on the number, and the frequency of respective usage of its entangled quantum data.

## 7 Conclusions

In essence, quantum computational agents and multi-agent systems are feasible to implement on hybrid quantum computers, and can be used to solve certain problems in practical applications such as information search and service matchmaking more efficiently than with classically computing agents. Type-II QC agents can take most computational advantages of quantum computing and communication, but at the cost of limited self-autonomy, due to non-local effects of quantum entanglement. Quantum-based communication between type-II agents is inherently secure.

Ongoing and future research on QC agents and multi-agent systems focuses on appropriate integration architectures for QCMAS of both types, type-II QC information and matchmaker agents, as well as potential new applications such as secure quantum based distributed constraint satisfaction, and qualitative measures and patterns of quantum computational autonomy in type-II QCMAS.

## References

1. D. Aharonov. Quantum Computation. LANL Archive quant-ph/981203, 1998.
2. D. Aharonov, M. Ben-Or. Polynomial Simulations of Decohered Quantum Computers. Proc. 37th Ann. Symp. Foundations of Computer Science (FOCS), 1996.
3. J.S. Bell. Speakable and Unspeakable in Quantum Mechanics. Cambridge University Press, 1987.
4. C.H. Bennett. Time/Space Trade-offs for Reversible Computation. SIAM Journal of Computing, 18(4):766-776, 1989.
5. C.H. Bennett, G. Brassard. Quantum Cryptography: Public Key Distribution and Coin Tossing. Proc. IEEE Intl. Conference on Computers, Systems, and Signal Processing, pp 175-179, 1984.
6. C.H. Bennett, D.P. DiVincenzo. Quantum Information and Computation. Nature, 404(6775):247-254, 2000.
7. C.H. Bennett, G. Brassard, C. Crepau, R. Josza, A. Peres, W.K. Wootters. Phys. Reviews Letters, 70, 1895, 1993.
8. C.H. Bennett, S.J. Wiesner. Communication via one- and two-particle operators on EPR states. Phys. Review Letters, 69(20), 1992.
9. S. Betteli, T. Calarco, L. Serafini. Toward an Architecture for Quantum Programming. LANL Archive cs.PL/0103009, March 2003.
10. E. Bernstein, U. Vazirani. Quantum complexity theory. SIAM Journal of Computing, 26(5):1411-1473, 1997.



11. E. Biham, G. Brassard, D. Kenigsberg, T. Mor. Quantum Computing Without Entanglement. LANL Archive quant-ph/0306182, 2003.
12. D. Bouwmeester, A. Ekert, A. Zeilinger. The Physics of Quantum Information. Springer, Heidelberg, 2000.
13. G. Brassard, I. Chuang, S. Lloyd, and C. Monroe. Quantum Computing. Proc. National Academy of Sciences, USA, vol. 95, p. 11032-11033, 1998.
14. G.K. Brennen, D.Song, C.J. Williams. A Quantum Computer Architecture using Nonlocal Interactions. LANL Archive quant-ph/0301012, 2003.
15. H. Buhrman, R. Cleve, A. Wigderson. Quantum vs. Classical Communication and Computation. Proc. 30th Ann. ACM Symp. Theory of Computing (STOC 98), 1998.
16. C. Carabelea, O. Boissier, A. Florea. Autonomy in Multi-agent Systems: A Classification Attempt. Proc. Intl. Autonomous Agents and Multiagent Systems Conference Workshop on Computational Autonomy, M Rovatso, M Nickles (eds.), Melbourne, Australia, 2003.
17. P.C.W. Davies, J.R. Brown (Eds.). The Ghost in the Atom. Cambridge University Press, Canto edition reprint, 2000.
18. D. Deutsch. Quantum Theory, the Church-Turing Principle, and the Universal Quantum Computer. Proc. Royal Society London A, 400:97, 1985
19. D.P. DiVincenzo. The Physical Implementation of Quantum Computation. LANL Archive quant-ph/0002077, 2000.
20. A. Einstein, B. Podolsky, N. Rosen. Can quantum mechanical description of physics be considered complete?. Phys. Review, 47:777-780, 1935.
21. L. Grover. A Fast Quantum Mechanical Algorithm for Database Search. Proc. 28th Annual ACM Symposium on Theory of Computation, ACM Press, NY USA, pp 212-219, 1996.
22. Gruska, Imai. Power, Puzzles and Properties of Entanglement. M. Margenster, Y. Rogozhin (eds.), Lecture Notes in Computer Science LNCS, 2055, Springer, 2001.
23. M. Hirvensalo. Quantum Computing. Natural Computing Series, Springer, 2001.
24. A.S. Holevo. Some estimates of the information transmitted by quantum communication channels. Problems of Information Transmission, 9:177-183, 1973.
25. R. Josza. Entanglement and Quantum Computation. Geometric Issues in the Foundations of Science, S. Huggett et al. (eds.), Oxford University Press, 1997.
26. B. Kalyanasundara, G. Schnitger. The probabilistic communication complexity of set intersection. SIAM Journal on Discrete Mathematics, 5(4), 1992.
27. B. Kane. A Silicon-Based Nuclear Spin Quantum Computer. Nature, 393, 1998
28. E. Knill, R. Laflamme, W.H. Zurek. Resilient Quantum Computation. Science, 279:342-345, 1998
29. M. Klusch. Information Agent Technology for the Internet: A Survey. Data and Knowledge Engineering, 36(3), Elsevier Science, 2001.
30. M. Klusch, K. Sycara. Brokering and Matchmaking for Coordination of Agent Societies: A Survey. Coordination of Internet Agents, A. Omicini et al. (eds), Springer, 2001.
31. I. Kremer. Quantum Communication. Master Thesis, Hebrew University, Jerusalem, Israel, 1995.
32. Los Alamos National Lab Archive, USA: <http://xxx.lanl.gov/archive/>
33. M.A. Nielsen, I.L. Chuang. Quantum Computation and Quantum Information. Cambridge University Press, Cambridge, UK, 2000.
34. B. Oemer. Quantum Programming in QCL. Master Thesis, Technical University of Vienna, Computer Science Department, Vienna, Austria, 2000.
35. M. Oskin, F.T. Chong, I.L. Chuang. A Practical Architecture for Reliable Quantum Computers. IEEE Computer, 35:79-87, January 2002.

36. M. Oskin, F.T. Chong, I.L. Chuang, J. Kubiawicz. Building Quantum Wires: The Long and the Short of it. Proc. 30th Intl Symposium on Computer Architecture (ISCA), 2003.
37. M. Ozawa. Quantum Turing Machines: Local Transitions, Preparation, Measurement, and Halting Problem. LANL Archive quant-ph/9809038, 1998.
38. R. Penrose. The Large, the Small, and the Human Mind. Cambridge University Press, 1997.
39. A. Pereira. The Quantum Mind/Classical Brain Problem. NeuroQuantology, 1:94-118, ISSN 1303-5150, Neuroscience & Quantum Physics, 2003.
40. P. Selinger. Towards a Quantum Programming Language. Mathematical Structures in Computer Science, 2003.
41. P. Shor. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. Proc. 35th Annual Symposium on Foundations of Computer Science, Los Alamitos, USA, 1994.
42. A. Steane. Quantum computing. LANL Archive quant-ph/9708022, 1997.
43. M. Steffen, L.M.K. Vandersypen, I.L. Chuang. Toward Quantum Computation: A Five-Qubit Quantum Processor. IEEE Micro, March/April, 2001.
44. K. Sycara, S. Widoff, M. Klusch, J. Lu. LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace. Autonomous Agents and Multi-Agent Systems, 5(2), 2002.
45. G. Weiss. Introduction to Multiagent Systems. MIT Press, 1999.
46. M. Wooldridge. An Introduction to Multiagent Systems. John Wiley & Sons, Chichester, UK, 2002.
47. W.K. Wootters, W.H. Zurek. A Single Quantum Cannot be Cloned. Nature, 299:802-803, 1982.
48. A.C-C. Yao. Quantum Circuit Complexity. Proc. 33rd Annual Symposium on Foundations of Computer Science (FOCS), pp. 352-361, 1993.
49. P. Zuliani. Logical Reversibility. IBM Journal of Res. & Devel., 45, 2001.

---

## Programming of Quantum Search Agents

M. Klusch, R. Schubotz: Programming and Simulation of Quantum Search Agents. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC), Montreal, Canada, IEEE Press, 2007.

# Programming and Simulation of Quantum Search Agents

Matthias Klusch and René Schubotz

**Abstract**—The extension of classical agents by the ability to perform quantum computation and communication provides an efficient and secure solution to applications such as information search and service matchmaking. In this paper, we propose a hybrid architecture for quantum computational agents, and demonstrate its principles by means of a simple type-I quantum search agent based on quantum pattern matching. Finally, we present preliminary results of the comparative evaluation of its implementation using different quantum programming languages and simulators.

## I. INTRODUCTION

Quantum computing technology based on quantum physics promises to eliminate some of the problems associated with the rapidly approaching ultimate limits to classical computers imposed by the fundamental law of thermodynamics. Quantum computing (QC) devices have been physically implemented since the late 1990's by use of, for example, nuclear magnetic resonance, and solid state technologies. Rapid progress and current trends in nanoscale molecular engineering, as well as quantum computing research carried out at research labs across the globe could make it happen to let us see increasingly sophisticated quantum computing devices in the era 2020 to 2050. The implied major research challenge of agent based computing in such environments is how to make the most of the potential of quantum computing and communication? We acknowledge that any answer to this question at the very moment will be, of course, highly speculative; though work in this direction already started such as in [7]. We build upon this work and focus more on its engineering aspects of by means of programming and simulation of a special kind of quantum computational agents, that is type-I quantum search agents. Key idea is to appropriately extend one prominent generic agent architecture, namely InteRRap, to the case of a type-I QC agent that is supposed to run on a hybrid quantum computer, and to show its feasibility by instantiating the respective QuantumInteRRap architecture for a programmed quantum pattern matching (QPM) based type-I quantum search agent. The remainder of the paper is structured as follows. After a brief introduction to quantum computation in section II, we comment on a recently proposed classification of QC agents and a quantum programming design flow in sections III-A, respectively, III-B. Based on this work, we present our quantum extension of the InteRRap architecture for QC agents in section III-C, and describe a slightly improved version of the quantum pattern matching algorithm of [9]

in section IV-A. The architecture and benchmarking results of the QPM based type-I quantum search agent simulated on different quantum simulators are presented in sections IV-B and V.

## II. QUANTUM COMPUTING IN VERY BRIEF

Quantum computation is built on the concept of the *qubit*. Any isolated physical 2-observable quantum system is appropriate to realize a single qubit. In mathematical terms, a qubit  $\psi$  is associated to its state space, a complex 2-dimensional Hilbert space  $H_2 = \text{span}\{|0\rangle, |1\rangle\}$  with orthonormal computational standard basis. Any quantum state  $|\psi\rangle$  of  $\psi$  is described by a *coherent superposition*,  $|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle$ ,  $|\alpha_0|^2 + |\alpha_1|^2 = 1$ . The state space  $H_2^{\otimes n}$  of a physical system composed of  $n$  single qubits  $\psi_i$  is the  $n$ -folded *tensor product* of the state spaces of its  $n$  constituting qubits  $H_2^{\otimes n} = \bigotimes^n H_2$ . Such systems can be regarded as *n-qubit register*  $\Psi$  with  $2^n$  computational basis states. If a state  $|\Psi\rangle$  of a  $n$ -qubit register can be written as a product of its constituting qubits in the form  $|\Psi\rangle = \bigotimes_i (\sum_j \alpha_{i,j} |j\rangle)$ , then  $|\Psi\rangle$  is called separable. Non-separable composite states are known as *entangled states*, allowing non-local effects of instantaneous state changes between spatially separated but entangled quantum states upon measurement. A *projective measurement* of  $\Psi$  is described by a set of pairwise orthogonal subspaces  $W_1, \dots, W_m$  satisfying  $H_2^{\otimes n} = \bigoplus_{k=1}^m W_k$  and results in  $j \in \{1, \dots, m\}$ . Let  $\{|\Phi_i^j\rangle\}$  define an orthonormal basis of subspace  $W_j$ , then the operator  $P_j = \sum_{i=1}^{\dim(W_j)} |\Phi_i^j\rangle \langle \Phi_i^j|$  projects  $\Psi$  on the subspace  $W_j$ . The probability of measuring  $j \in \{1, \dots, m\}$  is given by  $\langle \Psi | P_j | \Psi \rangle$ . Time evolution of  $\Psi$  is described by *unitary transformations* in its state space  $H_2^{\otimes n}$ . Any non-measuring quantum operation is *inherently reversible* since any unitary transformation  $U$  has an inverse. For a comprehensive introduction to quantum computation, we refer the reader to [12].

## III. AGENTS ON QUANTUM COMPUTERS

A *quantum computational agent* (QC agent) [7] is an intelligent software agent that is able to perform both classical and quantum computing to accomplish its goals individually, or in joint interaction with other QC agents. The future quantum internet is expected to consist of networked classical and quantum computers, and populated with QC agents that operate on quantum computers and communicate with each other according to the quantum communication model of either physical direct quantum transmission, or quantum teleportation, or superdense coding, each of which

M. Klusch is with the German Research Center for Artificial Intelligence, Multiagent System Group, Saarbrücken, Germany klusch@dfki.de  
R. Schubotz is with the Saarland University, Computer Science Department, Saarbrücken, Germany schubotz@gmx.de

has been experimentally verified by different research labs world wide.

### A. Classification of QC Agents

According to [8], QC agents can be classified based on the used quantum communication model. *Type-I QC agents*

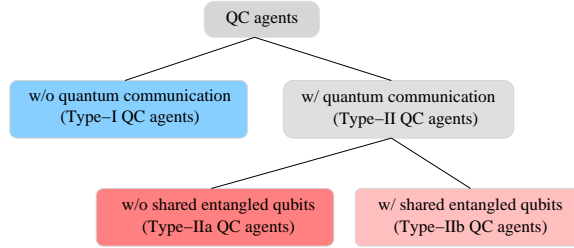


Fig. 1. Classification of QC agents

communicate over classical channels and are restricted to local quantum computing. Communication between type-I QC agents has to be additionally secured which is not necessary in case of inherently secure type-II QC agents. With respect to the classification of different types of *agent autonomy* in [4], it has been shown in [7] that the self-autonomy of individual type-I QC agents remains intact. The ability to autonomously reason about sets of goals, plans, and motivations for decision-making is not affected, since quantum computational effects are restricted to local quantum machine components. *Type-II QC agents* can be distinguished depending on whether entangled qubits for quantum communication are shared between communicating agents, or not. Communication between *type-IIa QC agents* is based on direct quantum particle transmission. In contrast to type-IIb QC agents, entangled qubits are not shared. Using the entangled qubits at it's disposal, a *type-IIb QC agent* is able to transmit superdense coded information to its communication partner, alternatively messages can be teleported. Type-II QC agents greatly benefit from computational advantages of quantum computing and communication, but at the cost of limited self-autonomy, due to non-local effects of quantum entanglement. The ability of individual type-II quantum internet agents to influence other type-II agents is inherently coupled with the risk of being influenced in turn by exactly the same agents in the same way. There is no other way of preventing such mutual remote influence than to dispense with sharing any supply of entangled qubits.

### B. Quantum Computing Design Flow

In [7] a master-slave architecture of a *hybrid quantum computer* is proposed. As depicted in figure 2, the CPU of a *classical machine* (CM) performs high-level dynamic control and scheduling of quantum operations carried out in a *quantum machine* (QM). The QM consists of quantum memory [7], quantum processing unit (QPU) with error correction, quantum bus, and quantum device controller (QDC) with interface to the CM. Using a high-level *quantum programming language* [13], [2], [14] (QPL) quantum algorithms are represented by QPL code containing high-level primitives

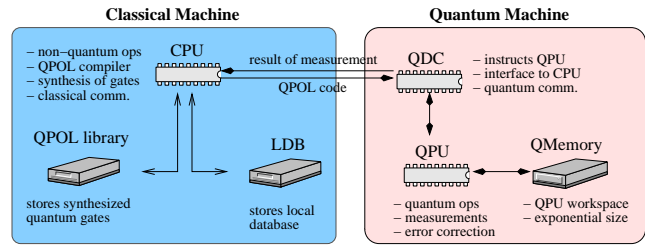


Fig. 2. Master-slave hybrid quantum computer

for logical quantum operations, interleaved with classical work-flow statements. A layered software architecture [15]

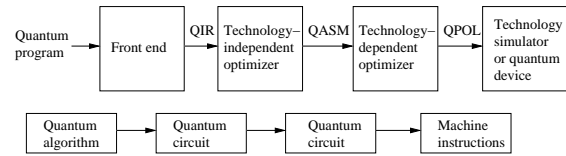


Fig. 3. Quantum design flow phases on a classical computer

involving a four-phase computer-aided design flow assists by mapping such QPL sources into efficient and robust physical implementations. During the flow's first phase as adumbrated in figure 3, a QPL source is transformed into a *quantum intermediate representation* (QIR) by the CM. Then, the CPU synthesizes and optimizes a *Quantum Assembly Language* (QASM) representation of a suitable quantum circuit. In the third phase, the QASM instructions are compiled into a device-specific *Quantum Physical Operations Language* (QPOL) representation. Finally, QPOL code is translated by the QDC into quantum machine instructions that are passed to and executed by the QPU. The QPU performs scheduled sequences of measurement and basic qubit operations with error correction to minimize quantum decoherence caused by imperfect control over qubit operations, measurement errors, number of entangled qubits, and the physical limits of the quantum systems. The QM returns the results of finalising quantum measurements to the CM.

### C. Generic QC Agent Architecture *QuantumInteRRap*

In [10], an architecture for multi-agent systems is presented. The proposed model *InteRRap* combines both the reactive and the deliberate paradigm, and explicitly represents knowledge, plans and strategies. Including a mechanism for devising joint plans, knowledge about protocols and communication strategies, *InteRRap* is suitable for describing high-level interactions of autonomous and intelligent agents. In the following, the *InteRRap* architecture is embedded in the context of QC multi-agent systems. Figure 4 shows the high-level components of the *QuantumInteRRap* architecture and their basic interplay. The *agent control unit* comprises the *world interface* (WIF), the *behaviour-based layer* (BBL), the *local planning layer* (LPL) and the *cooperative planning layer* (CPL). As adumbrated, the control components exchange goals, plans and information via communication. The agent knowledge database is organized in a hierarchical

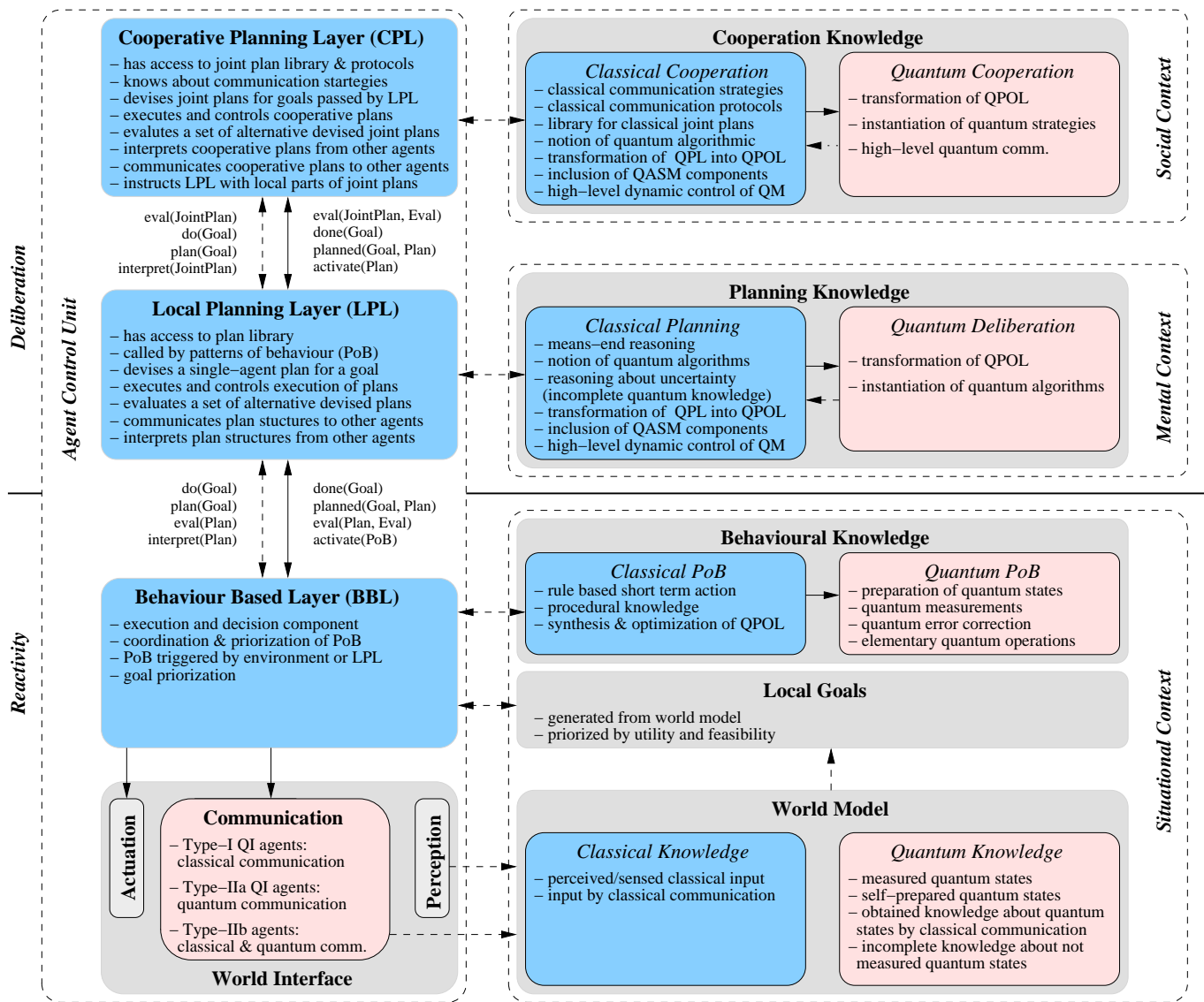


Fig. 4. QuantumInteRRap architecture

fashion and consists of the agent's *world model*, the agent's *local goals*, the *behavioural knowledge*, the *local planning knowledge* and the *cooperative planning knowledge*. In detail, the *WIF* provides the agent's means for perception, actuation and communication. According to the classification from section III-A, the *WIF* of *type-I QC agents* solely provides classical communicative facilities, whereas a *type-IIa QC agent's WIF* is solely equipped with facilities for direct quantum particle transmission, and a *type-IIb QC agent's WIF* bears both classical and quantum communicative facilities. Any information the agent receives or perceives need to be transformed into an explicit representation for storage in the agent's world model. In consequence of the deployed communicative facilities, *type-I QC agents* do not suffer from incomplete quantum knowledge. *Type-IIb QC agents* may obtain quantum knowledge by classical communication assuming a non-antagonistic agent scenario, whilst *type-IIa QC agents* are inherently uninformed about received but not

yet measured quantum bits. The *BBL* implements the agent's basic behaviour, its execution and decision component and is linked to the *WIF* for communication and actuation. The *BBL* has access to a set of executable *patterns of behaviour* (PoB) which can be activated by external stimulus or by the *LPL*. Patterns of behaviour divide into basic reactive short term actions and pieces of procedural knowledge that can be activated by the *LPL* and are appropriate to stimulate *quantum patterns of behaviour* (qPoB). Such a qPoB may be the process of preparing quantum states, or the synthesis and optimization of quantum primitives resulting in QASM code fragments (see section III-B). In order to coordinate (q)PoB, the *BBL* prioritizes the agent's goals that in turn correspond to (q)PoB. The goals of the agent are generated from its world model. The *LPL* has access to a plan library and a standard from-scratch planning algorithm. The plan library consists of classical hierarchical skeletal plans and QPL representations of *generic quantum algorithms*. The

LPL is able to device and to execute local single-agent plans, to interpret an incoming plan structure from another agent, to return plan structures to the BBL in order to tell another agent a plan for a certain goal, and to decide which plan to choose from a set of alternative plans. A chosen plan may include PoB and *quantum deliberation*. In case of quantum deliberation, the LPL instantiates an appropriate quantum algorithm from its plan library by transforming the respective QPL representation into valid QPOL representation and by including synthesized QASM fragments (see section III-B). The LPL controls the scheduling and high-level dynamic of the involved quantum machine. Again, in consequence of the deployed communication facilities, *type-II QC agents* may perform quantum operations on received quantum systems, whereas *type-I QC agents* can only operate on self-prepared quantum systems. The CPL implements a mechanism for devising joint plans based on the goals of the agent. It has a notion of protocols and communication strategies and access to a joint plan library. Similar to the LPL, the CPL is able to device and to execute joint plans for goals passed to it by its next lower layer, to interpret an incoming joint plan structure from another agent, to return joint plan structures to the LPL in order to tell other agents a joint plan for a common goal, and to decide which plan to choose from a set of alternative plans. In contrast to *type-I QC agents*, the CPL of *type-II QC agents* is able to devise joint plans involving quantum communication and quantum cooperation strategies. To this end, QPL representations of quantum concepts are transformed to valid QPOL concepts and executed on the quantum machine which is at the disposal of the type-II QC agent considered.

#### IV. A TYPE-I QUANTUM SEARCH AGENT

Quantum computers can be of avail for certain search problems. Exploiting the principles presented in chapter II, efficient quantum algorithms can be composed of a sequence of unitary operations interleaved with classical work-flow statements and a finalising measurement operation. Section IV-A explicates a *quantum pattern matching algorithm* [9] and provides the grounding for a *type-I quantum pattern matching agent* in section IV-B.

##### A. Quantum Pattern Matching

The problem of determining the *closest match* with a given pattern  $p \in \Sigma^M$  of size  $M \ll N$  in an unstructured database string  $w \in \Sigma^N$  of size  $N$  has been solved in [9] by extending Grover's quantum search algorithm [5][6]. The query complexity of QPM has been proven to be  $O(\sqrt{N-M})$  allowing for a significant speedup by an order of magnitude compared to classical matching approaches such as *approximate swapped matching*[1] in  $O(N \log(M) \log(\min(M, |\Sigma|)))$ . Applying a *compile once, run many* approach, the QPM algorithm enables to search for an arbitrary large number of distinct patterns in a given database. To this end, the  $i$ -th position of  $w$  is encoded by

$|i\rangle \in H_2^{\otimes \lceil \log(N) \rceil}$ . Hence, the quantum state

$$|\chi\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle$$

superposes all positions of database string  $w$ . For the purpose of actually generating this superposition from a simple initial state, we propose to apply the methods proposed in [16]. For each symbol  $\sigma \in \Sigma$ , a symbol oracle  $Q_\sigma$  is given by

$$Q_\sigma |\chi\rangle = (\mathbf{1} - 2 \sum_{1 \leq j \leq N} |j\rangle \langle j|) |\chi\rangle$$

for positions  $j$  of  $w$  satisfying  $w_j = \sigma$ . The algorithm is constituted by iterating the phase shifts induced by a symbol oracle  $Q_{\sigma_l}$  for a *random symbol*  $\sigma_l$ ,  $0 \leq l < M$  of the pattern followed by Grover's amplitude amplification through operator

$$P_\psi |\chi\rangle = (2 |\psi\rangle \langle \psi| - \mathbf{1}) |\chi\rangle, \quad |\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$$

In order to apply the effects of  $Q_{\sigma_l}$  to the correct starting positions  $|k\rangle$ ,  $0 \leq k \leq N - M$  of matching database substrings  $\langle w_k \dots w_{k+M-1} \rangle$ , the states  $|i\rangle$  corresponding to the positions of  $w$  need to be permuted in each iteration. This can be done reversibly using

$$P_\pi^l |\chi\rangle = P_\pi^l \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i+l \pmod N\rangle$$

If  $M \ll N$ , sampling randomly over  $M$  elements will lead to searching with high probability over all symbols of the pattern. On average, a position with a partial match of  $M' \leq M$  individual symbols will experience  $\frac{M'}{M}$  phase shifts.

---

#### Algorithm 1 Quantum pattern matching

---

**Input:**  $w \in \Sigma^N, p \in \Sigma^M, n = \lceil \log(N) \rceil$

**Output:**  $m \in \mathbb{N}$

**Quantum Variables:**  $|\chi\rangle \in H_2^{\otimes n}$

**Classic Variables:**  $r, i, j \in \mathbb{N}$

---

- 1: Choose  $r \in [0, \lfloor \sqrt{N-M+1} \rfloor]$  uniformly
  - 2: Set  $|\chi\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |k\rangle$
  - 3: **for**  $i = 1$  to  $r$  **do**
  - 4:   Choose  $j \in [0, M-1]$  uniformly
  - 5:   Set  $|\chi\rangle = P_\pi^j |\chi\rangle$
  - 6:   Set  $|\chi\rangle = Q_{\sigma_j} |\chi\rangle$
  - 7:   Set  $|\chi\rangle = (P_\pi^j)^{-1} |\chi\rangle$
  - 8:   Set  $|\chi\rangle = P_\psi |\chi\rangle$
  - 9: **end for**
  - 10: Set  $m$  to the result of the measurement of  $|\chi\rangle$
- 

##### B. Type-I Quantum Search Agent Architecture

How to model a type-I quantum search agent (QSA) by use of the QuantumInterRap architecture presented in section III-C? According to section III-B, the QSA is supposed to

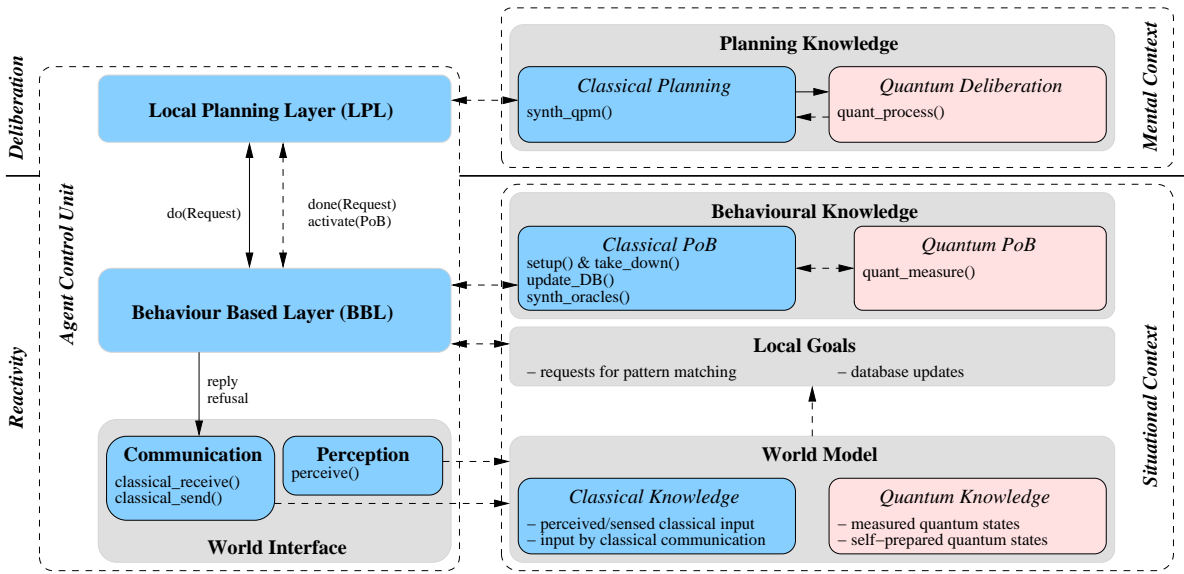


Fig. 5. QuantumInteRRap architecture of a QPM based type-I quantum search agent

run on a hybrid quantum computer which is, e.g., provided with a unstructured classical *local database* (LDB). For the sake of simplicity, we consider this LDB as a long static string  $w \in \Sigma^N$  of size  $N$ . Figure 5 illustrates how a *Type-I QPM agent* can be modeled as a *QuantumInteRRap agent*.

Once a setup or update of LDB is perceived (see figure 6), the symbol oracles  $Q_\sigma$  for  $\sigma \in \Sigma$  need to be synthesized by the BBL using a QASM compiler. The resulting quantum

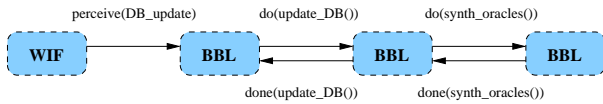


Fig. 6. Processing a database update

circuits are stored in a QASM representation for further usage and the oracle synthesis can be stimulated on the basis of section IV-A using function

```
void synth_oracles( ... )
```

Upon receipt of request  $s$  containing some pattern  $p \in \Sigma^M$  from another agent  $A_1$  (see figure 7), the QSA locally computes the index of the closest match  $m$  to  $p$  in LDB and returns  $m$  to  $A_1$ . Both requests and replies are transmitted via classical channel, and requests are directly quantum coded (see section III-B) prior to quantum pattern matching. In more detail, the BBL asks the LPL to process the transmitted pattern  $p \in \Sigma^M$ . To this end, the LPL devises and executes a local plan involving a proper instance of the quantum pattern matching algorithm by means of

```
synth_qpm( ... )
```

The instantiation of the QPM algorithm includes necessary symbol oracles computed by the BBL into the skeletal QPM algorithm scheme. The resulting QASM code makes use of qPoB to measure quantum states, for example. It is

transformed into valid QPOL and transmitted to the QDC via

```
quant_process( ... )
```

After instructing the QPU to perform a quantum pattern matching, the QDC returns the measurement result back to the LPL. By passing the result to lower level of the world interface, the requesting agent  $A_1$  receives the computed closest match to its query string, or a failure message in case of the search failed.

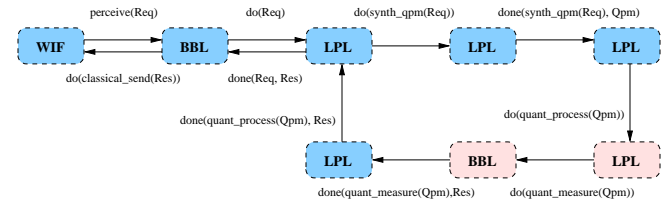


Fig. 7. Processing a pattern request

## V. SIMULATION AND BENCHMARKING

*Quantum computer simulators* enable the simulation of computational operations designed for quantum machines on classical computing machinery. To demonstrate the operability of our type-I QSA, we have programmed and simulated it by use of open source quantum simulators *QuIDDPro 2.1* [19], [18], [17], *QCL 0.6.1* [13] and *libquantum 0.2.4* [3]. For extensive review of quantum computer simulators, we refer the reader to [20]. Memory and runtime performances for simulations on a classical computer with query patterns of size  $|p| = 4$  and various database strings are displayed in figure 8. We simulated the QSA for three different use case scenarios: LDB set up with Matsuo Basho's *Frog Haiku* encoded in 6 qubits, LDB set up with Robert Frost's *Fire and Ice* encoded in 8 qubits, and LDB set up with



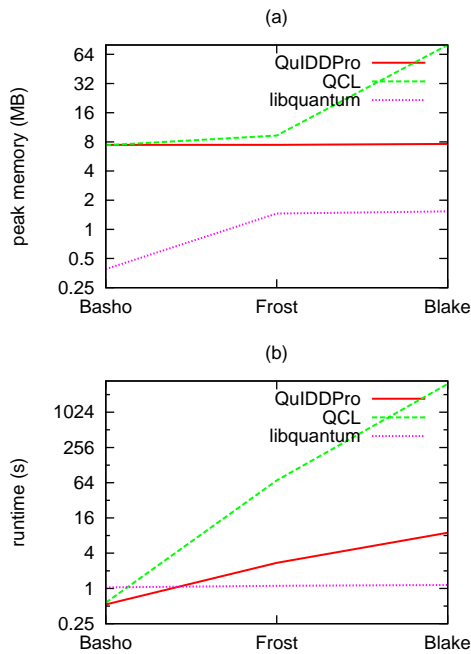


Fig. 8. Comparing peak memory (a) and runtime (b) for selected quantum computing simulators

William Blake's *The Tyger* encoded in 10 qubits. We strongly highlight, that our comparative evaluation solely investigates selected quantum computer simulators with respect to memory and runtime performance on classical machines. Using a not yet implemented quantum computing device, QPM's quantum query complexity of  $O(\sqrt{N - M})$  will allow for a significant speedup of our QSA.

## VI. RELATED WORK

To the best of our knowledge, the presented architecture, programming and simulation of a type-I quantum search agent is unique as there are no alternative QC agent implementations available yet. Our work combines related work from quantum computing and agent based computing, and builds upon existing work on QC agents in [7]. In particular, we did exploit an improved version of the recently proposed quantum pattern matching algorithm in [9] for speeding up the local search process of a type-I quantum search agent. The architecture of such an agent on a hybrid quantum computer is then proposed to be an appropriately extended version of the classic InteRRap agent architecture in [10] for QC agents on hybrid quantum computers. The programming of this special kind of QC agent is given by means of three different existing quantum simulators, and demonstrated by example.

## VII. CONCLUSION

In this paper, we presented a hybrid generic architecture for QC agents as an extension of the known InteRRap architecture, discussed basic engineering aspects of how to realize QC agents on hybrid quantum computers, instantiated the

QuantumInteRRap architecture by concrete means of a quantum pattern matching based type-I quantum search agent, showed the results of comparative run time performance testing of its simulation with different quantum simulators, and demonstrated its functionality also by example. The theoretical performance of the QPM based quantum search agent over classical edit based string matching provides strong evidence for the expected significant speed up of a service matchmaking process of type-I quantum matchmaker agents compared to the classical case. However, the quantum realization of semantic service matchmaking remains one open problem. Our ongoing research focuses on solving this problem by feasible type-II QC agents for service selection, that is the programming and simulation of type-II quantum matchmaker agents with QuantumInteRRap architecture and by use of the same quantum simulators.

## REFERENCES

- [1] Amir. Approximate swapped matching. *Inf. Process. Lett.*, 83(1):33–39, 2002.
- [2] Bettelli. Toward an architecture for quantum programming, *eur. phys. j.*, 25:181–200, 2003., 2003.
- [3] B. Butscher. Non-technical description of libquantum, [enyo.de/libquantum/](http://enyo.de/libquantum/).
- [4] C. Carabelea, O. Boissier, and A. Florea. Autonomy in multi-agent systems: A classification attempt. In Nickles et al. [11], pages 103–113.
- [5] L. K. Grover. A fast quantum mechanical algorithm for database search, [arxiv.org/quant-ph/9605043](http://arxiv.org/quant-ph/9605043), 1996.
- [6] L. K. Grover. From schrödinger's equation to the quantum search algorithm. *American Journal of Physics*, 69(7):769–777, 2001.
- [7] M. Klusch. Toward quantum computational agents. In Nickles et al. [11], pages 170–186.
- [8] M. Klusch. Coordination of quantum internet agents. In F. Dignum, V. Dignum, S. Koenig, S. Kraus, M. P. Singh, and M. Wooldridge, editors, *AAMAS*, pages 1221–1222. ACM, 2005.
- [9] P. Mateus and Y. Omar. Quantum pattern matching, [arxiv.org/quant-ph/0508237](http://arxiv.org/quant-ph/0508237). 2005.
- [10] J. Müller and M. Pischel. The agent architecture interrapp: Concept and application, technical report rr-93-26, dfki saarbrücken, 1993, 1993.
- [11] M. Nickles, M. Rovatsos, and G. Weiß, editors. *Agents and Computational Autonomy (AAMAS 2003)*.
- [12] M. A. Nielsen and I. L. Chuang. *Quantum computation and quantum information*. Cambridge Univ. Press, Cambridge, 2000.
- [13] B. Oemer. Quantum programming in qcl, master thesis, technical university of vienna, computer science department, 2000.
- [14] P. Selinger. Towards a quantum programming language. *Mathematical Structures in Comp. Sci.*, 14(4):527–586, 2004.
- [15] K. M. Svore, A. V. Aho, A. W. Cross, I. Chuang, and I. L. Markov. A layered software architecture for quantum computing design tools. *Computer*, 39(1):74–83, 2006.
- [16] C. A. Trugenberger. Phase transitions in quantum pattern recognition, [arxiv.org/quant-ph/0204115](http://arxiv.org/quant-ph/0204115). 2002.
- [17] Viamontes. Gate-level simulation of quantum circuits, [arxiv.org/quant-ph/0208003](http://arxiv.org/quant-ph/0208003), 2002.
- [18] G. F. Viamontes, I. L. Markov, and J. P. Hayes. Improving gate-level simulation of quantum circuits. *Quantum Information Processing*, 2:347, 2003.
- [19] G. F. Viamontes, I. L. Markov, and J. P. Hayes. Graph-based simulation of quantum computation in the density matrix representation. *Quantum Information and Computing*, 5:113, 2005.
- [20] J. Wallace. Quantum computer simulators - a review version 2.1, [citeseer.ist.psu.edu/wallace99quantum.html](http://citeseer.ist.psu.edu/wallace99quantum.html).

---

## Quantum Matchmaker Agents

M. Klusch: Quantum Matchmaking. Extended version of paper "Coordination of Quantum Internet Agents" published in the Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS), New York, USA, ACM Press, 2005.

# Quantum Matchmaking

Matthias Klusch

German Research Center for Artificial Intelligence  
Deduction and Multiagent Systems

Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany.  
e-mail: klusch@dfki.de

**Abstract**—Intelligent agents in the quantum internet are supposed to operate on networked hybrid quantum computers to individually or jointly accomplish their goals by means of both classical and quantum computation, and communication. We present initial quantum based solutions to the classical coordination problem of matchmaking which can be performed under certain conditions more efficient and secure than in the classical case.

## I. INTRODUCTION

Quantum computation provides a paradigm for information processing that differs fundamentally from ordinary digital computation: Information is mechanical, that is the way in which quantum systems such as spins, photons, and atoms store and process information is inherently governed by the laws of quantum physics. Quantum physics uses quantum mechanics as a mathematical language to explain nature at the atomic scale, in particular, superposition of quantum states that enables for quantum parallelism, interference effects during the course of unitary state evolution, and non-local effects of spatially separated but quantum entangled data that are impossible to realize by means of classical physics. Quantum computing devices have been physically implemented since the late 1990's by use of, for example, nuclear magnetic resonance, and solid state technologies. Current efforts in nanoscale molecular engineering, and achievements made in realizing few qubit quantum processors and quantum communication channels provide strong evidence in favor of the development of more sophisticated and networked quantum computing devices that will make up the so called quantum internet beyond 2020. How shall intelligent software agents perform service matchmaking in the quantum internet, that is how to connect the ultimate service requester with the ultimate service provider?

## II. QUANTUM INTERNET AGENTS

Quantum computation is the extension of classical computation to the processing of quantum information based on physical two-state quantum systems such as photons. The unit of quantum information is the quantum bit (qubit) with coherent superposed basis states; qubit registers can be described as a tensor product of its component qubit states in complex Hilbert space. In particular, measuring one of entangled qubits can instantaneously affect the probability amplitudes of the other qubit no matter how far they are spatially separated. For a comprehensive introduction

to quantum computing and communication we refer the interested reader to, for example, [18].

A quantum computational agent (QC agent) [15] is an intelligent software agent that is able to perform both classical and quantum computing to accomplish its goals individually, or in joint interaction with other QC agents. The future quantum internet is expected to consist of networked classical and quantum computers, and populated with QC agents, so called quantum internet agents, that operate on quantum computers and communicate with each other according to the quantum communication model of either physical direct quantum transmission, or quantum teleportation, or quantum dense coding, each of which has been experimentally verified. Quantum internet agents can be classified based on the used quantum communication model.

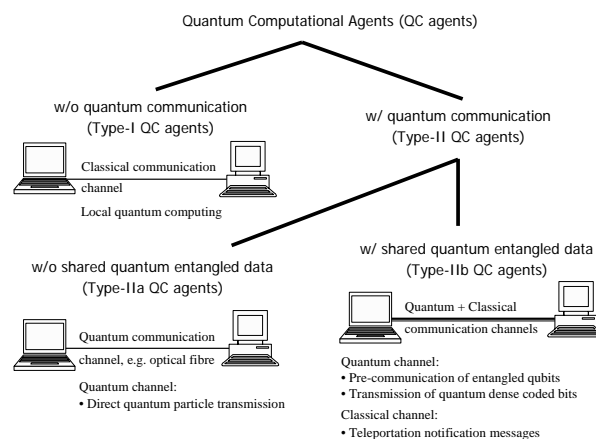


Fig. 1. Communication based classification of quantum internet agents.

It has been shown in [15] that QC agents are feasible to implement on a hybrid quantum computer in principle.

## III. QUANTUM MATCHMAKING

The so-called connection problem of today's internet is to effectively connect the ultimate service requester agent with the ultimate service provider agent. Prominent classical solutions to this coordination problem propose the use of appropriate intelligent middle agents such as matchmakers, brokers, and mediators [16]. The classical matchmaking cycle usually starts with service providers submitting advertisements to the matchmaker. Upon receipt of a service request from some requester agent, the matchmaker searches

its local advertisement database for those which best match the request, and returns these together with the addresses of the corresponding service provider agents to the requester. In contrast to classical brokerage, the matchmaker leaves it to the requester agent to contact the relevant providers, and to negotiate access to the selected relevant services.

In the future quantum internet, the same problem can be solved, in principle, more efficient in terms of both computation and communication complexity, by use of quantum matchmaker agents. In the following, we distinguish between the scenarios of type-I and type-II quantum matchmaking depending on the type of QC agents involved.

### A. Type-I quantum matchmaking

The problem of coordinating type-I QC agents in the quantum internet can be solved by use of a central type-I quantum matchmaker agent. As in the classical case, service provider and requester agents are supposed to communicate to the matchmaker over classical channels. The only difference to the quantum case is that the matchmaker agent performs an oracle-based quantum search of its local classical database of size  $N$  for those advertisements that best match a given request. The quantum matchmaking protocol for type-I QC agents is provided in figure 2.

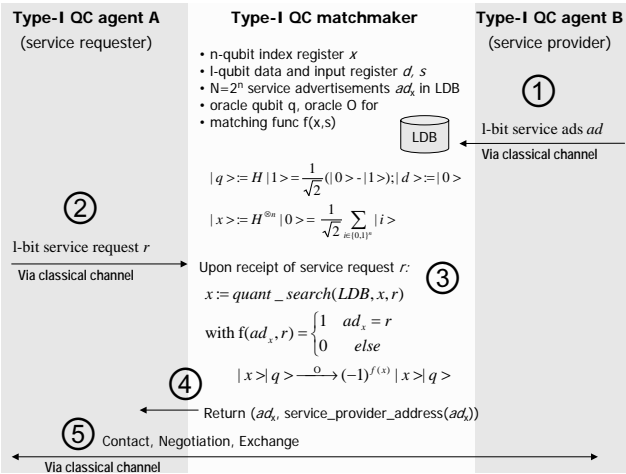


Fig. 2. Quantum matchmaking of type-I QC agents

In order to perform its quantum search the matchmaker makes use of an  $n$ -qubit index register,  $l$ -qubit data and input registers, and 1-qubit oracle register of the quantum processing unit of the hybrid quantum computer upon which it is operating. Each of the  $N = 2^n$   $l$ -bit advertisements  $ad_x$  stored in the database is indexed by value  $x = 0 \dots N-1$ , with initial superposed  $n$ -qubit quantum index state  $|x\rangle$  created by  $n$ -folded Hadamard operation  $H$  on state  $|0\rangle$  yielding  $H^{\otimes n}|0\rangle = \frac{1}{\sqrt{2^n}} \sum_{i \in \{0,1\}^n} |i\rangle$ . The quantum index can be implemented by means of an quantum addressing scheme such as the one proposed by Nielsen and Chuang ([18], p.268). Like in classical schemes, it takes  $O(\log N)$  time per operation to access, load, and restore indexed data  $ad_x$  to the data register  $|d\rangle$ , and recreate respectively measured

quantum index states  $|x\rangle$  for further processing by the quantum matchmaker.

Upon receipt of a  $l$ -bit service request  $r$ , the matchmaker searches its local database on the quantum index by use of a given search oracle  $O$  with quantum counted  $1 \leq M \leq N$  solutions. The oracle is representing the core classical matching operation  $f$  to be applied to  $r$  and each indexed advertisement  $ad_x$ , checking whether  $ad_x$  matches the request  $r$ , hence is a solution to the local search problem such as an exact match  $f(x) = 1$  if  $ad_x = r$ , else  $f(x) = 0$ .

Any classical search of the matchmaker agent would take an average of  $O(N)$  or  $O(N/M)$  oracle calls, to find one or  $M$  solutions, respectively. However, the type-I quantum matchmaker agent can polynomially speed up this search to be performed in  $O(\sqrt{N})$  ( $O(\sqrt{N/M})$ ) time by using Grover's quantum search algorithm [8]. This search is performed on the  $n$ -qubit index register  $|x\rangle$  which initial state is in superposition of all  $N$  index values  $x$ . Key for speeding up the search process as opposed to the classical case is that the oracle  $O$  marks the  $M$  solutions  $|x\rangle \rightarrow (-1)^{f(x)}|x\rangle$  with  $f(x) = 1$  each of which gets iteratively amplified such that the probability that they will be found upon measurement of the index register after  $O(\sqrt{N/M})$  ( $O(\sqrt{N/M})$ ) iterations sufficiently increases.

Each classical matching algorithm computing  $f$  can be converted into an equivalent quantum operator  $U_f$  that is implementable in an appropriate quantum logical circuit with the same order of efficiency [1]. In cases the matchmaker is making use of quantum coded matching algorithms, each advertisement  $ad_x$  and request  $r$  are loaded into its qubit data and input register, respectively. The matching problem  $|ad_x\rangle = |r\rangle$  can then be solved, for example, under certain conditions in  $O(1)$  using the general Deutsch-Josza algorithm.

It follows that any type-I quantum matchmaker agent could perform service matchmaking of type-I quantum internet agents more efficient than any of its classical counterparts in principle. However, classical communication between agents involved have to be additionally secured by application of appropriate security means. This is not necessary in case of type-II quantum matchmaking.

### B. Type-II quantum matchmaking

We distinguish between two scenarios of type-II quantum matchmaking depending on whether the type-II QC agents involved are sharing sufficient supplies of entangled qubits for quantum communication, or not. Since each implied quantum communication model (cf. section 2) is physically secure, this holds, in particular, for the corresponding case of type-II quantum matchmaking.

1) *Matchmaking with shared entanglement*: Suppose that type-IIb quantum internet agent  $A$  transmits  $N$  quantum dense coded  $n$ -bit service advertisements and requests each of size  $n/2$  qubits to a type-IIb QC matchmaker  $M$  over a quantum channel. Alternatively,  $A$  could teleport its messages to  $M$  at the cost of  $2n$  bits via a classical channel. Like in the scenario of type-I quantum matchmaking, the

matchmaker then quantum searches its database and returns those advertisements that match according to the applied individual matching oracle.

The complexity of quantum dense coding based communication between each pair of agents involved in the matchmaking process is reduced by half to  $O(l/2)$  per 1-qubit message exchanged. For example, type-IIb QC agent  $A$  could transmit one 2Mbit message to the matchmaker  $M$  by manipulating its share of entangled 1Mqubit pairs, and then sending only those to  $M$ . This immense reduction in communication could be of potential benefit also for a variety of today's data-intensive applications such as large scale distributed data mining and information gathering, and communication with or between autonomous spacecrafts where transmissions are prohibitively costly.

The type-IIb quantum matchmaking process restricted to the interaction between service requester and matchmaker agent using quantum dense coding for communication is provided in figure 3.

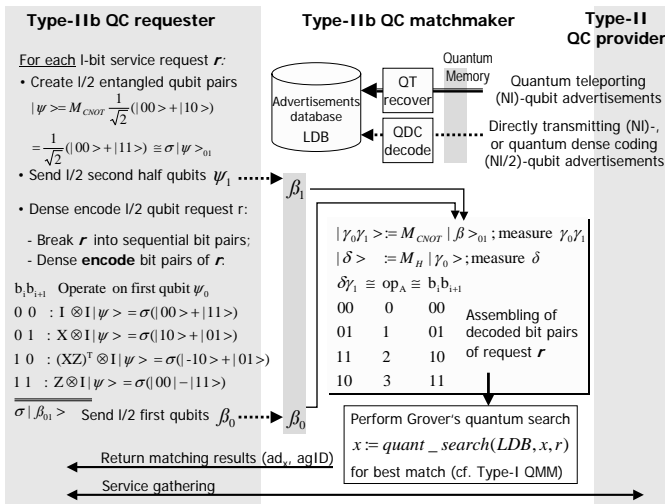


Fig. 3. Type-IIb quantum matchmaking with dense coding based communication

Each type-IIb QC agent is able to communicate via quantum channels either using entanglement based quantum dense coding, or teleportation. In the considered matchmaking scenario, we assume that the sender of messages is responsible for creating and pre-communicating required supplies of entangled qubits to the intended receiver. In addition, communicating type-IIb QC agent pairs are supposed to agree in prior on the individually used order of quantum en-/decoding operations for sequential pairs of classical bits of messages to be transmitted (cf. section 3.1).

Alternatively, the agents could contact some trusted third party for creating and obtaining sufficient shares of entangled qubits on demand. However, in some application environments third trusted parties and pre-communicating supplies of entangled qubits could be unrealistic. Though in both cases of entanglement based quantum communication there is no real need to pre-communicate entangled qubits, it turns out, that the corresponding solutions come at the cost of inherent security and increase of communication complexity

between agent pairs involved.

For example, in case of teleportation, any type-IIb QC agent  $A$  could rather exclusively create sufficient supplies of entangled pairs of qubits depending on the actual volume of data to be transmitted to some type-IIb QC agent  $B$ . It then performs local operations on one half of the qubit pairs according to the standard teleportation procedure (cf. section 3.1), thereby implicitly projecting the other half of qubits, and sends the latter together with the 2-bit teleportation notification messages to agent  $B$ . One drawback of this variant of teleportation is, that it is not only introducing an additional quantum channel between  $A$  and  $B$  but is insecure, since the complete information required by  $B$  to reconstruct  $A$ 's message would become available for any third party agent that intercepts both quantum and classical channel between  $A$  and  $B$ .

Alternatively, agent  $A$  could transmit the complete set of pairs of qubits of quantum dense coded messages to  $B$  over a quantum channel. The drawback of this solution is, that it is not only insecure, but one would even lose the benefit of reducing the volume of data to be transmitted by half between communicating pairs of agents.

On the other hand, in some application scenarios this benefit may even be out of proportion to the mutual trust required among the type-IIb QC agents involved. In particular, any type-II QC agent can change the state of non-local qubits of other type-II QC agents that are entangled with its own qubits by local Bell state measurements. As a consequence, any type-II QC agent is in principle able to instantaneously corrupt specific individual or joint quantum computation with other type-II QC agents, as it can be influenced by the same agents the same way.

There is no other way of preventing such mutual remote influence than to dispense with sharing any supply of entangled quantum data. This is the case for type-II quantum matchmaking without shared entanglement.

2) *Matchmaking without shared entanglement*: Suppose type-IIa quantum internet agents directly transmit their  $l$ -qubit service advertisements  $ad$ , or requests  $r$  to the type-II quantum matchmaker agent via quantum wires. Upon receipt of request  $r$  the matchmaker checks whether  $r$  exactly matches with any of its locally stored and indexed advertisements  $ad_x$  ( $|ad_x\rangle = |r\rangle$ ) with the extra promise that the Hamming distance  $h(r, ad_x)$  between both qubit strings, that is the number of qubits where  $ad_x$  and  $r$  are different, is either 0 or  $l/2$ . Hence, in the following, we restrict the quantum service matching operation to the qubit comparison level. The corresponding protocol for quantum matchmaking of type-IIa QC agents using Grover's quantum search combined with Deutsch-Josza's matching function evaluation is provided in figure 4.

The requester agent either submits quantum coded or binary coded request  $r$  to the matchmaker. In first case, that means sending  $\log(l) + 1$  qubits in prepared quantum state ( $l = 2^m$ )

$$|r\rangle = \frac{1}{\sqrt{m}} \sum_{i \in \{0,1\}^{\log(l)}} |i\rangle \frac{1}{\sqrt{2}} (|0 \oplus s_i\rangle - |1 \oplus s_i\rangle)$$

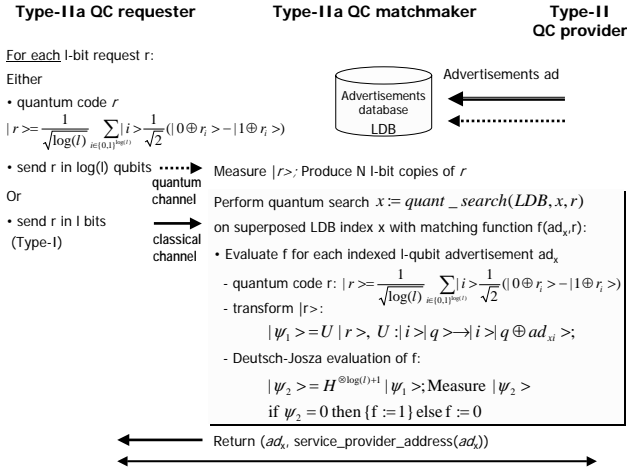


Fig. 4. Quantum matchmaking by integrated application of Grover's quantum search, and Deutsch-Josza's quantum evaluation

of its l-bit request  $r$  to the matchmaker via a quantum channel. The matchmaker produces  $N$  copies of  $r$  for repeated quantum coding of  $r$  during quantum search.

The key idea is, that under the Hamming distance promise, the quantum matchmaker just has to figure out for each indexed advertisement  $ad_x$  whether  $ad_{x1} \oplus r_1 \dots ad_{xl} \oplus r_l = 0$ , hence is constant, or balanced, in which case  $ad_x = r$  or  $ad_x \neq r$  holds, respectively. For this purpose, the matchmaker agent quantum searches the superposed index of its advertisement database while using Deutsch-Josza's parallel quantum evaluation algorithm to evaluate the matching function used.

During quantum search, for each considered advertisement  $ad_x$  it applies the unitary operation  $|i\rangle |q\rangle \rightarrow |i\rangle |q \oplus ad_{xi}\rangle$  to the quantum coded request state  $|r\rangle$  yielding state  $|\psi_1\rangle$  which is equal to

$$\frac{1}{\sqrt{m}} \sum_{i \in \{0,1\}^m} |i\rangle \frac{1}{\sqrt{2}} (|0 \oplus r_i \oplus ad_{xi}\rangle - |1 \oplus r_i \oplus ad_{xi}\rangle)$$

This state can be rewritten as

$$|\psi_1\rangle = \frac{1}{\sqrt{m}} \sum_{i \in \{0,1\}^{\log(l)}} (-1)^{f_{\oplus}(i)} |i\rangle \frac{(|0\rangle - |1\rangle)}{\sqrt{2}}$$

The search oracle, or matching function  $f(ad_x \oplus r)$  is checked through evaluation of  $f_{\oplus}(i) = ad_{xi} \oplus r_i \in \{0,1\}$  for all  $i \in \{1..l\}$  in parallel according to the general Deutsch-Josza quantum algorithm [18]. For this purpose, the agent applies the Hadamard transform  $H^{\oplus \log(l)+1}$  to state  $|\psi_1\rangle$  resulting in state

$$|\psi_2\rangle = \sum_{z \in \{0,1\}^m} \sum_{i \in \{0,1\}^m} \frac{(-1)^{f_{\oplus}(i)+i \cdot z}}{2^n} |z\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

Measurement of  $|\psi_2\rangle$  yields 0 only if the matching function  $f$  is constantly 0, hence  $ad_x = r$ .<sup>1</sup>

<sup>1</sup>If  $f$  is not constant, the probability amplitudes of this case destructively interfere to produce a probability of zero, hence measurement would yield anything except 0.

Evaluation of  $f$  during quantum search is performed in  $O(1)$  calls which is an exponential speed up over the classical case of  $O(2^l/2)$  evaluations. Hence, the overall computational complexity is  $O(\sqrt{N})$  per service request and  $N$  indexed advertisements communicated in  $O(N \cdot \log(l))$ .

Matchmaker Agent	Computation	Communication	Privacy	Influence
Quantum				
Type-I	$O(\sqrt{N} \cdot m)$	$O(N \cdot n)$	no	no
Type-IIa	$O(\sqrt{N} \cdot m)$	$O(N \cdot \log(l))$	yes	no
Type-IIb	$O(\sqrt{N} \cdot m)$	$O(N \cdot n/2)$	yes	yes
Classical	$O(N \cdot m)$	$O(N \cdot n)$	no	no

$N$  advertisements of  $n$ -bit or  $l$ -qubits

$m$  = complexity of (quantum) matching

Influence = Instantaneous, remote manipulation of entangled qubits possible

Fig. 5. Comparison of quantum and classical matchmaking

Figure 5 summarizes the features of quantum versus classical service matchmaking.

#### IV. CONCLUSIONS

We presented means of coordinating quantum internet agents in terms of quantum matchmaking which can be performed under certain conditions more efficient than in the classical case. In addition, quantum communication between any pair of type-II quantum internet agents involved in the process of quantum matchmaking is per se physically secure. Our current cross disciplinary work includes the development of secure quantum coordination means of type-II multi-agent quantum matchmaking without a central quantum matchmaker agent.

#### REFERENCES

- [1] D. Aharonov. Quantum Computation. LANL Archive quant-ph/981203, 1998.
- [2] C.H. Bennett, S.J. Wiesner. Communication via one- and two-particle operators on EPR states. Phys. Review Letters, 69(20), 1992.
- [3] C.H. Bennett, G. Brassard, C. Crepau, R. Josza, A. Peres, W.K. Wootters. Phys. Reviews Letters, 70, 1895, 1993.
- [4] S. Betteli, T. Calarco, L. Serafini. Toward an Architecture for Quantum Programming. LANL Archive cs.PL/0103009, March 2003.
- [5] H. Buhrman, R. Cleve, A. Wigderson. Quantum vs. Classical Communication and Computation. Proc. 30th Ann. ACM Symp. Theory of Computing (STOC 98), 1998.
- [6] D. Bouwmeester, A. Ekert, A. Zeilinger. The Physics of Quantum Information. Springer, Heidelberg, 2000.
- [7] G.K. Brennen, D.Song, C.J. Williams. A Quantum Computer Architecture using Nonlocal Interactions. LANL Archive quant-ph/0301012, 2003.
- [8] L. Grover. A Fast Quantum Mechanical Algorithm for Database Search. Proc. 28th Annual ACM Symposium on Theory of Computation, ACM Press, NY USA, pp 212-219, 1996.
- [9] P. Hayden, D. Deutsch. Information flow in entangled quantum systems. Proc. Royal Society London, A456, 2000.
- [10] A.S. Holevo. Some estimates of the information transmitted by quantum communication channels. Problems of Information Transmission, 9:177-183, 1973.

- [11] B.A. Huberman and T. Hogg. Quantum Solution of Coordination Problems. Information Dynamics Laboratory, HP Labs, <http://www.hpl.hp.com/research/idl/papers/coordination/>, 2004
- [12] R. Josza. Entanglement and Quantum Computation. Geometric Issues in the Foundations of Science, S. Huggett et al. (eds.), Oxford University Press, 1997. quant-ph/9707034.
- [13] B. Kane. A Silicon-Based Nuclear Spin Quantum Computer. *Nature*, 393, 1998
- [14] M. Klusch. Information Agent Technology for the Internet: A Survey. *Data and Knowledge Engineering*, 36(3), Elsevier Science, 2001.
- [15] M. Klusch. Toward Quantum Computational Agents. In: Nickles, Rovatsos, Weiss (eds.), *Computational Autonomy. Lecture Notes in Artificial Intelligence*, vol. 2969, Springer, 2004.
- [16] M. Klusch, K. Sycara. Brokering and Matchmaking for Coordination of Agent Societies: A Survey. In: *Coordination of Internet Agents*, A. Omicini et al. (eds), Springer, 2001.
- [17] S. Lloyd, M.S. Shariar, P.R. Hemmer. Long Distance, Unconditional Teleportation of Atomic States Via Complete Bell State Measurements. <http://xxx.lanl.gov/pdf/quant-ph/0003147>. 2004.
- [18] M.A. Nielsen, I.L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, UK, 2000.
- [19] B. Oemer. *Quantum Programming in QCL*. Master Thesis, Technical University of Vienna, Computer Science Department, Vienna, Austria, 2000.
- [20] M. Oskin, F.T. Chong, I.L. Chuang. A Practical Architecture for Reliable Quantum Computers. *IEEE Computer*, 35:79-87, January 2002.
- [21] M. Oskin, F.T. Chong, I.L. Chuang, J. Kubiawicz. Building Quantum Wires: The Long and the Short of it. *Proc. 30th Intl Symposium on Computer Architecture (ISCA)*, 2003.
- [22] P. Selinger. *Towards a Quantum Programming Language*. *Mathematical Structures in Computer Science*, 2003.
- [23] P. Shor. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. *Proc. 35th Annual Symposium on Foundations of Computer Science*, Los Alamitos, USA, 1994.
- [24] M. Steffen, L.M.K. Vandersypen, I.L. Chuang. Toward Quantum Computation: A Five-Qubit Quantum Processor. *IEEE Micro*, March/April, 2001.
- [25] P. Weiss. Quantum Internet - Possible use of quantum mechanics in computer networks. *Science News*, Vol. 155, No. 14, April 3, 1999.
- [26] M. Wooldridge. *An Introduction to Multiagent Systems*. John Wiley & Sons, Chichester, UK, 2002.
- [27] W.K. Wootters, W.H. Zurek. A Single Quantum Cannot be Cloned. *Nature*, 299:802-803, 1982.





---

## References

1. K. Aberer, P. Cudre-Mauroux, M. Hauswirth: Semantic Gossiping: fostering semantic interoperability in peer data management systems. S. Staab, H. Stuckenschmidt (eds.): *Semantic Web and Peer-to-Peer*, Springer, Chapter 13, 2006.
2. AgentLink III: 50 facts about agent-based computing. University of Southampton, UK, AgentLink ([www.agentlink.org](http://www.agentlink.org)), 2005.
3. G. Agre, Z. Marinova: An INFRAWEBs Approach to Dynamic Composition of Semantic Web Services. *Cybernetics and Information Technologies*, 7(1), 2007.
4. S. Agarwal, S. Handschuh, S. Staab: Annotation, composition and invocation of semantic web services. *Web Semantics*, 2, 2004.
5. M.S. Aktas, G. Fox, M. Pierce: Managing Dynamic Metadata as Context. Proceedings of Intl. Conference on Computational Science and Engineering (ICCSE), Istanbul, 2005.
6. G. Alonso, F. Casati, H. Kuno, V. Machiraju: *Web Services*. Springer, 2003
7. S. Amer-Yahia, C. Botev, J. Shanmugasundaram: TeXQuery: A Full-Text Search Extension to XQuery. Proceedings of the World-Wide-Web Conference WWW 2004, 2004.
8. J. Angele, G. Lausen: Ontologies in F-logic. In: *Handbook on Ontologies*, eds. S. Staab, R. Studer, Springer, 2004.
9. A. Ankolekar, M. Paolucci, K. Sycara: Spinning the OWL-S Process Model - Toward the Verification of the OWL-S Process Models. Proceedings of International Semantic Web Services Workshop (SWSW), 2004.
10. S. Androutsellis-Theotokis, D. Spinellis: A survey of peer-to-peer content distribution technologies. *ACM Computing Surveys*, 36(4), 2004.
11. G. Antoniou, F. van Harmelen: *A Semantic Web Primer*. MIT Press, 2004.
12. T. Arnold, U. Schwalbe: Dynamic coalition formation and the core. *Economic Behavior & Organization*, 49(3), 2002.
13. I.B. Arpinar, B. Aleman-Meza, R. Zhang, A. Maduko: Ontology-driven web services composition platform. Proceedings of IEEE International Conference on E-Commerce Technology (CEC), San Diego, USA, IEEE Press, 2004.
14. S. Arroyo-Camejo: *Skurrile Quantenwelt*. Springer, 2006.
15. D. Artz, Yolanda Gil: A survey of trust in computer science and the semantic web. *Web Semantics*, 5(2), Elsevier, 2007.
16. M.A. Aslam, S. Auer, J. Shen: From BPEL4WS Process Model to Full OWL-S Ontology. Proceedings of 2nd European Conference on Semantic Web Services (ESWC), Buda, Montenegro, 2006.

17. Autonomous Agents and Multi-Agent Systems. International Journal, Springer, M. Wooldridge, K. Sycara (eds.), <http://springerlink.metapress.com/content/1573-7454/>
18. F. Baader, B. Hollunder: Embedding defaults into terminological knowledge representation. Proceedings of International Conference on Knowledge Representation and Reasoning, 1992.
19. F. Baader, D. Calvanese, D. McGuinness (eds.): The Description Logic Handbook: Theory, Implementation and Applications. Cambridge University Press, 2003.
20. F. Baader, C. Lutz, M. Milicic, U. Sattler, F. Wolter: Integrating Description Logics and action formalisms: First results. Proceedings 20th National Conference on Artificial Intelligence (AAAI), Pittsburgh, USA, AAAI Press, 2005
21. J. Bae, L. Liu, J. Caverlee, W.B. Rouse: Process Mining, Discovery, and Integration using Distance Measures. Proceedings of International Conference on Web Services (ICWS), 2006.
22. M. Baldoni, C. Baroglio, A. Martelli, V. Patti, and C. Schifanella: Verifying the conformance of web services to global interaction protocols: a first step. Proceedings of the 2nd International Workshop on Web Services and Formal Methods, Lecture Notes in Computer Science (LNCS) vol. 3670, Springer, 2005.
23. S. Bansal, J. Vidal: Matchmaking of Web Services Based on the DAMLS Service Model. Proceedings of International Joint Conference on Autonomous Agents and Multiagent Systems AAMAS, 2003.
24. C. Baral, M. Gelfond: Logic programming and knowledge representation. *Logic Programming*, 1994.
25. A. Barros, M. Dumas, P. Oaks: A Critical Overview of the Web Services Choreography Description Language (WS-CDL). BPTrends Newsletter ([www.bptrends.com](http://www.bptrends.com)), 2005.
26. U. Basters, M. Klusch: RS2D: Fast Adaptive Search for Semantic Web Services in Unstructured P2P Networks. Proceedings of 5th International Semantic Web Conference (ISWC), Athens, USA, Lecture Notes in Computer Science (LNCS), 4273:87-100, Springer, 2006.
27. S. Bechhofer et al.: DIG 2.0 Towards a flexible interface for Description Logic reasoners. Proceedings of International Workshop on OWL Experiences and Directions (OWLED), USA, 2006.
28. S.C. Benjamin, P.M. Hayden. Multi-Player Quantum Games. *Physical Review A* 64(3):030301, 2001.
29. D. Berardi, D. Calvanese, G. De Giacomo, M. Lenzerini, M. Mecella: Automatic service composition based on behavioral descriptions. *Cooperative Information Systems*, 14(4), 2005.
30. D. Berardi, D. Calvanese, G. De Giacomo, R. Hull, M. Mecella: Automatic Composition of Transition-based Semantic Web Services with Messaging. Proceedings of 31st International Conference on Very Large Data Bases (VLDB 2005), Trondheim, Norway, 2005.
31. T. Berners-Lee, J. Hendler, O. Lasilla: The Semantic Web. *Scientific American*, May 17, 2001.
32. A. Bernstein, C. Kiefer: Imprecise RDQL: Towards Generic Retrieval in Ontologies Using Similarity Joins. Proceedings ACM Symposium on Applied Computing, Dijon, France, ACM Press, 2006.

33. P. Bertoli, A. Cimatti, P. Traverso: Interleaving Execution and Planning for Nondeterministic, Partially Observable Domains. Proceedings of European Conference on Artificial Intelligence (ECAI), 2004.
34. C. Bettstetter, C. Renner: A comparison of service discovery protocols and implementation of the SLP. Proceedings of the EUNICE Open European Summer School, Twente, Netherlands, Sept 13-15, 2000.
35. D. Bianchini, V. De Antonellis, M. Melchiori, D. Salvi: Semantic-enriched Service Discovery. Proceedings of IEEE ICDE 2nd International Workshop on Challenges in Web Information Retrieval and Integration (WIRI06), Atlanta, Georgia, USA, 2006.
36. W. Binder, I. Constantinescu, B. Faltings, K. Haller, C. Tuerker: A Multi-Agent System for the Reliable Execution of Automatically Composed Ad-hoc Processes. Proceedings of 2nd European Workshop on Multi-Agent Systems (EUMAS), Barcelona, Spain, 2004.
37. **B. Blankenburg, M. He, M. Klusch, N. Jennings: Risk Bounded Formation of Fuzzy Coalitions Among Service Agents. Proceedings 10th Intl. Workshop on Cooperative Information Agents, Edinburgh, UK, Lecture Notes in Artificial Intelligence (LNAI), 4149, Springer, 2006.**
38. **B. Blankenburg and M. Klusch: BSCA-F: Efficient Fuzzy Valued Stable Coalition Forming Among Agents. Proceedings 4th IEEE Conference on Intelligent Agent Technology (IAT), Compiegne, France, IEEE Computer Society Press, 2005.**
39. **B. Blankenburg and M. Klusch: BSCA-P: Privacy Preserving Coalition Forming Among Rational Web Service Agents. *Knstliche Intelligenz*, 1/06:19 - 25, BtcherIT Verlag, 2006.**
40. **B. Blankenburg and M. Klusch: On Safe Kernel Stable Coalition Forming Among Agents. Proceedings 3rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS), New York, USA, pp 580 - 587, ACM Press, 2004.**
41. **B. Blankenburg, M. Klusch, O. Shehory: Fuzzy Kernel-Stable Coalitions Between Rational Agents. Proceedings 2nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Melbourne, Australia, ACM Press, 2003.**
42. B. Blankenburg, L. Botelho, F. Calhau, A. Fernandez, M. Klusch, S. Ossowski: Service Composition. Chapter 11 in [330], 2008.
43. H. Boley, M. Dean, B. Grosz, M. Sintek, B. Spencer, S. Tabet, G. Wagner: FOL RuleML: The first-order logic web language. Available at [www.w3.org/Submission/2005/SUBM-FOL-RuleML-20050411/](http://www.w3.org/Submission/2005/SUBM-FOL-RuleML-20050411/) and [www.ruleml.org/fol/](http://www.ruleml.org/fol/), 2005.
44. H. Boley, M. Kifer, P.-L. Patranjan, A. Polleres: Rule interchange on the Web. Reasoning Web 2007, LNCS 4636, Springer, 2007.
45. L. Botelho, A. Fernandez, M. Klusch, L. Pereira, T. Santos, P. Pais, M. Vasirani: Service Discovery. Chapter 10 in [330], 2008.
46. L. Botelho, A. Lopes, T. Möller, H. Scholdt: Semantic Web Service Execution. Chapter 12 in [330], 2008.
47. D. Bouwmeester, K. Mattle, J.-W. Pan, H. Weinfurter, A. Zeilinger, M. Zukowski: Experimental quantum teleportation of arbitrary quantum states. Applied Physics B: Lasers and Optics, 67(6), December, 1998.

48. D. Bouwmeester, A.K. Ekert, A. Zeilinger (Hrsg.): The Physics of Quantum Information - Quantum Cryptography, Quantum Teleportation, Quantum Computation. Springer, 2000
49. D. Bruss, Gerd Leuchs (eds.): Lectures on Quantum Information. Wiley-VCH, Physics Textbook, 2007.
50. P.D. Bruza, W. Lawless, K. van Rijbergen, D.A. Sofge (Eds.): Quantum interaction. Proceedings of the AAAAI Spring Symposium, Stanford, USA, Technical Report SS-07-08, AAAI Press, 2007.
51. J. Bryson: The Behavior-Oriented Design of Modular Agent Intelligence. Proceedings of Workshop on Agent Technologies, Infrastructures, Tools, and Applications for E-Services, Erfurt, Germany, Lecture Notes in Computer Science (LNCS) vol. 2592, 2003.
52. C. Bussler: The fractal nature of web services. *IEEE Computer*, IEEE Press, March 2007.
53. C. Caceres, A. Fernandez, H. Helin, O. Keller, M. Klusch: Context-aware Service Coordination for Mobile Users. Proceedings IST eHealth Conference, 2006.
54. M. Cadoli, F.M. Donini, A. Schaerf: A survey of complexity results for non-monotonic logics. *Logic Programming*, 17, 1993.
55. D. Calvanese, G. De Giacomo, I. Horrocks, C. Lutz, B. Motik, B. Parsia, P. Patel-Schneider: OWL 1.1 Web Ontology Language Tractable Fragments. W3C Member Submission, 19 December 2006. [www.w3.org/Submission/2006/SUBM-owl11-tractable-20061219/](http://www.w3.org/Submission/2006/SUBM-owl11-tractable-20061219/). Updated version at [www.webont.org/owl/1.1/tractable.html](http://www.webont.org/owl/1.1/tractable.html) (6 April 2007)
56. D. Calvanese, G. De Giacomo, M. Lenzerini: Representing and Reasoning on XML Documents: A Description Logic Approach. *Logic and Computation*, 9(1):295-318, 1999.
57. J. Cardoso, A. Sheth (Eds.): Semantic Web Services: Processes and Applications. Springer book series on Semantic Web & Beyond: Computing for human Experience, 2006.
58. F. Casati, M.C. Shan: Dynamic and Adaptive Composition of E-services. *Information Systems*, 6(3), 2001.
59. CASCOM Project Deliverable D3.2: Conceptual Architecture Design. September 2005. [www.ist-cascom.org](http://www.ist-cascom.org)
60. C.D. Cavanaugh, L.R. Welch, B.A. Shirazi, E. Huh, S. Anwar: Quality of Service Negotiation for Distributed, Dynamic Real-time Systems. Proceedings of Parallel and Distributed Processing IPDPS Workshop, Cancun, Mexico, Lecture Notes in Computer Science (LNCS) vol. 1800, Springer, 2000.
61. D. Chakraborty, A. Joshi: Dynamic service composition: State-of-the-art and research directions. Technical Report TR-CS-01-19, CSEE. University of Maryland, Baltimore County, December 2001.
62. D. Chakraborty, F. Perich, S. Avancha, A. Joshi: DReggie: Semantic Service Discovery for M-Commerce Applications. Proceedings of the International Workshop on Reliable and Secure Applications in Mobile Environment, 2001.
63. D.A. Chappell, T. Jewell: Java Web Services: Using Java in Service-Oriented Architectures. O'Reilly, 2002.
64. H. Chen, A. Joshi, and T. Finin: Dynamic service discovery for mobile computing: Intelligent agents meet JINI in the aether. 4(4):343-354, 2001.
65. K-Y. Chen, T. Hogg, B.A. Huberman: Behavior of Multi-Agent Protocols Using Quantum Entanglement. In [50].

66. W.K. Cheung, X. Zhang, H. F. Wong, J. Liu, Z. Luo, F. Tong: When Distributed Data Mining Goes Service Oriented. *IEEE Internet Computing*, July-August, 2006.
67. M.d. Chiara, R. Giuntini, R. Greechie: Reasoning in Quantum Theory - Sharp and Unsharp Quantum Logics. Springer, Trends in Logic, Band 22, 2004.
68. A. Church: A note on the "Entscheidungsproblem". *Symbolic Logic*, 1:40-41, 1936
69. S. Colucci, T.C. Di Noia, E. Di Sciascio, F.M. Donini, M. Mongiello: Concept Abduction and Contraction for Semantic-based Discovery of Matches and Negotiation Spaces in an E-Marketplace. *Electronic Commerce Research and Applications*, 4(4):345361, 2005.
70. S. Cong, E. Hunt, K.R. Dittrich: IEIP: An inter-enterprise integration platform for e-commerce based on web service mediation. Proceedings of 4th European Conference on Web Services (ECOWS), Zurich, IEEE CS Press, 2006.
71. D. Connolly, F. van Harmelen, I. Horrocks, D. McGuinness, P. Patel-Schneider, L. Stein: DAML+OIL reference description. W3C Note, 18 December 2001. Available at [www.w3.org/TR/2001/NOTE-daml+oil-reference-20011218](http://www.w3.org/TR/2001/NOTE-daml+oil-reference-20011218).
72. I. Constantinescu, B. Faltings: Efficient matchmaking and directory services Proceedings of IEEE Conference on Web Intelligence WI, 2003.
73. J. Costa da Silva, M. Klusch: Inference in Distributed Data Clustering. *Engineering Artificial Intelligence Applications*, 6(3), Elsevier, 2006.
74. J. Costa Da Silva, M. Klusch: Privacy Preserving Pattern Discovery in Distributed Time Series. Proceedings of 3rd International Workshop on Privacy Data Management PDM, Istanbul, Turkey, IEEE CS Press, 2007.
75. J. Costa Da Silva, M. Klusch, S. Lodi, G.L. Moro: Inference attacks in peer-to-peer homogeneous data mining. Proceedings of the European Conference on Artificial Intelligence (ECAI), Valencia, Spain, 2004.
76. **J. Costa da Silva, M. Klusch, S. Lodi, G. Moro: Privacy-preserving agent-based distributed data clustering. *Web Intelligence and Agent Systems*, 4(2):221 - 238, IOS Press, 2006.**
77. J. Contreras, M. Klusch, J. Krawczyk: Numerical Solutions to Nash-Cournot Equilibria in Coupled Constraint Electricity Markets. *IEEE Transactions on Power Systems*, 19(1), IEEE Press, 2004.
78. C. Cumbo, W. Faber, G. Greco, N. Leone: Enhancing the magic-set method for disjunctive datalog programs. Proceedings of 20th International Conference on Logic Programming (ICLP'04), September, 2004, Saint-Malo, France, pp. 371-385
79. E. DHondt, P. Panangaden: Leader Election and Distributed Consensus with Quantum Resources. [www.vub.ac.be/CLEA/ellie/homepage/PDFdirectory/QDA\\_long.pdf](http://www.vub.ac.be/CLEA/ellie/homepage/PDFdirectory/QDA_long.pdf), 2005.
80. E. DHondt, P. Panangaden: Reasoning about quantum knowledge. 25th Conference on Foundations of Software Technology and Theoretical Computer Science, Hyderabad, India, Lecture Notes in Computer Science (LNCS) vol. 3821, Springer, 2005.
81. C.V. Damasio, A. Analyti, G. Antoniou, G. Wagner: Supporting open and closed world reasoning on the Web. Proceeding of Conference on Principles and Practice of Semantic Web Reasoning (PPSWR), LNCS vol. 4187, Springer 2006.
82. J. Dang, M. Huhns: Concurrent multiple-issue negotiation for Internet-based services. *IEEE Internet Computing*, 10(6):42-49, 2006.

83. V. Danos, E. D'Hondt, E. Kashefi, P. Panangaden: Distributed measurement-based quantum computation. Proceedings of the 3rd International Workshop on Quantum Programming Languages (QPL2005), 2005. See also Ellie D'Hondt PhD Thesis (2005) at [http://www.vub.ac.be/CLEA/ellie/homepage/PDFdirectory/PhD\\_DistributedQC.pdf](http://www.vub.ac.be/CLEA/ellie/homepage/PDFdirectory/PhD_DistributedQC.pdf)
84. T. Datz: What You Need to Know About Service-Oriented Architecture. CIO Magazine, January 2004. <http://www.cio.com/archive/011504/soa.html>
85. P.C.W. Davies, J.R. Brown (Eds.). The Ghost in the Atom. Cambridge University Press, Canto edition reprint, 2000.
86. S. Davis, and T. Walton: Engineering Knowledge. Proceedings of the 42nd Annual ACM Southeast Conference, Huntsville, AL, USA, 2004.
87. J. De Bruijn. Ontology Languages Around FOL and LP. WSMO Deliverable D28.3 v0.1 Available at: [www.wsmo.org/TR/d28/d28.3/v0.1/20051015/](http://www.wsmo.org/TR/d28/d28.3/v0.1/20051015/)
88. J. De Bruijn, S. Heymanns: WSMO Ontology Semantics. WSMO Deliverable D28.3 v0.2 Final Available at [www.wsmo.org/TR/d28/d28.3/v0.2/20070416/](http://www.wsmo.org/TR/d28/d28.3/v0.2/20070416/)
89. J. De Bruijn, T. Eiter, A. Polleres, H. Tompits: On representational issues about combinations of classical theories with nonmonotonic rules. DERI Technical Report 2006-05-29, May 2006.
90. G. De Giacomo, M. Lenzerini, A. Poggi, R. Rosati: On the Update of Description Logic Ontologies at the Instance Level. Proceedings of the AAAI Conference, AAAI Press, 2006.
91. R. De Wolf: Quantum Computing and Communication Complexity. Phd thesis, University of Amsterdam, 2001.
92. M. Dean, D. Connolly, F. van Harmelen, J. Hendler, I. Horrocks, D. McGuinness, P. Patel-Schneider, L. Stein: OWL web ontology language reference. W3C Working Draft, 31 March 2003. Available at [www.w3.org/TR/2003/WD-owl-ref-20030331](http://www.w3.org/TR/2003/WD-owl-ref-20030331)
93. R. Dearden, N. Meuleauy, S. Ramakrishnany, D.E. Smith, R. Washington: Incremental Contingency Planning. Proceedings of ICAPS-03 Workshop on Planning under Uncertainty, Trento, Italy, 2003.
94. E. Della Valle, D. Cerizza, I. Celino: The Mediators Centric Approach to Automatic Web Service Discovery of Glue. Proceedings of 1st International Workshop on Mediation in Semantic Web Services (MEDIATE), CEUR Workshop proceedings, 168, 2005.
95. G. Denker, L. Kagal, T. Finin, M. Paolucci, K. Sycara: Security For DAML Web Services: Annotation and Matchmaking. Proceedings of the Second International Semantic Web Conference (ISWC 2003), USA, 2003.
96. D. DeRoure, N.R. Jennings, N. Shadbolt: The semantic grid: past, present, and future. *Proceedings of the IEEE*, 93(3), 2005.
97. D. Deutsch: Quantum Theory, the Church-Turing Principle, and the Universal Quantum Computer. Proceedings of the Royal Society of London, A400, 1985.
98. I. Dickinson, M. Wooldridge: Agents are not (just) web services: Considering BDI agents and web services HPL-2005-123 Technical Report, 2005.
99. T. Di Noia, E.D. Sciascio, F.M. Donini, M. Mogiello: A System for Principled Matchmaking in an Electronic Marketplace. *Electronic Commerce*, 2004.
100. T. Di Noia, E. Di Sciascio, F.M. Donini: Semantic Matchmaking as Non-Monotonic Reasoning: A Description Logic Approach. *Artificial Intelligence Research (JAIR)*, 29:269–307, 2007.

101. J. Domingue, S. Galizia, L. Cabral: Choreography in IRS-III: Coping with Heterogeneous Interaction Patterns in Web Services. Proceedings International Semantic Web Conference, LNAI, Springer, 2005.
102. F.M. Donini, M. Lenzerini, D. Nardi, A. Schaerf: AL-log: Integrating Datalog and description logics. *Intelligent Information Systems*, 10(3), 1998.
103. T. Eiter, G. Ianni, T. Krennwallner, A. Polleres: Rules and Ontologies for the Semantic Web. Reasoning Web Summer School 2008, Venice, Italy, to appear.
104. T. Eiter, A. Polleres: Towards automated integration of guess and check programs in answer set programming: A meta-interpreter and applications. *Theory and Practice of Logic Programming*, 6(1-2):23-60, 2006.
105. T. Eiter, G. Ianni, A. Polleres, R. Schindlauer, H. Tompits: Reasoning with rules and ontologies. Proceedings of REVERSE Summer School: Reasoning Web 2006, LNCS, 4126, pages 93-127, Springer, 2006.
106. T. Eiter, T. Lukasiewicz, R. Schindlauer, H. Tompits: Combining Answer Set Programming with Description Logics for the Semantic Web. Proceedings of International Conference on Principles of Knowledge Representation and Reasoning (KRR), 2004.
107. T. Eiter, T. Lukasiewicz, R. Schindlauer, H. Tompits: Well-founded semantics for description logic programs in the semantic Web. Proceedings of International Conference on RuleML, 2004.
108. H. B. Enderton: A Mathematical Introduction to Logic. Academic Press, 2002
109. O. Etzioni, K. Golden, D. Weld: Tractable closed world reasoning with updates. Proceedings of the 4th International Conference on Principles of Knowledge Representation and Reasoning (KRR), 1994.
110. J. Euzenat, P. Shvaiko: Ontology matching. Springer, 2007
111. D. Fahland, W. Reisig: ASM-based semantics for BPEL: The negative Control Flow. Proceedings of the 12th International Workshop on Abstract State Machines (ASM'05), 2005.
112. A. Fadhil, V. Haarslev: GLOO: A Graphical Query Language for OWL ontologies. Proceedings of Intl Workshop on OWL: Experience and Directions (OWLed) at ISWC 2006, Athens, USA, 2006.
113. D. Fensel, F. van Harmelen: Unifying reasoning and search to Web scale. *IEEE Internet Computing*, March/April 2007.
114. D. Fensel, J. Hendler, H. Lieberman, W. Wahlster: Spinning the Semantic Web. Bringing the World Wide Web to Its Full Potential. MIT Press, 2005.
115. D. Fensel, H. Lausen, A. Polleres, J. de Bruijn, M. Stollberg, D. Roman, J. Domingue: Enabling Semantic Web Services - The Web Service Modeling Ontology. Springer, 2006.
116. A. Fernandez, M. Vasirani, C. Caceres, S. Ossowski: A role-based support mechanism for service description and discovery. In: huang et al. (eds.), Service-Oriented Computing: Agents, Semantics, and Engineering. LNCS 4504, Springer, 2007.
117. R.P. Feynman: Simulating physics with computers. *Theoretical Physics*, 21, 1982.
118. R. Fielding, R. Taylor: Principled Design of the Modern Web Architecture. *ACM Transactions on Internet Technology*, 2(2), 2002.
119. R. Fikes, P. Hayes, I. Horrocks: OWL-QL - a language for deductive query answering on the semantic web. Technical Report KSL-03-14, Knowledge Systems Lab, Stanford University, CA, USA (2003)

120. K. Fischer, C. Ruß, G. Vierke. Decision Theory and Coordination in Multiagent Systems. DFKI Research Report RR-98-02, 1998.
121. A. Fokoue, A. Kershenbaum, L. Ma, E. Schonberg, K. Srinivas: The summary ABox: Cutting ontologies down to size. Proceedings of International Semantic Web Conference ISWC, LNCS 4273, 2006.
122. X. Fu, T. Bultan, J. Su: WSAT: A tool for formal analysis of web services. In R. Alur and D. Peled, editors, Proceedings of the 16th International Conference on Computer Aided Verification, LNCS, vol 3114, Springer, 2004.
123. N. Fuhr, K. Grojohann: XIRQL: An XML Query Language Based on Information Retrieval Concepts. *ACM Transactions on Information Systems*, 22, 2004.
124. U. Furbach, M. Maron, K. Read: Location based informationsystems. *Künstliche Intelligenz*, 3/07, BöttcherIT, 2007.
125. D. Gabbay, C.J. Hogger, J.A. Robinson: Handbook of Logic in Artificial Intelligence and Logic Programming. Vol. 3, Oxford University Press, 1994.
126. A. Gerber: Flexible Kooperation zwischen Autonomen Agenten in Dynamischen Umgebungen. Dissertation, Universität des Saarlandes, Saarbrücken, Germany, 2004.
127. **A. Gerber, M. Klusch: Agent-based Integrated Services Network for Timber Production and Sales. *Intelligent Systems*, 17 (1):32 - 39, IEEE CS Press, Jan/Feb 2002**
128. A. Gerber, N. Kammenhuber, M. Klusch: CASA: A Distributed Holonic Multiagent Architecture for Timber Production. Proceedings of 1st International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), Bologna, Italy, 2002.
129. **A. Gerber, M. Klusch: AGRICOLA: Agenten für mobile Planungsdienste in der Landwirtschaft. *Künstliche Intelligenz*, 1/04:38 - 42, arendtap Verlag, 2004.**
130. M. Ghallab, D. Nau, P. Traverso: Automated planning. Elsevier, 2004.
131. B. Glimm, I. Horrocks, C. Lutz, U. Sattler: Conjunctive Query Answering for the Description Logic SHIQ. Proceedings of International Joint Conference on AI (IJCAI), 2007.
132. C. Goble, D. DeRoure: The Grid: an application of the semantic web *ACM SIGMOD Record*, 31(4), 2002.
133. C. Golbreich: Combining rule and ontology reasoners for the semantic web. Proceedings of 3rd International Workshop on RuleML, Hiroshima, Japan, 2004.
134. H. Goradia, J. Vidal: An equal excess negotiation algorithm for coalition formation. Proceedings of the International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), 2007.
135. H. Goradia, J. Vidal: A distributed algorithm for finding nucleolus-stable payoff divisions. Proceedings of the IEEE/ACM International Conference on Intelligent Agent Technology, 2007.
136. B.C. Grau, B. Parsia, E. Sirin, A. Kalyanpur: Automatic partitioning of OWL ontologies using  $\epsilon$ -connections. Proceedings of the Workshop on Description Logics (DL), 2005.
137. S.D. Gribble, M. Welsh, R. von Behren, E.A. Brewer, D. Culler, N. Borisov, S. Czerwinski, R. Gummadi, J. Hill, A.D. Joseph, R.H. Katz, Z. Mao, S. Ross, and B. Zhao: The NINJA architecture for robust internetscale systems and services. Special Issue of Computer Networks on Pervasive Computing, pages 473-497, March 2001.



138. D. Grigori, V. Peralta, M. Bouzeghoub: Service Retrieval Based on Behavioral Specifications and Quality Requirements. LNCS 3649, Springer, 2005.
139. S. Grimm: Discovery - Identifying relevant services. In [348], 2007.
140. S. Grimm, B. Motik: Closed World Reasoning in the semantic Web through Epistemic Operators. Proceedings of the Workshop OWL - Experiences and Directions, Galway, Ireland, 2005.
141. S. Grimm, B. Motik, C. Preist: Matching semantic service descriptions with local closed-world reasoning. Proceedings of 3rd European Semantic Web Conference (ESWC), Springer, LNCS, 2006.
142. S. Grimm, P. Hitzler, A. Abecker: Knowledge representation and ontologies: Logic, Ontologies and Semantic Web Languages. In: [348] (Chapter 3), 2007.
143. B. Grosz, I. Horrocks, R. Volz, S. Decker: Description Logic Programs: Combining Logic Programs with Description Logic. Proceedings of the 12th International World Wide Web Conference (WWW), 2003.
144. T.R. Gruber: A Translation Approach to Portable Ontology Specification. *Knowledge Acquisition*, 6(2), 1993.
145. N. Guarino: Semantic Matching: Formal Ontology Distinctions for Information Organization, Extraction and Integration. LNCS, 1299, Springer, 1997
146. L. Guo, Yun-He, C. Burger, D. Roberston: Mapping a business process model to a semantic Web service model Proceedings of the IEEE International Conference on Web Services, 2004.
147. L. Guo, F. Shao, C. Botev, J. Shanmugasundaram: XRANK: Ranked Keyword Search over XML Documents. Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, San Diego, USA, 2003.
148. P. Haase, R. Siebes, F. van Harmelen: Expertise-based Peer selection in Peer-to-Peer Networks. *Knowledge and Information Systems*, Springer, 2006
149. C. Halashek-Wiener, B. Parsia, E. Sirin: Towards Continuous Query Answering on the Semantic Web. Proceedings of 2nd Intl. Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS 2006), 2006.
150. R. Hamadi, B. Benatallah: A Petri-Net Based Model for Web Service Composition. Proceedings 14th Australian Conference on Database Technologies, ACM Press, 2003.
151. P. Hayes. RDF Semantics. [www.w3.org/TR/rdf-mt/](http://www.w3.org/TR/rdf-mt/).
152. B. He, M. Patel, Z. Zhang, K.Chang: Accessing the Deep Web. *Communications of the ACM*, 50(5), 2007.
153. L. Henoque, M. Kleiner: Composition - Combining Web Service Functionality in Composite Orchestrations. Chapter 9 in [348], 2007.
154. P. Hitzler, M. Krotzsch, S. Rudolph, Y. Sure: Semantic Web - Grundlagen. Springer, 2008.
155. J. Hoffmann, R. Brafman: Conformant Planning via Heuristic Forward Search: A New Approach. *Artificial Intelligence*, 170(6-7), 2006.
156. J. Hoffmann, B. Nebel: The FF Planning System: Fast Plan Generation Through Heuristic Search, *Artificial Intelligence Research*, 14, 2001.
157. M.J. Holler, G. Illing: Einföhrung in die Spieltheorie. 4. Auflage, Springer, 2000.
158. I. Horrocks, P. Patel-Schneider: Optimizing description logic subsumption. *Logic and Computation*, 9(3):267293, 1999.
159. I. Horrocks, P. Patel-Schneider: Reducing OWL entailment to description logic satisfiability. Proceedings of International Semantic Web Conference (ISWC), 2003, Springer, LNCS, 2870, 2003.

160. I. Horrocks, P. Patel-Schneider: A proposal for an OWL rules language. Proceedings of 13th International World Wide Web Conference (WWW), 2004.
161. I. Horrocks, P. Patel-Schneider, F. van Harmelen: From SHIQ and RDF to OWL: The Making of a Web Ontology Language. *Web Semantics*, 1, Elsevier, 2004.
162. I. Horrocks, B. Parsia, P. Patel-Schneider, J. Hendler: Semantic web architecture: Stack or two towers? Principles and Practice of Semantic Web Reasoning (PPSWR 2005), LNCS 3703, Springer, 2005.
163. I. Horrocks, U. Sattler: A Tableaux Decision Procedure for SHOIQ. Proceedings of 19th International Joint Conference on Artificial Intelligence (IJCAI), Scotland, UK, 2005.
164. I. Horrocks, U. Sattler, S. Tobies: Practical Reasoning for Very Expressive Description Logics. *Logic Journal of the IGPL*, 8(3):239263, 2000.
165. I. Horrocks and S. Tessaris: A conjunctive query language for description logic ABoxes. Proceedings of the AAAI Conference, 2000.
166. R.A. Howard: Information value theory. *IEEE Trans. on Systems Sciences and Cybernetics*, 2, 1966.
167. M.-C. Hsu, P. Hsueh-Min Chang, Y.-M. Wang, V. Won Soo Multi-agent Travel Planning through Coalition and Negotiation in an Auction. LNCS, 2891, Springer, 2003.
168. B.A. Huberman, Tad Hogg: Quantum Solution of Coordination Problems. *Quantum Information Processing*, 2(6), December 2003.
169. M.N. Huhns: Agents as Web Services. *IEEE Internet Computing*, 6 (4), 2002.
170. D. Hull, U. Sattler, E. Zolin, R. Stevens, A. Bovykin, I. Horrocks: Deciding semantic matching of stateless services. Proceedings of 21st National Conference on Artificial Intelligence (AAAI), AAAI Press, 2006
171. P.C.K. Hung, H. Li, J.-J. Jeng: Web service negotiation: An overview of research issues. Proceedings of 37th Annual Hawaii International Conference on System Sciences (HICSS), Washington DC, USA, IEEE CS Press, 2004.
172. U. Hustadt, B. Motik, U. Sattler: Reducing SHIQ Description Logic to Disjunctive Datalog Programs. Proceedings of 9th International Conference on Knowledge Representation and Reasoning (KR2004), Whistler, Canada, 2004.
173. U. Hustadt, B. Motik, U. Sattler: Data complexity of reasoning in very expressive description logics. Proceedings of 19th Intl Joint Conference on Artificial Intelligence IJCAI, 2005.
174. D. Hutter, M. Klusch, M. Volkamer: Information Flow Analysis Based Security Checking of Health Service Composition Plans. Proceedings of 1st European Conference on eHealth, Fribourg, Switzerland, 2006.
175. D. Hutter, M. Volkamer, M. Klusch, A. Gerber: Provably Secure Execution of Composed Semantic Web Services. Proceedings of 1st International Workshop on Privacy and Security in Agent-based Collaborative Environments (PSACE 2006), Hakodate, Japan, 2006.
176. S. Jacobi, C. Madrigal-Mora, E. Leon-Soto, K. Fischer: AgentSteel: An agent-based Online System for the Planning and Observation of Steel Production. Proceedings 4th Intl. Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), Utrecht, The Netherlands, ACM Press, 2005
177. M.C. Jaeger, G. Rojec-Goldmann, C. Liebetruhl, G. M<sup>n</sup>uhl, K. Geihs: Ranked Matching for Service Descriptions Using OWL-S. Proceedings of 14. GI/VDE Fachtagung Kommunikation in Verteilten Systemen KiVS, Kaiserslautern, 2005

178. C.M. Jonker, V. Robu, J. Treur: An agent architecture for multi-attribute negotiation using incomplete preference information. *Autonomous Agents and Multi-Agent Systems*, 15(2):221–252, 2007.
179. L. Kagal, T. Finin, M. Paolucci, N. Srinivasan, K. Sycara, G. Denker: Authorization and Privacy for Semantic Web Services. *IEEE Intelligent Systems*, July/August, 2004.
180. Kahan, Rapoport: Theories of coalition formation. LEA Publisher. 1984.
181. F. B. Kashani, C.C. Shen, C. Shahabi: SWPDS: Web service peer-to-peer discovery service. Proceedings of Intl. Conference on Internet Computing, 2004.
182. **F. Kaufer and M. Klusch: WSMO-MX: A Logic Programming Based Hybrid Service Matchmaker. Proceedings 4th IEEE European Conference on Web Services (ECOWS), Zurich, Switzerland, IEEE CS Press, 2006.**
183. F. Kaufer, M. Klusch: Performance of Hybrid WSML Service Matching with WSMO-MX: Preliminary Results. Proceedings of 1st International Joint Workshop on Semantic Matchmaking and Resource Retrieval, Busan, Korea, CEUR vol. 243, 2007.
184. H.A. Kautz, B.Selman: Hard problems for simple default logics. *Artificial Intelligence*, 49, 1991.
185. U. Keller, R. Lara, H. Lausen, A. Polleres, D. Fensel: Automatic Location of Services. Proceedings of the 2nd European Semantic Web Conference (ESWC), Heraklion, Crete, LNCS 3532, Springer, 2005.
186. C. Kiefer, A. Bernstein: The Creation and Evaluation of iSPARQL Strategies for Matchmaking. Proceedings of European Semantic Web Conference, Springer, 2008.
187. M. Kifer, G. Lausen, J. Wu: Logical Foundations of Object-Oriented and Frame-Based Languages. *Journal of the ACM*, 42(4), 1995.
188. T. Kleemann, A. Sinner: Description logic based matchmaking on mobile devices. Proceedings of 1st Workshop on Knowledge Engineering and Software Engineering (KESE 2005), 2005.
189. M. Klein, B. König-Ries: Coupled Signature and Specification Matching for Automatic Service Binding. European Conference on Web Services (ECOWS 2004), Erfurt, 2004.
190. M. Klusch: On Agent Based Semantic Service Coordination - Addendum. Available at: [www.dfki.de/~klusch/habil-aux.pdf](http://www.dfki.de/~klusch/habil-aux.pdf)
191. M. Klusch: Kooperative Informationsagenten im Internet. Dissertation, Christian-Albrechts-University of Kiel, Germany, Kovacs Verlag, Forschungsergebnisse aus der Informatik, 1998.
192. M. Klusch (ed.): Intelligent Information Agents - Agent-Based Information Discovery and Management on the Internet. Springer, 1999.
193. M. Klusch: Information Agent Technology for the Internet: A Survey. *Data and Knowledge Engineering*, 36(3), Elsevier Science, 2001.
194. **M. Klusch: Toward Quantum Computational Agents. In: Computational Autonomy and Agents. M Nickles, M Rovatsos, G Weiss (eds.), LNAI, 2969, Springer, 2004**
195. M. Klusch: Semantic Web Service Description. In M. Schumacher, H. Hellin (Eds.): CASCOM - Intelligent Service Coordination in the Semantic Web. Chapter 3. Birkhäuser Verlag, Springer, 2008.

196. M. Klusch: Semantic Web Service Coordination. In M. Schumacher, H. Hellin (Eds.): CASCOM - Intelligent Service Coordination in the Semantic Web. Chapter 4. Birkh"auser Verlag, Springer, 2008.
197. **M. Klusch, U. Basters: Risk Driven Semantic P2P Service Retrieval. Proceedings of the 6th IEEE International Conference on P2P Computing (P2P 2006), Cambridge, UK, IEEE CS Press, 2006.** See also [26].
198. M. Klusch, S. Bergamaschi, P. Edwards, P. Petta (eds.): Intelligent Information Agents: The AgentLink Perspective. LNAI 2586, Springer, 2003.
199. M. Klusch, B. Fries: Hybrid OWL-S Service Retrieval with OWLS-MX: Benefits and Pitfalls. Proceedings 1st International Joint Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web (SMR2), Busan, Korea, CEUR vol. 243, 2007.
200. M. Klusch, P. Kapahnke, B. Fries: Hybrid Semantic Web Service Retrieval: A Case Study With OWLS-MX. Proceedings of 2nd IEEE Internataional Conference on Semantic Computing (ICSC), Santa Clara, USA, IEEE Press, 2008.
201. **M. Klusch, B. Fries, K. Sycara: Automated Semantic Web Service Discovery with OWLS-MX. Proceedings 5th International. Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), Hakodate, Japan, ACM Press, 2006.**
202. M. Klusch, B. Fries, M. Khalid, K. Sycara: OWLS-MX: Hybrid OWL-S Service Matchmaking. Proceedings of AAAI Fall Symposium on Semantic Web and Agents, Arlington, Washington DC, USA, 2005.
203. M. Klusch, A. Gerber: An agent-based mobile e-commerce service platform for forestry and agriculture. Proceedings of the 2nd Asia-Pacific Conference on Intelligent Agent Technology (IAT), Japan, 2001.
204. **M. Klusch, A. Gerber: Dynamic Coalition Formation among Rational Agents. *Intelligent Systems*, 17(3), IEEE CS Press, May/June 2002**
205. **M. Klusch, A. Gerber: Fast Composition Planning of OWL-S Services and Application. Proceedings of 4th IEEE European Conference on Web Services (ECOWS), Zurich, Switzerland, IEEE CS Press, 2006.**
206. **M. Klusch, A. Gerber, M. Schmidt: Semantic Web Service Composition Planning with OWLS-XPlan. Proceedings 1st International AAAI Fall Symposium on Agents and the Semantic Web, Arlington VA, USA, AAAI Press, 2005.**
207. M. Klusch, F. Kaufer: Hybrid Semantic WSMML Service Matchmaking with WSMO-MX. *Web Intelligence and Agent Systems*, IOS Press, 2008.
208. M. Klusch, S. Lodi, G. Moro: Distributed Clustering Based on Sampling Local Density Estimates. Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Acapulco, Mexico, 2003.
209. M. Klusch, S. Lodi, G.L. Moro: Issues of Agent-Based Distributed Data Mining. Proceedings of the 2nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Melbourne, Australia, 2003.
210. **M. Klusch, K-U. Renner: Dynamic Re-Planning of Composite OWL-S Services. Proceedings of 1st IEEE Workshop on Semantic Web Service Composition, Hongkong, China, IEEE CS Press, 2006.**

211. M. Klusch, O. Shehory: A polynomial kernel-oriented coalition algorithm for rational information agents. Proceedings of 2nd International Conference on Multi-Agent Systems (ICMAS), AAAI Press, 1996.
212. M. Klusch, K. Sycara: Brokering and Matchmaking for Coordination of Agent Societies: A Survey. In: Coordination of Internet Agents, A. Omicini et al. (eds.), chapter 8, Springer, 2001.
213. M. Klusch, Z. Xing: Semantic Web Service in the Web: A Preliminary Reality Check. Proceedings of 1st International Joint Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web (SMR2), Busan, Korea, CEUR vol. 243, 2007.
214. J. Koistinen, A. Seetharaman: Worth-Based Multi-Category Quality-of-Service Negotiation in Distributed Object Infrastructures Hewlett-Packard Technical Report HPL-98-51R1, 1998.
215. V. Kolovski, B. Parsia, Y. Katz, J. Hendler: Representing Web services policies in OWL-DL. Proceedings of 4th International Semantic Web Conference ISWC, 2005.
216. H. Konishi, D. Ray: Coalition formation as a dynamic process. *Economic Theory*, 110, 2003.
217. R. Kontchakov, A. Kurucz, M. Zakharyashev: Undecidability of first-order intuitionistic and modal logics with two variables. *Bulletin of Symbolic Logic*, 11(3):428-438, 2005
218. E. Kovacs: Vermittlung und Management von Diensten in offenen Systemen. Dissertation, Forschungsergebnisse zur Informatik, Band 53, Kovacs Verlag, 1999.
219. S. Kraus: Strategic Negotiation in Multiagent Environments. MIT Press, Cambridge MA, USA, 2001.
220. S. Kraus, O. Shehory, G. Taase: Coalition formation with uncertain heterogeneous information. Proceedings of 2nd international joint conference on Autonomous Agents and Multiagent Systems (AAMAS), USA, ACM Press, 2003.
221. S. Kraus, O. Shehory, G. Taase: The advantages of compromising in coalition formation with incomplete information. Proceedings of 3rd international joint conference on Autonomous Agents and Multiagent Systems (AAMAS), USA, IEEE CS Press, 2004.
222. M. Kroetzsch, S. Rudolph, P. Hitzler: On the complexity of Horn description logics. Proceedings of 2nd Workshop on OWL Experiences and Directions, 2006.
223. P. Kungas, M. Matskin: Semantic Web Service Composition through a P2P-Based Multi-Agent Environment. Proceedings of 4th International Workshop on Agents and Peer-to-Peer Computing (in conjunction with AAMAS 2005), Utrecht, Netherlands, LNCS 4118, 2006.
224. U. Kuster, B. König-Ries, M. Stern, M. Klein: DIANE: An Integrated Approach to Automated Service Discovery, Matchmaking and Composition. Proceedings of the World Wide Web Conference WWW, Banff, Canada, ACM Press, 2007.
225. O. Kutz, I. Horrocks, U. Sattler: The even more irresistible SROIQ. Proceedings of 10th International Conference on Principles of Knowledge Representation and Reasoning, Lake District, UK, 2006.
226. P. Lambrix, H. Tan: SAMBO - A System for Aligning and Merging Biomedical Ontologies. *Web Semantics*, 4(3), 2006.

227. P. Lambrix, H. Tan: A Method for Recommending Ontology Alignment Strategies. Proceedings of 6th International Semantic Web Conference, Busan, Korea. Springer, LNCS, 4825, 2007.
228. S. Lamarter, A. Ankolekar: Automated Selection of Configurable Web Services. 8. Internationale Tagung Wirtschaftsinformatik. Universitätsverlag Karlsruhe, Karlsruhe, Germany, March 2007.
229. F. Lecue, A. Leger: Semantic Web service composition through a matchmaking of domain. Proceedings of 4th IEEE European Conference on Web Services (ECWS), Zurich, 2006.
230. F. Lecue, A. Delteil, A. Leger: Applying Abduction in Semantic Web Service Composition. Proceedings of IEEE International Conference on Web Services (ICWS 2007), 2007.
231. A. Levy, M.-C. Rousset: Combining Horn rules and description logics in CARIN. *Artificial Intelligence*, 104, 1998.
232. L. Li, I. Horrocks: A software framework for matchmaking based on semantic Web technology. Proceedings of the World Wide Web conference (WWW), Budapest, 2003.
233. C. Li, U. Rajan, S. Chawla, K. Sycara: Mechanisms for coalition formation and cost sharing in an electronic marketplace. Proceedings of the 5th international conference on Electronic commerce (ICEC), New York, NY, USA, ACM Press, 2003.
234. V. Lifschitz: Nonmonotonic databases and epistemic queries. Proceedings of the International Joint Conference on AI (IJCAI), Sydney, Australia, 1991.
235. H. Liu, C. Lutz, M. Milicic, F. Wolter: Updating Description Logic ABoxes. Proceedings of the International Conference on Knowledge Representation and Reasoning (KRR), 2006.
236. J. Liu, H. Zhuge: A Semantic-Link-Based Infrastructure for Web Service. Proceedings of the International World Wide Web Conference, 2005.
237. S. Liu, P. Kungas, M. Matskin: Agent-Based Web Service Composition with JADE and JXTA. Proceedings of International Conference on Semantic Web and Web Services (SWWS), Las Vegas, USA, 2006.
238. S. Lloyd: Ultimate physical limits to computation. *Nature*, 406, 2000. See also: S. Lloyd: Programming the Universe: A Quantum Computer Scientist Takes On the Cosmos. Alfred A. Knopf Publisher, New York, 2006.
239. S. Lloyd: Deutsch vs. Lloyd, Brain Tennis, Hotwired, July 1997. Available at: [http://web.mit.edu/slloyd/Public/Deutsch-Lloyd\\_Many\\_Worlds\\_Debate.pdf](http://web.mit.edu/slloyd/Public/Deutsch-Lloyd_Many_Worlds_Debate.pdf)
240. S. Lloyd, J.H. Shapiro, F.N.C. Wong, P. Kumar, S.M. Shahriar, H.P. Yuen: Infrastructure for the quantum internet. *Computer Communication Review*, 34(5): 9-20, 2004.
241. B.T. Loo, R. Huebsch, I. Stoica, J.M. Hellerstein: The Case for a Hybrid P2P Search Infrastructure. Proceedings of International Workshop on P2P Systems (IPTPS), USA, Springer, LNCS, 2004.
242. A. Löser, C. Tempich, B. Quilitz, W.-T. Balke, S. Staab, W. Nejdl: Searching Dynamic Communities with Personal Indexes. Proceedings of the International Semantic Web Conference, 2005.
243. N. Lohmann: A Feature-Complete Petri Net Semantics for WS-BPEL 2.0. Proceedings of the International Workshop on Formal Approaches to Business Processes and Web Services (FABPWS'07), 2007.

244. P. Lord, P. Alper, C. Wroe, C. Goble: Feta: A Light-Weight Architecture for User Oriented Semantic Service Discovery. Proceedings of 2nd European Semantic Web Conference (ESWC), Heraklion, Crete, LNCS vol. 3532, Springer, 2005.
245. Q. Lu, P. Cao, E. Cohen, K. Li, S. Shenker: Search and Replication in Unstructured Peer-to-Peer Networks. Proceedings of 6th ACM International Conference on Supercomputing (ICS), New York, USA, 2002.
246. M. Luck, P. McBurney, O. Shehory, S. Willmott: Agent Technology: Computing as Interaction - A Roadmap for Agent Based Computing. AgentLink, ISBN 085432 845 9, 2005.
247. T. Madhusudan, N. Uttamsingh: A declarative approach to composing Web services in dynamic environments. *Decision Support Systems*, 41(2):325357, 2006.
248. P. Marcer, P. Rowlands: How Intelligence Evolved? In [50]. 2007.
249. M. Mares: Fuzzy cooperative games: cooperation with vague expectations. Studies in fuzziness and soft computing, vol 72, Physica Verlag, 2001.
250. A. Martens: Analyzing Web Service based Business Processes. Proceedings of International Workshop on Fundamental Approaches to Software Engineering (FASE), 2005.
251. N. Matos, C. Sierra: Evolutionary computing and negotiating agents. Proceedings of International Workshop on Agent-Mediated Trading, LNAI vol. 1571, Springer, 1998.
252. S. McIlraith, T.C. Son: Adapting Golog for composition of semantic Web services. Proceedings of International Conference on Knowledge Representation and Reasoning (KRR), Toulouse, France, 2002.
253. B.T. Messmer: New approaches on graph matching. PhD Thesis, University of Bern, Switzerland, 1995.
254. M. Mecella, F.P. Presicce, B. Pernici: Modeling E-service Orchestration through Petri Nets. LNCS vol. 2444, Springer, 2002.
255. B. Medjahed, A. Bouguettyaya, A.K. Elmagarmid: Composing Web services on the semantic Web. *Very Large Data Bases (VLDB)*, 12(4), 2003.
256. J. Mei, H. Boley: Interpreting SWRL Rules in RDF Graphs. Electronic Notes in Theoretical Computer Science, 151, Elsevier, 2006.
257. M. Milicic: Planning in Action Formalisms based on DLS: First Results. Proceedings of the International Workshop on Description Logics (DL), 2007.
258. D.S. Milojevic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, Z. Xu: Peer-to-peer computing. Technical Report HPL-2002-57, Hewlett-Packard, 2002.
259. R.C. Moore: Semantical considerations on nonmonotonic logic. *Artificial Intelligence*, 25, 1985.
260. B. Motik: Reasoning in Description Logics using Resolution and DEductive Databases. PhD Thesis, University of Karlsruhe, Germany, 2006.
261. B. Motik, I. Horrocks, U. Sattler: Adding Integrity Constraints to OWL. Proceedings of 3rd Workshop on OWL Experiences and Directions, CEUR vol. 258, 2007.
262. B. Motik, U. Sattler, R. Studer: Query answering for OWL-DL with rules. Proceedings of 3rd International Semantic Web Conference (ISWC), Springer, LNCS vol. 3532, 2004.
263. B. Motik, I. Horrocks, R. Rosati, U. Sattler: Can OWL and Logic Programming live together happily ever after? Proceedings of 5th International Semantic Web Conference (ISWC), Athens, USA, Springer, LNCS vol. 4273, 2006

264. B. Motik, R. Rosati: Closing semantic Web ontologies. Technical report, U Manchester, UK, URL: [www.cs.man.ac.uk/~bmotik/publications/papers/mr06closing-report.pdf](http://www.cs.man.ac.uk/~bmotik/publications/papers/mr06closing-report.pdf), 2006
265. T. Möller, H. Schuldt, A. Gerber, M. Klusch: **Next Generation Applications in Healthcare Digital Libraries using Semantic Service Composition and Coordination**. *Health Informatics*, 12 (2):107-119, SAGE publications, 2006
266. H. Moulin, S. Shenker: Strategyproof sharing of submodular costs: Budget balance versus efficiency. *Economic Theory*, 1997.
267. I. Müller, P. Braun, R. Kowalczyk: A Classification Scheme for the Integration of Software Agent and Service Oriented Paradigms. Proceedings of International Workshop on Service-Oriented Computing and Agent-Based Engineering (SOCABE), 2005.
268. I. Müller, R. Kowalczyk, P. Braun: Towards Agent-Based Coalition Formation for Service Composition. Proceedings of the IEEE International Conference on Intelligent Agent Technology (IAT), Washington, USA, 2006.
269. R.B. Myerson, M.A. Satterthwaite: Efficient mechanisms for bilateral trading. *Economic Theory*, 29(2), 1983.
270. S. Nakajima: Model-checking of safety and security aspects in web service flows. Proceedings of 4th International Conference on Web Engineering, Munich, Germany, LNCS, vol 3140, Springer, 2004
271. S. Narayanan, S. McIlraith: Simulation, verification and automated composition of web services. Proceedings of 11th International Conference on the World Wide Web, New York, ACM Press, 2002.
272. D. S. Nau, T. C. Au, O. Ilghami, U. Kuter, J.W. Murdock, D. Wu, F. Yaman: SHOP2: An HTN planning system. *Artificial Intelligence Research*, 20, 2003.
273. S. Nesbigall: Quantenbasierte Koordination von Multiagentensystemen. Master Thesis, Computer Science Department, University of the Saarland, 2008.
274. X.T. Nguyen, R. Kowalczyk, M.B. Chhetri, A. Grant: WS2JADE: A tool for run-time deployment and control of web services. In: [365].
275. M.A. Nielsen, I.L. Chuang. Quantum Computation and Quantum Information. Cambridge University Press, Cambridge, UK, 2000.
276. R. Nieuwenhuis, A. Rubio: Theorem Proving with Ordering and Equality Constrained Clauses. *Symbolic Computation*, 19:4, 1995.
277. H. Nottelmann, N. Fuhr: Adding probabilities and rules to OWL Lite subsets based on probabilistic Datalog. *Uncertainty, Fuzziness and Knowledge Based Systems*, 14(1), World Scientific, 2006.
278. D. Olawsky, and M. Gini: Deferred planning and sensor use. Proceedings of the DARPA Workshop on Innovative Approaches to Planning, Scheduling and Control, 1990.
279. N. Oldham, K. Verma, A. Sheth, F. Hakimpour: Semantic WS-agreement partner selection. Proceedings of the 15th International Conference on World Wide Web, Edinburgh, UK, ACM Press, 2006.
280. A. Omicini, F. Zambonelli, M. Klusch, R. Tolksdorf: Coordination of Internet Agents: Models, Technologies and Applications. Springer, 2001.
281. S. Ortiz: Getting on board the enterprise service bus. *IEEE Computer*, April 1007.
282. M.J. Osborne, A. Rubinstein: A Course in Game Theory. MIT Press, Cambridge MA, USA, 1994.



283. J. Pan, I. Horrocks: RDFS(FA): Connecting RDF(S) and OWL DL. *IEEE Transactions on Knowledge and Data Engineering*, 19(2):192-206, 2007.
284. M. Papazoglou: Web Services: Principles and Technology. Pearson - Prentice Hall, September 2007.
285. W. Park: Dynamische und vertrauensbasierte Koalitionsbildung zwischen Informationsagenten. Diplomarbeit. Universität des Saarlandes, Saarbrücken, Germany, 2004.
286. M. Paolucci, T. Kawamura, T.R. Payne, K. Sycara: Semantic Matching of Web Services Capabilities. Proceedings of 1st International Semantic Web Conference (ISWC), 2002.
287. M. Paolucci, K. Sycara, T. Nishimara, N. Srinivasan: Using DAML-S for P2P Discovery. Proceedings of International Conference on Web Services (ICWS), Erfurt, Germany, 2003.
288. M. Papazoglou, G. Schlageter: Cooperative Information Systems: Trends and Directions. Academic Press, 1998.
289. M. Papazoglou, P. Traverso, s. Dustdar, F. Leymann: Service-Oriented Computing: State of the Art and REsearch Challenges. *IEEE Computing*, November, pp 38-45, 2007.
290. J. Peer: Web Service Composition as AI Planning: A Survey. Technical Report, University of St. Gallen, Switzerland, 2005. Available at [elektra.mcm.unisg.ch/pbwsc/docs/pfwsc.pdf](http://elektra.mcm.unisg.ch/pbwsc/docs/pfwsc.pdf)
291. J. Peer: A POP-Based Replanning Agent for Automatic Web Service Composition. Proceedings of the 2nd European Semantic Web Conference (ESWC), Heraklion, Crete, LNCS vol. 3532, Springer, 2005.
292. C. Peltz: Web services orchestration: A review of emerging technologies, tools, and standards. Hewlett-Packard, 2003.
293. R. Penrose: The Large, the Small, and the Human Mind. Cambridge University Press, 1997.
294. A. Pfalzgraf: Ein robustes System zur automatischen Komposition semantischer Web Services in SmartWeb. Master Thesis, University of the Saarland, Saarbrücken, Germany, Juni 2006.
295. M. Pistore, P. Traverso: Planning as model checking for extended goals in non-deterministic domains. In: Proceedings of 7th International Joint Conference on Artificial Intelligence (IJCAI), 2001.
296. M. Pistore, P. Traverso, P. Bertoli, A. Marconi: Automated synthesis of composite BPEL4WS web services. Proceedings of the IEEE International Conference on Web Services (ICWS), Orlando, USA, IEEE Press, 2005.
297. M. Pistore, P. Roberti, P. Traverso: Process-Level Composition of Executable Web Services: On-the-fly Versus Once-for-all Composition Proceedings of 2nd European Semantic Web Conference (ESWC), Heraklion, Crete, LNCS vol. 3532, Springer, 2005.
298. J. Polkinghorne: Quantum Theory: A Very Short Introduction. Oxford University Press, 2002.
299. C. Preist: Semantic Web Services - Goals and Vision. Chapter 6 in [348], 2007.
300. C. Preist, C. Bartolini, A. Byde: Agent-based service composition through simultaneous negotiation in forward and reverse auctions. Proceedings of 4th ACM International Conference on Electronic Commerce, San Diego, California, USA, 2003.

301. C. Preist, A. Byde, C. Bartolini, G. Piccinelli: Towards agent-based service composition through negotiation in multiple auctions. *The Interdisciplinary Journal of Artificial Intelligence and the Simulation of Behaviour*, (1):109–124, 2001.
302. T. C. Przymusiński: On the declarative and procedural semantics of logic programs. *Automated Reasoning*, 5(2):167-205, 1989.
303. E. Rahm, P. Bernstein: A Survey of Approaches to Automatic Schema Matching. *The VLDB Journal*, 2001.
304. I. Rahwan, R. Kowalczyk, H.H. Pham: Intelligent agents for automated one-to-many e-commerce negotiation. *Australian Computer Science Communication*, 24(1):197–204, 2002.
305. T. Rahwan, S.D. Ramchurn, V.D. Dang, A. Giovannucci, N.R. Jennings: Anytime optimal coalition structure generation. Proceedings of National Conference on Artificial Intelligence (AAAI), 2007.
306. T. Rahwan, S.D. Ramchurn, V.D. Dang, A. Giovannucci, N.R. Jennings: Near-optimal anytime coalition structure generation. Proceedings of International Joint Conference on Artificial Intelligence (IJCAI), 2007.
307. B. Raman, S. Agarwal, Y. Chen, M. Caesar, W. Cui, P. Johansson, K. Lai, T. Lavian, S. Machiraju, Z.M. Mao, G. Porter, T. Roscoe, M. Seshadri, J. Shih, K. Sklower, L. Subramanian, T. Suzuki, S. Zhuang, A.D. Joseph, R.H. Katz, and I. Stoica: The SAHARA model for service composition across multiple providers. In: Pervasive Computing. LNCS 2414, Springer, August 2002.
308. S.D. Ramchurn, D. Huynh, N. Jennings: Trust in multi-agent systems. *The Knowledge Engineering Review*, 19(1), 2004.
309. J. Rao, P. Kuengas, M. Matskin: Composition of Semantic Web services using Linear Logic theorem proving. *Information Systems*, 31, 2006.
310. K.-U. Renner, P. Kapahnke, B. Blankenburg, M. Klusch: OWLS-XPlan 2.0 - A Dynamic OWL-S Service Composition Planner. BMB+F project SCALLOPS, Internal Project Report, DFKI Saarbrücken, Germany, 2007. [www.dfki.de/~klusch/owls-xplan2-report-2007.pdf](http://www.dfki.de/~klusch/owls-xplan2-report-2007.pdf)
311. W. Reisig: Modeling- and analysis techniques for web services and business processes. Proceedings of the 7th IFIP WG 6.1 International Conference on Formal Methods for Open Object-Based Distributed Systems (FMOODS05), Athens, Greece, LNCS, 3535, Springer, 2005.
312. R. Reiter: On closed world data bases. In H. Gallaire, J. Minker (eds): *Logic and Data Bases*, Plenum Publisher, 1978.
313. S. Russell, P. Norvig: *Artificial Intelligence: A Modern Approach*. Prentice-Hall, 2002.
314. S. Russel: Readings about the question: It is said that reasoning on the semantic web must be monotonic. Why is this so, when human reasoning, which seems to have served us well, is nonmonotonic? Compiled and Last updated on January 31, 2003. Available at: <http://robustai.net/papers/Monotonic Reasoning on the Semantic Web.html>
315. R.T. Rust, C. Mu: What academic research tells us about service. *Communications of the ACM*, 49(7), Special section on services science, ACM Press, July 2006.
316. R. Rosati: DL+lo: Tight integration of description logics and disjunctive Datalog. Proceedings of 10th Intl Conference on Principles of Knowledge Representation and Reasoning (KRR), UK, AAAI Press, 2006.

317. R. Rosati: The limits and possibilities of combining description logics and Datalog. Slides of invited talk given at RuleML 2006, Athens, GA, USA, November 2006. Available from the author rosati at dis.uniroma1.it.
318. A. Rosenfeld, C. Goldman, G. Kaminka, S. Kraus: An Agent Architecture for Hybrid P2P Free-Text Search. Proceedings of 11th Intl Workshop on COoperative Information Agents (CIA), Delft, Springer, LNAI 4676, 2007.
319. J. Rosenschein, G. Zlotkin: Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers. MIT Press, Cambridge MA, USA, 1994.
320. S. Rudolph, M. Kroetzsch, P. Hitzler, M. Sintek, D. Vrandečić: Efficient OWL reasoning with logic program - Evaluations. 2007.
321. J. Sabater, C. Sierra: Review on computational trust and reputation models. *Artificial Intelligence Review*, 24(1), 2005.
322. G. Salaün, L. Bordeaux, M. Schaerf. Describing and reasoning on web services using process algebra. Proceedings of the IEEE International Conference on Web Services, San Diego, USA, IEEE Press, 2004.
323. T. Sandholm: Distributed rational decision making. In G. Weiss (ed.): Multi-agent Systems, Chapter 5, MIT Press, 1999.
324. T. Sandholm: An implementation of the contract net protocol based on marginal cost calculations. Proceedings of the 12th International Workshop on Distributed Artificial Intelligence, Pennsylvania, 1993.
325. T. Sandholm, V. Lesser: Leveled commitment contracts and strategic breach. *Games and Economic Behavior*, 35(1-2), 2001.
326. T. Sandholm: Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1-2):1-54, 2002.
327. T. Sandholm, V. Lesser: Coalition formation among bounded rational agents. Proceedings of International Joint Conference on AI (IJCAI), Morgan Kaufman, San Francisco CA, USA, 1995.
328. M. Schlosser, M. Sintek, S. Decker, W. Nejdl: A Scalable and Ontology-based P2P Infrastructure for Semantic Web Services. Proceedings of 2nd IEEE Intl Conference on Peer-to-Peer Computing (P2P), Linköping, Sweden, 2003
329. B. Schnizler, D. Neumann, D. Veit, C. Weinhardt: Trading Grid Services - A Multi-attribute Combinatorial Approach. *European Journal of Operational Research*, 2006.
330. M. Schumacher, H. Helin, H. Schuldt (Eds.): CASCOM - Intelligent Service Coordination in the Semantic Web. Birkh<sup>o</sup>user Verlag, Springer, 2008.
331. A. Serafini, S. Mancini, S. Bose: Distributed Quantum Computation via Optical Fibers. *Phys. Rev. Lett.* 96, 010503, 2006.
332. A. Sheth, K. Verma, K. Gomadam: Semantics to energize the full services spectrum. *Communications of the ACM*, 49(7), 2006.
333. P. Shor: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal of Computing*, 26, 1997.
334. P. Shvaiko, J. Euzenat: A Survey of Schema-based Matching Approaches. *Data Semantics*, 2005.
335. M.P. Singh, M. Huhns: Service-Oriented Computing - Semantics, Processes, Agents. John Wiley & Sons Ltd., 2005
336. C. Sierra, P. Faratin, N. Jennings: Deliberative automated reasoning using fuzzy similarities. Proceedings of EUSFLAT-ESTYLF Joint Conference on Fuzzy Logic, 1999.

337. E. Sirin, B. Parsia, J. Hendler: Filtering and Selecting Semantic Web Services with Interactive Composition Techniques. *IEEE Intelligent Systems*, July/August, 2004.
338. E. Sirin, B. Parsia, D. Wu, J. Hendler, D. Nau: HTN planning for Web service composition using SHOP2. *Web Semantics*, 1(4), Elsevier, 2004.
339. E. Sirin, J. Hendler, B. Parsia: Semi-automatic Composition of Web Services using Semantic Descriptions. Proceedings of Intl Workshop on Web Services: Modeling, Architecture and Infrastructure workshop in conjunction with ICEIS conference, 2002.
340. D.E. Smith, D.S. Weld: Conformant Graphplan. Proceedings of 15th AAAI Conference on on AI, Pittsburgh, USA, 1998.
341. R. G. Smith: The contract net protocol: High level communication and control in a distributed problem solver. *IEEE Transactions on Computing*, 1980.
342. L. Soh and C. Tsatsoulis: Allocation algorithms in dynamic negotiation-based coalition formation. Proceedings of AAMAS 2002 Workshop on Teamwork and coalition formation, 2002.
343. J. Spohrer, P.P. Maglio, J. Bailey, D. Gruhal: Steps toward a science of service systems. *IEEE Computer*, IEEE CS Press, January 2007
344. B. Srivastava, J. Koehler: Web Service Composition: Current Solutions and Open Problems. Proceedings of the ICAPS 2003 Workshop on Planning for Web Services, 2003.
345. S. Staab, H. Stuckenschmidt (eds.): Semantic Web and Peer-to-Peer. Springer, 2006.
346. J. Stolze, D. Sutter: Quantum Computing. A Short Course from Theory to Experiment. Wiley-VCH, Physics Textbook, 2004.
347. S. Staab, R. Studer (eds.): Handbook on Ontologies. Springer, 2004.
348. R. Studer, S. Grimm, A. Abecker (eds.): Semantic Web Services. Concepts, Technologies, and Applications. Springer, 2007.
349. M. Stollberg, U. Keller, H. Lausen, S. Heymans: Two-phase web service discovery based on rich functional descriptions. Proceedings of European Semantic Web Conference, Buda, Montenegro, LNCS, Springer, 2007.
350. J. Suijs, P. Borm, A. De Waegenaere, S. Tijs: Cooperative games with stochastic payoffs. *European Journal for Operational Research*, 113:193–205, 1999.
351. SWRL: Semantic Web Rule Language Combining OWL and RuleML. [www.w3.org/Submission/2004/SUBM-SWRL-20040521/](http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/)  
W3C Team Comment: [www.w3.org/Submission/2004/03/Comment](http://www.w3.org/Submission/2004/03/Comment)
352. P. Suppes, J.A. De Barros: Quantum Mechanics and the Brain. In [50]. 2007.
353. J. Sutton: Non-cooperative Bargaining Theory: An Introduction. *Review of Economic Studies*, 53(5), 1986.
354. K. Sycara, D. Zeng: Benefits of learning in negotiation. Proceedings of National Conference on Artificial Intelligence (AAAI), USA, 1997.
355. **K. Sycara, M. Klusch, S. Widoff, J. Lu: LARKS: Dynamic Matchmaking Among Heterogeneous Software Agents in Cyberspace. *Autonomous Agents and Multi-Agent Systems*, 5(2):173 - 204, Kluwer Academic, 2002.**
356. K. Sycara, M. Klusch, S. Widoff, J. Lu: Dynamic Service Matchmaking Among Agents in Open Information Environments. *ACM SIGMOD Record*, 28(1), March 1999.

357. S. Tani, H. Kobayashi, K. Matsumoto: Exact Quantum Algorithms for the Leader Election Problem. Proceedings of 22nd Annual Symposium on Theoretical Aspects of Computer Science, Stuttgart, Germany, Springer, LNCS 3404, 2005.
358. M. Tegmark: Why the brain is probably not a quantum computer. *Information Science*, 128(3-4), 2000.
359. S. Tobies: The Complexity of Reasoning with Cardinality Restrictions and Nominals in Expressive Description Logics. *Artificial Intelligence Research (JAIR)*, 12, 2000.
360. D. Toman, G.E. Weddell: On Reasoning about Structural Equality in XML: A Description Logic Approach. Proceedings Intl. Conference on Database Theory ICDT, LNCS, 2572:96 - 110, 2003.
361. D. Trastour, C. Bartolini, C. Priest: Semantic Web Support for the Business-to-Business E-Commerce Lifecycle. Proceedings of the International World Wide Web Conference (WWW), 2002.
362. P. Traverso, M. Pistore: Automated Composition of Semantic Web Services into Executable Processes. Int Semantic Web Conference, LNCS 3298, Springer, 2004.
363. D. Tsarkov, A. Riazanov, S. Bechhofer, I. Horrocks: Using Vampire to reason with OWL. Proceedings of International Semantic Web Conference (ISWC), Springer, LNCS, 2004.
364. D. Tsoumakos, N. Roussopoulos: Adaptive Probabilistic Search (APS) for Peer-to-Peer Networks. Proceedings Int. IEEE Conference on P2P Computing, 2003.
365. R. Unland, M. Klusch, M. Calisti (eds.): Software Agent-Based Applications, Platforms and Development Kits. Whitestein Series in Software Agent Technologies and Autonomic Computing , Birkhäuser Verlag, 2005.
366. R. Vaculin, K. Sycara: Towards automatic mediation of OWL-S process models. IEEE International Conference on Web Services (ICWS 2007), 2007.
367. W.M.P. van der Aalst, A.J.M.M. Weijters: Process mining: a research agenda. *Computers in Industry*, 53, 2004.
368. A. van Gelder, K. Ross, J. S. Schlipf: The well-founded semantics for general logic programs. *ACM*, 38(3):620-650, 1991.
369. L.Z. Varga, A.Hajnal, Z. Werner: The WSDL2Agent Tool. In [365].
370. H.M.W. Verbeek, W.M.P. van der Aalst: Analyzing BPEL processes using Petri nets. Proceedings of the Second International Workshop on Applications of Petri Nets to Coordination, Workflow and Business Process Management, Miami, USA, 2005.
371. W.M.P. van der Aalst, M. Dumas, C. Ouyang, A. Rozinat, H.M.W. Verbeek: Conformance Checking of Service Behavior. *ACM Transactions on Internet Technology (TOIT)*, Special issue on Middleware for Service-Oriented Computing, 2007.
372. K. Verma, K. Sivashanmugam, A. Sheth, A. Patil, S. Oundhakar, J. Miller: METEORS WSDI: A Scalable P2P Infrastructure of Registries for Semantic Publication and Discovery of Web Services. *Information Technology and Management*, Special Issue on Universal Global Integration, Vol. 6, No. 1, 2005.
373. G. Vetere, M. Lenzerini: Models for semantic interoperability in service-oriented architectures. *IBM Systems Journal*, 44(4): 887-903, 2005.
374. J. von Neumann: *Mathematische Grundlagen der Quantenmechanik*. SpringerVerlag, 1932.

375. J. von Neumann, O. Morgenstern: Theory of games and economic behaviour. Princeton University Press, USA, 1944.
376. J. Vokrinek, J. Biba, J. Hodik, J. Vybihal, M. Pechoucek: Competitive contract net protocol. In Jan van Leeuwen et Al., editor, *SOFSEM 2007: Theory and Practice of Computer Science*, LNCS vol. 4362 (1), Springer, 2007.
377. L.H. Vu, M. Hauswirth, F. Porto, K. Aberer: A Search Engine for QoS-enabled Discovery of Semantic Web Services. Ecole Polytechnique Federal de Lausanne, LSIR-REPORT-2006-002, Switzerland, 2006. Also available in the Special Issue of the International Journal of Business Process Integration and Management (IJBPIIM) (2006).
378. Web Intelligence and Agent Systems. International Journal, IOS Press. <http://wi-consortium.org/html/journal.php>
379. M. Wellman, P.R. Wurman: Market-aware agents for a multiagent world. *Robotics and Autonomous Systems*, 24, 1998.
380. G. Weiss (ed.): Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. MIT Press, 1999.
381. T. Weithöner: ABox Benchmarking OWL Reasoners. Technical Report ITR: I-FN-83, DoCoMo Euro-Labs GmbH, September 2006.
382. M. Weske: Business Process Management - Concepts, Languages, Architectures. Springer, 2007.
383. E. Wolfstetter: Auctions: An introduction. *Economic Surveys*, 10(4), 1996.
384. M. Wooldridge, N. Jennings: Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 9(4), 1999.
385. M. Wooldridge: Reasoning About Rational Agents. MIT Press, 2000.
386. Z. Wu, K. Gomadam, A. Ranabahu, A. Sheth, J. Miller: Automatic Composition of Semantic Web Services using Process Mediation. Proceedings of the 9th Intl. Conf. on Enterprise Information Systems ICES 2007, Funchal, Portugal, 2007.
387. J. Yan, R. Kowalczyk, J. Lin, M.B. Chhetri, S.K.Goh, J. Zhang: Autonomous service level agreement negotiation for service composition provision. *Future Generation Computing Systems*, 23(6), Elsevier, 2007.
388. J. Yang, W.J. Heuvel, M.P. Papazoglou: Tackling the Challenges of Service Composition in E-marketplaces. Proceedings of RIDE-2EC 2002, San Jose, CA, USA, 2002.
389. G. Yang and M. Kifer: Well-Founded Optimism: Inheritance in Frame-Based Knowledge Bases. Proceedings of 1st International Conference on Ontologies, Databases and Applications of Semantics (ODBASE), Irvine, California, 2002.
390. Y. Yang, Q. Tan, Y. Xiao: Verifying web services composition based on hierarchical colored Petri nets. Proceedings of the 1st International Workshop on Interoperability of Heterogeneous Information Systems, Bremen, Germany, ACM Press, 2005.
391. A.M. Zaremski, J.M. Wing: Specification Matching of Software Components. *ACM Transactions on Software Engineering and Methodology*, 6(4), 1997.
392. A. Zeilinger: Einsteins Schleier - Die neue Welt der Quantenphysik. C.H. Beck Verlag, 2003.
393. O. Zimmermann: Building Service-Oriented Architectures with Web Services. Tutorial at the 4th European Conference on Web Services (ECOWS), Zurich, Switzerland, 2006.