Towards the Analysis of Components Interoperability by means of *i**

Karina Abad[1] and Wilson Pérez[2] and Juan Pablo Carvallo[3]

[1] CEDIA, Cuenca, Ecuador
[2] Universidad de Cuenca, Cuenca, Ecuador
[3] Universidad del Azuay, Cuenca, Ecuador
karina.abad@cedia.org.ec; wilson.perez@ucuenca.edu.ec;
jpcarvallo@uazuay.edu.ec

Abstract. Information Systems lay at the heart of today's enterprise organizations. As the organizations grow, Information Systems should evolve to adapt to new requirements emerging from changes in business models, including the adoption of new services paradigms or operational improvements. These emerging requirements impact Information Systems Architecture, requiring a continuous analysis and adaptation to new realities. In previous work, we have proposed the DHARMA Method aimed at the identification of Information System Architecture by means of the *i** framework. In this on-going work we focus in the results of activity 4 of the method, which uses SR diagrams to analyze interoperability among System Actors. The results of the application of this method in a large industrial case, are used to illustrate the proposal.

Keywords: DHARMA Method · interoperability · Information Systems.

1 Introduction

Information Systems (IS) are essential in today's enterprise organizations, being a fundamental tool to support operation and decision-making processes. As the organizations grow, IS should evolve in accordance and adapt to new requirements emerging from changes in business models, including the adoption of new services paradigms or operational improvements. In the last decades, several alternatives to the traditional development of software from scratch, have emerged in order to construct such IS. In the usual case, they are built as hybrid systems which integrate several software components of different nature and origins e.g., legacy systems, web services, commercial components (typically referred as COTS) and, Free and/or Open Source Software (FOSS). In previous work [6] we have proposed the DHARMA method, intended to support the identification of hybrid IS architecture. The method encompasses four main activities, which make intensive use of the *i** notation, to produce a set of deliverables in the form of *i** SD and SR diagrams.

The DHARMA method has been applied in over 130 industrial cases, which helped to discover: a set of patterns [1] to ease the construction of Enterprise Context Models (CM); a catalog of actors and dependencies [5] to guide in the construction of such

models; and parametric actors and dependencies intended to automate this task [11], among others. We have also used this method to support involvement of non-technical stakeholders in the requirements engineering process [11]. In this work, we will further analyze interoperability among System Actors (e.g. software components) in IS, for which we will focus in the fourth activity of the DHARMA method.

This document is structured as follows: Section 2 presents the background and related work, section 3 presents the case study used to illustrate this on-going work, section 4 introduces a discussion how DHARMA can be used to support interoperability analysis and section 5 presents some conclusions and future lines of work.

2 Background and Related Work

The DHARMA method [6] has been proposed to support the definition of enterprise architectures using the *i** framework. The theoretical bases to support the method in the analysis of enterprise context, structure and strategy, are two concepts defined by Porter [7]: 1) the model of the market forces, used to reason about potential available strategies and how to make them profitable, by analyzing existing dependencies with external actors within five market forces, and 2) Value chain, which includes primary and support activities helpful to identify internal actors and dependencies in the scope of the organization. The DHARMA method consists in four activities, as shown in figure 1:

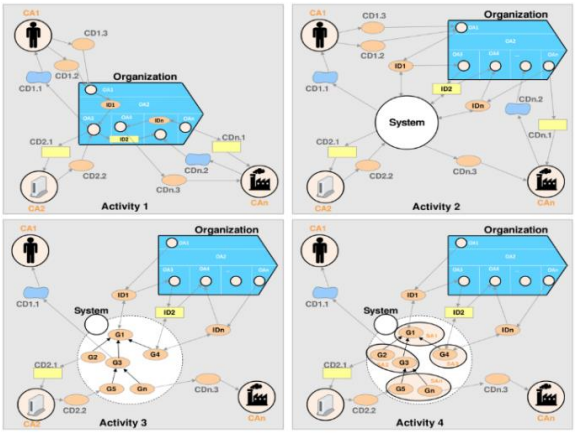

Fig. 1. The DHARMA Method

Activity 1. Modelling the enterprise context. The organization and its strategy are carefully analyzed to identify its role inside the context, allowing the definition of Context Actors (CA) and Organizational Areas (OA). At the end of this activity, *i** SD models are built and used to support reasoning and represent results from this activity.
Activity 2. Modelling the environment of the system. In this activity, a system-to-be is placed into the organization and its impact over the elements in the CM is analyzed. The strategic dependencies of OAs and CAs are inspected to determine which of them

may be totally or partially satisfied by system. The result of this activity is also an *i** SD model representing the dependencies that the system can satisfy in relation to the different CAs or OAs.

Activity 3. Decomposition of system goals. Dependencies included in the CM are analyzed and decomposed into a hierarchy of goals required to satisfy them. The goals represent the services that the system must provide. An *i** SR diagram for the system is built.

Activity 4. Identification of system architecture. This activity includes the identification of System Actors (SA), which play a role in the system and represent atomic software domains. Goals identified in Activity 3 are analyzed and semantically grouped. Each aggrupation revels the services that are expected to be covered by SA.

In previous works, we have presented a catalog of actors and dependencies [5] to avoid the construction of CM from scratch, an ontology describing the *i** elements and their relations in SD and SR models, and a web application to ease the construction of CM and to create and visualize the resulting *i** SD and SR diagrams [8], the creation of a semantic repository with many CM that can be reused [9], among others. The method has been applied in 36 organizations to validate its construction [10] and to identify frequent problems in the *i** framework and the application of DHARMA.

As consequence of this work, we were able to confirm that, the size of resulting models makes difficult to visualize IS architecture, and thus, to analyze functional and non-functional coverage SA by software components, and communication interfaces among them, represented by dependencies among covered SA. To support this process, in this work, we focus in Activity 4 of the DHARMA method and present a first result in the effort to analyze interoperability among SA in a graphical environment.

In [3], Amr and Mansouri define interoperability of IS as the ability of systems to communicate and interact between them through a common language, facilitating the sharing of information, essentially data, and reaching a better performance. Also, authors in [4], emphasizes that interoperability is more than just communication or the integration of many systems into one, but a software approach to maximize benefits of diversity.

3 The Study

As part of this work, we have applied the DHARMA method and therefore the *i** framework in a Public Institute of Higher Education in Ecuador, which purpose is to provide professional education services and research activities. The organization is composed by 26 Departments and Administrative Areas, representing the Organizational Actors (OAs).

The aim of applying DHARMA in that institution was to perform the strategic planning of IT. The resulting model was used to identify projects of development, acquisition and evolution of software needed to implement the IS, as well as communication interfaces between them.

A technical team of 8 experts was formed and trained in the *i** framework and the DHARMA method; the team performed interviews to different OAs and created several models as explained in the next paragraphs.

Activity 1. At the end of the interviews, 42 Actors were identified, 26 Organizational Actors (OA) and 16 Context Actors (CA). A total of 596 Strategic Dependencies (SD) were identified, among them: 43 represent Resources, 500 Goals and 53 Qualities. Since the information was collected by different teams, we performed a validation of all actors and dependencies identified and combine them into a final model. At the end we deleted 32 duplicated dependencies and added 49 new ones. Combined SD model includes 613 dependencies (30 Resources, 533 Goals and 50 Qualities).

Activity 2. Each dependency in the combined SD model was analyzed in order to define its feasibility for being automated. We found that 497 out of 613 dependencies were prone be totally automated, whilst 94 could be partially automated. These dependencies were included into the resulting SD IS Context model.

Activity 3. We created a SR decomposition, using a Use Case based analysis considering CRUDs, transactions, procedures and queries, required to automate each dependency in the SD IS context model. Final SR model of the IS included 976 elements.

Activity 4. Elements in the SR model of the IS were semantically clustered into 123 atomic SA, which at the end where categorized into 18 components (or modules) grouping SA in similar functional areas. There are several possible scenarios for this: functionality of a SA covered by a single component; functionality of several SA covered by a single component; functionality of a SA covered by several components for redundancy reasons -e.g ubiquity-; or even the case where no software components exist to cover the functionality of a SA, requiring the construction of bespoke software. Identified components interoperate with each other through one or more interfaces, represented by links among SR elements within each component boundary (see figure 2 for an excerpt of the resulting IS model).

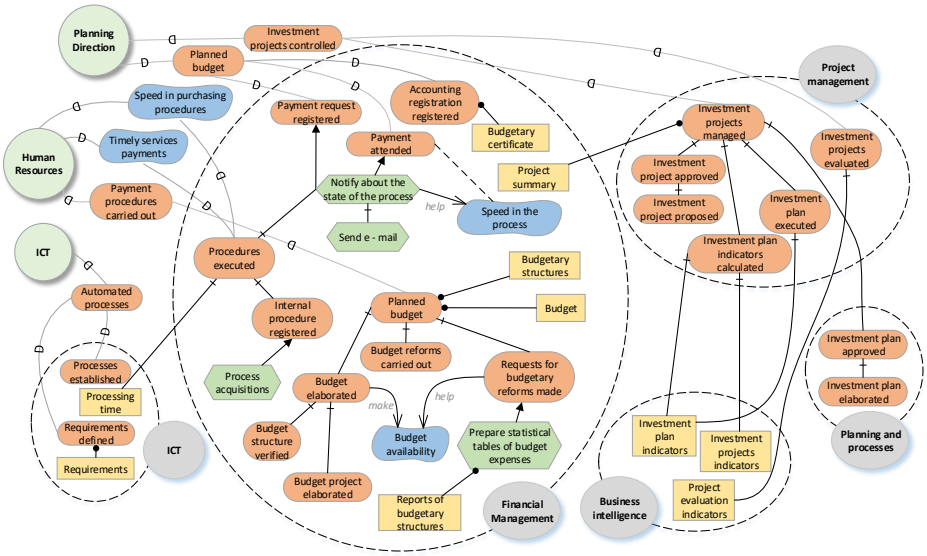

Fig. 2. Excerpt of the *i** model of the IS.

4 Discussion

Deeping the analysis of the *i** model resulting from Activity 4 of DHARMA, we argue that the analysis consists in the identification of which SA should implement the intentional elements identified in Activity 3, and which should use the intentional elements implemented in other SA (by linking them as intentional elements outside their boundary). At the end, these links represent the communication interfaces among SA.

Table 1 has been constructed from the resulting SR IS model, by counting the number of intentional element links between pairs of SA. Table 1 (a) shows the number of interfaces considering the direction of the links, whilst table 1 (b) shows the sum of the communication interfaces existing between a pair of SA.

Table 1. Interfaces required by pairs of SA. a) Considering its direction; b) Total.

	ACADEMY	AUDIT	LIBRARY	COMMUNICATION AND EVENTS	AGREEMENTS	PHISICAL SPACE	ORGANIZATIONAL STRUCTURE	PERFORMANCE EVALUATION	EVALUATION AND ACCREDITATION	FINANCIAL	PROJECT MANAGEMENT	DOCUMENTARY MANAGER	BUSINESS INTELLIGENCE	RESEARCH	PLANNING AND PROCESSES	OCCUPATIONAL HEALTH AND SAFETY	HUMAR RESOURCES	ICT
ACADEMY		3	3	3	2	1	2	1	2	12	2	9	6	1	3	2	10	3
AUDIT							1					2					1	
LIBRARY	3						1		1			4	2		1			1
COMMUNICATION AND EVENTS	3	1	1		1		1			3	8	3	5	1	1		2	1
AGREEMENTS	1						1					3					1	
PHISICAL SPACE	2						1		1	2	1		1					1
ORGANIZATIONAL STRUCTURE	2	1	1		1	1		1	1	8		3		1	1		3	2
PERFORMANCE EVALUATION	1	1							1			4	2		1		2	
EVALUATION AND ACCREDITATION	2			1		1	1	1		3	1		2	1			5	
FINANCIAL	12	6		7	6	2	5		2		20	15	14	6	5	1	14	
PROJECT MANAGEMENT	3									2			3	1	2			
DOCUMENTARY MANAGER	14	1	3	9	4	3	3	3	3	26	14		8	3	8	1	19	2
BUSINESS INTELLIGENCE	1	2	1	1					2	1	1				1	1	2	
RESEARCH	3	1	1	2	1			1	1		2	4	2		2		4	
PLANNING AND PROCESSES	6		1	2	2		2	1	1	6	4	6	4	1				3
OCCUPATIONAL HEALTH AND SAFETY																		
HUMAR RESOURCES	11	1	1	6	3	1	5	5	5	23	5	21	11	3	8	1		1
ICT				1			1					1	2				3	

(a)

	ACADEMY	AUDIT	LIBRARY	COMMUNICATION AND EVENTS	AGREEMENTS	PHISICAL SPACE	ORGANIZATIONAL STRUCTURE	PERFORMANCE EVALUATION	EVALUATION AND ACCREDITATION	FINANCIAL	PROJECT MANAGEMENT	DOCUMENTARY MANAGER	BUSINESS INTELLIGENCE	RESEARCH	PLANNING AND PROCESSES	OCCUPATIONAL HEALTH AND SAFETY	HUMAR RESOURCES	ICT
ACADEMY		3	6	6	3	3	4	2	4	24	5	23	7	4	9	13	10	3
AUDIT				1			2	1		6		3	2	1			2	
LIBRARY				1			2		1			7	3	1	2		1	1
COMMUNICATION AND EVENTS					1		1		1	10	8	12	6	3	3		8	2
AGREEMENTS							2			6		7		1	2		4	
PHISICAL SPACE							2		2	4	1	3	1				1	1
ORGANIZATIONAL STRUCTURE								1	2	13		6		1	3		8	3
PERFORMANCE EVALUATION									2			7	2	1	2		7	
EVALUATION AND ACCREDITATION										5	1	3	4	2	1		10	
FINANCIAL											22	41	15	6	11	1	37	
PROJECT MANAGEMENT												14	4	3	6		5	
DOCUMENTARY MANAGER													8	7	14	1	40	3
BUSINESS INTELLIGENCE														2	5	1	13	2
RESEARCH															2		4	
PLANNING AND PROCESSES																	8	3
OCCUPATIONAL HEALTH AND SAFETY																	1	
HUMAR RESOURCES																		4
ICT																		

(b)

As example, for table 1 (a) consider the SA Project Management which interacts with Business Intelligence through three links, and Business Intelligence requires one interaction from Project Management. On the other hand, table 1 (b) shows that Project Management and Business Intelligence interact between them a total of four times.

In other words, the construction of the final *i** SR diagram allowed us to determine the communication interfaces between different components, which also are directly related to the interoperability of the whole IS. These communication interfaces – showed in table 1 (b) - have been represented in a chord diagram (figure 3). Each SA is represented by one color and the lines between them represent the intentional element links. The width of each line is directly related with the number of intentional element links between a pair of actors. This figure summarizes the findings from the final *i** SR model and provides an overview of how interoperable SA are within IS architecture and thus, which SA requires more attention because of the degree of interconnection

with other IS components. Also, departing from this figure, information systems architects will be able to justify, or not, the need of a service bus, to estimate the number of services to be hosted and the characteristics or kind of data that services to be implemented shall carry.

Additionally, this representation helps to focus into the SA with greater interoperability requirements (e.g. those demanding the most communication interfaces) and to prioritize its implementation or adaptation. Bidirectional traceability provided by the models created through the activities of the DHARMA method, help to improve impact analysis. Since dependences with CA and OA are linked to intentional elements within the scope of SA, any strategic decision in relation to them (e.g. inclusion of new ECA or OA, change in the intentional element in relation to them), can be used to map the impact on intentional elements in IS architecture.

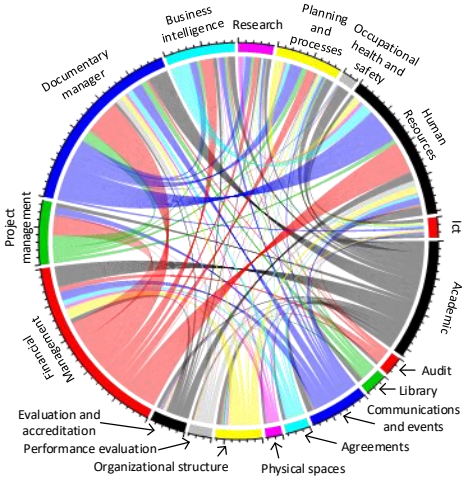

Fig. 3. Interfaces of communication between the SA

5 Conclusions and Future Work

In this work we presented an on-going work in relation to interoperability of software components in hybrid IS. We started by applying the *i** framework and the DHARMA method to define IS architecture in a large-scale project. The resulting artifacts were used to analyze interoperability among SA (atomic software domains within an IS and the software components covering its functionality). This analysis allowed for the visual identification of the more critical components in terms of interoperability within the IS and the nature of the interfaces required to implement data communication among them. As future work, we pretend to analyze in depth the identified interfaces and how they can be translated to communication mechanisms e.g. web services. By identifying these elements in an early stage of IS engineering, we hope to be able to reason about the best possible scenarios to simplify hybrid IS architecture, as well as its implementation and evolution.

References

1. Carvallo, J. P., Franch, X.: Building Strategic Enterprise Context Models with *i**: A Patern-Based Approach. TEAR 2012. (2012)
2. Amr, M. F. et al.: Toward the development of a hybrid model of ontological database for the interoperability of several information systems. 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt) (2016): 648-653.
3. Amr, M. F., Mansouri, K.: Toward the elaboration of model of adaption and INTERFACAGE for the interoperability of severan information systems. Mediteraan Cogress of Telecommunications, CMT 2016 (2016).
4. Agostinho C., et al.: Interoperability of Complex Business Networks by Language Independent Information Models. In: Advanced Concurrent Engineering. Springer, London. 2010.
5. Abad K., Pérez W., Carvallo J.P., Franch X.: A Catalogue of Reusable Context Model Elements Based on the *i** Framework. In: Conceptual Modeling. ER 2017. (2017).
6. Carvallo, JP., Franch, X. Descubriendo la arquitectura de sistemas de software híbridos: un enfoque basado en Modelos *i**. Proceedings of RE'09, pp. 45- 56. (2009).
7. Porter, M.: Competitive Strategy. Free Press. New York, NY, USA. (1980).
8. Pérez, Wilson et al.: Supporting *i**-Based Context Models Construction through the DHARMA Ontology. iStar'17. (2017).
9. Abad, K., Carvallo, JP., Espinoza, M., Saquicela, V.: Hacia la Creación de un Repositorio Semántico de Modelos de Contexto Basados en *i** y el método DHARMA. RISTI - Revista Ibérica de Sistemas e Tecnologías de Informac˜ao, (17), 41-56. (2016).
10. Abad, K., Pérez, W., Carvallo, JP., Franch, Xavier: *i** in practice: Identifying frequent problems in its application. ACM Symposium on Applied Computing. (2017). 1122-1129. 10.1145/3019612.301975.
11. Carvallo, J.P., Franch, X.: An empirical study on the use of i* by non-technical stakeholders: the case of strategic dependency diagrams. Requirements Engineering. (2019).