Mediating Between Qualitative and Quantitative Representations for Task-Orientated Human-Robot Interaction

Michael Brenner
Albert-Ludwigs-University
Freiburg, Germany
brenner@informatik.uni-freiburg.de

Nick Hawes
School of Computer Science
University of Birmingham, UK
N.A.Hawes@cs.bham.ac.uk

John Kelleher
Dublin Institute of Technology
Dublin, Ireland
John.Kelleher@comp.dit.ie

Jeremy Wyatt
School of Computer Science
University of Birmingham, UK
J.L.Wyatt@cs.bham.ac.uk

Abstract

In human-robot interaction (HRI) it is essential that the robot interprets and reacts to a human's utterances in a manner that reflects their intended meaning. In this paper we present a collection of novel techniques that allow a robot to interpret and execute spoken commands describing manipulation goals involving qualitative spatial constraints (e.g. "put the red ball near the blue cube"). The resulting implemented system integrates computer vision, potential field models of spatial relationships, and action planning to mediate between the continuous real world, and discrete, qualitative representations used for symbolic reasoning.

1 Introduction

For a robot to be able to display intelligent behaviour when interacting with humans, it is important that it can reason qualitatively about the current state of the world and possible future states. Being an embodied cognitive system, a robot must also interact with the continuous real world and therefore must link its qualitative representations to perceptions and actions in continuous space. In this paper, we present an implemented robot system that mediates between continuous and qualitative representations of its perceptions and actions.

To give an impression of the robot's capabilities, consider a hypothetical household service robot which is able to accept an order to lay the dinner table such as "put the knives to the right of the plate and the forks to the left of the plate." The robot has to interpret this utterance and understand it as a goal it must achieve. It has to analyse its camera input to find the objects referred to in the owner's command and it must also interpret the spatial expressions in the command in terms of the camera input. Finally, it must plan appropriate actions to achieve the goal and execute that plan in the real world. In this paper, we present a system that is able to accomplish such tasks. In our domain we use cubes and balls in place of cutlery as our robot's manipulative abilities are limited (see Figure 7). In this domain our system acts on commands such as, "put the blue cube near the red cube" and "put the red cubes and the green balls to the right of the blue ball".

We are particularly interested in the consistent interpretation and use of spatial relations throughout the modalities available to a robot (e.g. vision, language, planning, manipulation). For their different purposes, these modalities use vastly different representations, and an integrated system must be able to maintain consistent mappings between them. This is a hard problem because it means mediating between the quantitative information about objects available from vision (e.g. where they are in the world), the qualitative information available from language (e.g. descriptions of objects including spatial prepositions), the qualitative information that must be generated to reason about actions (e.g. hypothetical future configurations of objects), and the quantitative information required by an action system in order manipulate objects. Additionally, when a robot interacts with humans, mediation capabilities must extend across system borders: the robot must be able to interpret the intended meaning of human input in terms of its own representational capabilities and react in a way that reflects the human's intentions. Our system makes the following contributions in order to tackle these problems:

i) **Planning-operator driven interpretation of commands**: we describe a generic method which uses formal planning operators to guide the interpretation of commands in natural language and automatically generates formal planning goals. Referential expressions in the goal are kept for "lazy" resolution by the planner in the context of a given world state. This allows replanning to dynamically adapt behaviour without having to re-evaluate commands.

ii) **Spatial model application**: we use potential field models of spatial prepositions to generate qualitative representations of goals that satisfy a human command. This model accords with the ways that humans talk about spatial relations [Costello and Kelleher, 2006]. This approach allows us to generate discrete solutions that fit typical human descriptions of continuous space.

Although the individual techniques are still somewhat limited

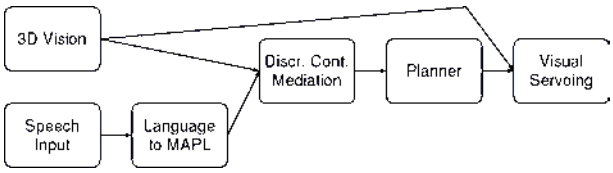


Figure 1: The system architecture.

```
(:action put
 :parameters
    (?a - agent ?obj - movable ?wp - waypoint)
 :precondition (and
    (pos ?obj : ?a)
    (not (exists (?obj2 - movable) (pos ?obj2 : ?wp))))
 :effect
    (pos ?obj : ?wp))
```

Figure 2: MAPL operator *put* for placing objects.

in scope, by combining them we provide each component in the architecture access to more information than it would have in isolation. Thus, the overall system is able to demonstrate intelligent behaviour greater than the sum of its parts.

In the following section we describe our robot platform and the system architecture. We then expand on this in sections §4 (planning domain), §5 (command interpretation), §6 (potential field models for spatial relations) and §7 (qualitative spatial information from vision). Finally §8 presents an example of the functionality of the complete system.

2 Relation to Previous Work

The work presented in this paper is related to various subfields of robotics and artificial intelligence. In particular it is closely related to human-robot interaction and situated language understanding [Kruijff *et al.*, 2006]. We do not focus solely on the process of understanding an utterance, but instead examine the steps necessary to mediate between the various representations that can exist in systems that must act on the world as a result of a command in natural language.

In terms of gross functionality there are few directly comparable systems, e.g. those presented in [Mavridis and Roy, 2006] and [Mcguire *et al.*, 2002]. Whereas these systems specify complete architectures for following manipulation commands, we focus on a particular aspect of this behaviour. As such our approach could be utilised by existing systems. For example, it could be used in layer L3 of Mavridis and Roy's Grounded Situation Model [2006] to produced discrete, categorical encodings of spatial relationships.

There are many plan-based dialogue systems that are used (or potentially usable) for HRI (e.g., [Sidner *et al.*, 2003; Allen *et al.*, 2001]). Most such systems try to exploit the context of the current (dialogue) plan to interpret utterances. We are not aware, however, of any system that, like ours, actually uses the formal action representation from the planning domain to resolve referential expressions in at least a semi-formal way. Critically, the "guidance" provided by the planning domain leads to a logical representation of the command that the planner can reason about. For example, the planner is able resolve referential expressions as part of the problem solving process. This can be significant in dynamic environments: if a situation changes then the planner can resolve the same referential expression differently.

3 Architecture

To enable a robot to follow action commands such as those described in the introduction, we break the problem into a number of processing steps. These steps are reflected by the design of our overall processing architecture, which can be seen in Figure 1. Our system is based on a combination of an iRobot B21r mobile robot and a table-mounted Katana 6M robotic arm. Mounted on the B21r is a pan-tilt unit supporting two parallel cameras which we use for visual input. From these cameras we create a 3D representation of the scene using depth information from stereo to instantiate a collection of simple object models. To produce actions, information from vision is fed into a workspace-based visual-servoing system along with instructions about which object to grasp, and where to put it. Actions are limited to pick and place. This suffices for the current experimental scenarios.

4 Planning Domain

For the purpose of this paper, a simple ontology was designed which consists mainly of agents and objects. Objects may be movable or not. They can have properties, e.g. colours, that can be used to describe them or constrain subgroups of objects in a scene. Positions of objects in a scene are described by *waypoints*. Concrete instances of waypoints are generated on-the-fly during the problem-solving process (cf. §7). Relations between waypoints include *near*, *right of*, and *left of*. Despite being quite simple, this ontology allows us to represent complex situations and goals. Moreover, it is very easy to extend to richer domains. For example, adding just one new subtype of movable objects would enable the robot to distinguish between objects that are stackable.

The ontology has been modelled as a planning domain in MAPL [Brenner, 2005], a planning language for multiagent environments based on PDDL [Fox and Long, 2003]. MAPL is suitable for planning in HRI because it allows us to model the beliefs and mutual beliefs of agents, sensory actions, communicative actions, and different forms of concurrency. Although these features make MAPL highly suitable for human-robot interaction, in this paper we mostly use the ADL subset of MAPL. Figure 2 shows the operator for placing objects. Note that MAPL uses non-boolean state variables, e.g. *(pos obj)*, which are tested or changed with statements like *(pos obj : ?wp)*. Thus, in MAPL there is no need to state that the robot no longer holds the object after putting it down (a statement which would be necessary in PDDL).

Currently, no planner is available that is specifically designed for MAPL. Instead, we use a compiler for transforming MAPL into PDDL and back. This enables us to use a state-of-the-art planner in our system without losing the descriptive power of MAPL; the planning system used currently is FF [Hoffmann and Nebel, 2001].

5 Converting Linguistic Input to MAPL

In AI Planning, goals are typically formulated in (a subset of) first-order logic, i.e. as formulae that must hold in the state achieved by the plan (see, for example, the definition of goals in the ADL subset of PDDL [Fox and Long, 2003]). Humans, however, usually use imperative commands, like "clear the table", when communicating goals. One reason for verbalising an action command instead of a goal description could be that the former provides a very compact representation of the latter by means of its postconditions, i.e. the immediate changes to the world caused by the action. Speaking in AI Planning terms, if the action "clear table" has an ADL effect saying that after its execution "there exists no object that is on the table", the action name plus its parameters is a much simpler means to convey that goal than the effect formula. What complicates the matter is that, in contrast to AI planners, humans usually do not use unique names for objects, but refer to them in expressions that constrain the possible referents (i.e. "the red ball" instead of `object17`). Altogether, the "human way" to describe goals can be described as *goal = action + parameters + reference constraints*.

Deliberative agents that have ADL-like action representations can exploit this goal description scheme when trying to understand a natural language command: after matching the verb phrase of a command with an appropriate planning operator, this operator can be used to guide the further understanding of the command, namely determining the action parameters and reference constraints.

We will illustrate this process with the command "put the blue cubes to the left of the red ball". Our system parses the command using a simple English grammar and a chart parser. The parse tree of the example command describes the phrase as a verb, followed by a nominal phrase and a prepositional phrase (V NP PP). When the system detects the verb "put", it is matched to the planning operator *put* (cf. Figure 2). The subsequent interpretation procedure is specific to that operator and aims at determining the constraints describing the three parameters of the operator, *?a*, *?obj* and *?wp*. This prior knowledge drives the interpretation of the phrase and simplifies this process significantly. In our example, the NP is interpreted to describe the object *?obj* that is to be moved while the PP describes the target position *?wp*. The following logical constraint on the parameters *?a*, *?obj* and *?wp* is found (in which *?obj1* is the landmark[1] object in relation to which the goal position is described):

(blue ?obj) ∧(type ?obj : cube) ∧
∃?wp1. ((left-of ?wp ?wp1) ∧ ∃?obj1. ((red ?obj1) ∧
(type ?obj1 : ball) ∧ (pos ?obj1 : wp1)))

Additionally, the interpretation states that *all* objects satisfying the constraints on *?obj* must be moved. This quantification becomes visible in the final translation of the command into a MAPL goal, shown in Figure 3 (where type constraints are transformed into the types of the quantified variables).

One important aspect of the natural language command is that it refers both to the goal state (where should the blue cubes be put?) and to the initial state (the reference constraints determining the objects). It is crucial for the planning representation to be able to model this difference, otherwise contradictory problems may be generated. For example, the command "put down the object that you are holding" provides two constraints on the object's position: that it is held by the robot *now*, but is on the ground *after plan execution*. Therefore, MAPL supports referring back to the initial state in the goal description as shown in Figure 3. The facts that must hold *after* execution of the plan are described by the effect of the *put* action. In our example, this effect describes the new position of the object in question.

```
(forall (?obj - cube) (imply
  (and (initially (blue ?obj)))
  (exists (?wp - waypoint)
     (exists (?obj1 - ball ?wp1 - waypoint) (and
        (initially (red ?obj1))
        (initially (pos ?obj1 : ?wp1))
        (initially (left-of ?wp ?wp1))
        (pos ?obj : ?wp))))))
```

Figure 3: Automatically generated MAPL goal for "put the blue cubes to the left of the red ball"

It is important to realise that the goal descriptions generated by this process still contain the referential expressions from the original command, i.e. they are not compiled away or resolved directly. Instead they will be resolved by the planner. We call this "lazy" reference resolution. It enables the robot to dynamically re-evaluate its goals and plans in dynamic situations. If, for example, another blue cube is added to the scene, the planner will adapt to the changed situation and move all of the blue blocks.

6 Computational Models of Spatial Cognition

To act on the kinds of action commands we are interested in, the robot must be able to translate from the qualitative spatial linguistic description of the location to place the object, to both a geometric description of the location that can be used by the manipulation system (i.e. a geometric waypoint positioned in the robot's world), and a logical description for the planning domain (i.e. a symbolic representation of this waypoint and its relationships with other waypoints). This translation involves constructing computational geometric models of the semantics of spatial terms.

Spatial reasoning is a complex activity that involves at least two levels of representation and reasoning: a *geometric level* where metric, topological, and projective properties are handled; and a *functional level* where the normal function of an entity affects the spatial relationships attributed to it in a context. In this paper we concentrate on the geometric level, although using functional spatial information would not require any significant changes to our overall system.

Psycholinguistic research [Logan and Sadler, 1996; Regier and Carlson, 2001; Costello and Kelleher, 2006] indicates that people decide whether the spatial relation associated with a preposition holds between and landmark object and the regions around it by overlaying a spatial template on the landmark. A *spatial template* is a representation of the regions of acceptability associated with a given preposition. It is centred

[1]For the rest of the paper we will refer to the object or objects that should be moved as the *target*, and the object or objects that are used to define the desired position of the target as the *landmark*.

on the landmark, and for each point in space it denotes the acceptability of the spatial relationship between it and the landmark. Figure 4 illustrates the spatial template for the preposition "near" reported in [Logan and Sadler, 1996].

1.74	1.90	2.84	3.16	2.34	1.81	2.13
2.61	3.84	4.66	4.97	4.90	3.56	3.26
4.06	5.56	7.55	7.97	7.29	4.80	3.91
4.47	5.91	8.52	**O**	7.90	6.13	4.46
3.47	4.81	6.94	7.56	7.31	5.59	3.63
3.25	4.03	4.50	4.78	4.41	3.47	3.10
1.84	2.23	2.03	3.06	2.53	2.13	2.00

Figure 4: Mean goodness ratings for the relation *near*.

If a computational model is going to accommodate the gradation of a preposition's spatial template it must define the semantics of the preposition as some sort of continuum function. A *potential field model* is one widely used form of continuum measure [Olivier and Tsujii, 1994; Kelleher *et al.*, 2006]. Using this approach, a spatial template is built using a construction set of normalised equations that for a given origin and point computes a value that represents the cost of accepting that point as the interpretation of the preposition. Each equation used to construct the potential field representation of a preposition's spatial template models a different geometric constraint specified by the preposition's semantics. For example, for projective prepositions, such as "to the right of", an equation modelling the angular deviation of a point from the idealised direction denoted by preposition would be included in the construction set. The potential field is then built by assigning each point in the field an overall potential by integrating the results computed for that by point by each of the equations in the construction set. The point with the highest overall potential is then taken as the location that the object should be placed at to satisfy the relationship.

7 Qualitative Representations from Vision

The previous sections have discussed the representation we use for planning, how we translate action commands into goal states in this representation, and how we model spatial relationships. This section describes a process that produces an initial state description for the a planning process by applying these techniques to mediate between geometric visual information and the symbolic planning representation.

We break the task of generating a state description from vision and language into three steps: converting information about visible objects into a symbolic representation, adding information about specific spatial relationships to this representation, and generating new information required by the planning process. These last two steps use potential field models in two different ways. The first applies them to known waypoints in the world to generate logical predicates (e.g. "left of") for the planning domain. The second applies a field to a single known waypoint to generate a new set of waypoints that all satisfy a predicate for the planning domain.

for rel in goal relationships **do**
  for wp_l in $waypoints$ **do**
    initialise scene sc
    add landmark wp_l to sc
    for wp_t to in $waypoints - wp_l$ **do**
      add $waypoints - wp_l - wp_t$ to s as distractors
      compute field pf for rel in s
      check value val of pf at wp_{target}
      if $val > 0$ **then**
        add (rel wp_t wp_l) to state

Figure 5: Algorithm for generating spatial relationship rel.

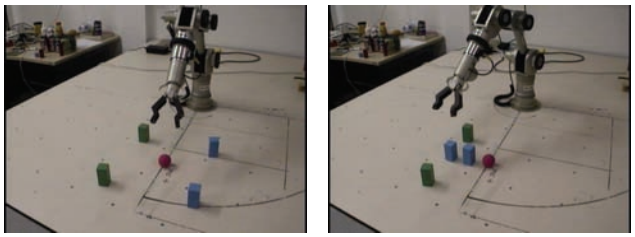
The first step in the process of interpreting and acting upon a command is to translate the information directly obtainable from vision into our planning domain. This is done in a straight-forward way. For each object in the world we generate a description in the language of the planning domain. Each object is represented in the planning domain by an ID which is stored to allow other process to index back into the geometric vision representation via the planning representation. Representing an object involves describing its colour and type (information which is directly available from our vision system). To position the object in the world we must also place the object at a waypoint. To do this we add a waypoint to the planning domain at the centre of the object's bounding box on the table. Waypoints are also represented by a stored ID that can be used access its position in the real world, so that later processes can use this information.

The second step in the process of generating an initial state is to add information about the spatial relations of the waypoints to the planning problem. This allows the planner to reason about how moving objects between waypoints changes their spatial relationships. Rather than add information about all of the spatial relationships that exist between all of the waypoints, we focus only on the relationships included in the goal state because any additional information would be irrelevant to the current task. Thus our approach is explicitly *task-orientated*. The algorithm we use is presented in Figure 5. In this algorithm, *distractors* represent points in the potential field that may influence the field in some way (e.g reducing its value, or altering its extent).

The final step of the initial state generation process is to add additional waypoints in order to give the planner enough suitable object locations to produce a plan. This is necessary because the waypoints from the previous step are all initially occupied, and may not satisfy the spatial constraints in the goal description. To add waypoints, we first ground the target description (e.g. "the blue cubes" from "put the blue cubes left of the red ball") in the visual information to provide a count of the number of objects that must be moved. We then find the waypoint for the landmark object (e.g. "the red ball"), and generate the required number of waypoints in the potential field around the landmark for the given spatial relationship (e.g. "left of"). The algorithm we use to generate the new waypoints is presented in Figure 6. Because

initialise scene sc
add landmark wp_l to sc
add $waypoints$ - $targets$ - wp_l to s as distractors
compute field pf for rel in s
for $i = 0$ to n **do**
  get max of pf
  if $max > 0$ **then**
    add new waypoint at location of max
  else
    return failure

Figure 6: Algorithm for generating n new waypoints for $targets$ at the spatial relationship rel around landmark wp_l.



(a) The initial state. (b) The state after execution

Figure 7: Images of the world before and after plan execution.

this algorithm is greedy, it may fail to place the waypoints in the potential field even if enough space is available. This is something that must be addressed in future work.

8 Worked Example

This section presents an example processing run from the implementation of our system. The initial scene for the example can be seen in Figure 7(a). A visualisation generated by the system is presented in Figure 8(a). The scene contains a red ball, two green cubes and two blue cubes. Processing is started by the command "put the blue cubes to the left of the red ball". This is passed into the linguistic processing component. This component processes the text as described in §5, which produces the MAPL goal state shown in Figure 3. The linguistic input triggers the current scene to be pulled from vision. This returns a scene with a red ball centred at $(200, 200)$, green cubes at $(150, 150)$ and $(150, 250)$, and blue cubes at $(250, 250)$ and $(250, 150)$ (these numbers have been adjusted for a simpler presentation).

The goal and visual information is then used as input into the discrete-continuous mediation process. As described in §7, this process assigns IDs for each object and a waypoint for each object position. This results in the following mapping for the scene (the brackets contain information that is accessible from vision via the IDs):

```
obj_d0 (blue cube) at wp_d1 (250,250)
obj_d2 (blue cube) at wp_d3 (250,150)
obj_d4 (green cube) at wp_d5 (150,250)
obj_d6 (green cube) at wp_d7 (150,150)
obj_d8 (red ball) at wp_d9 (200,200)
```

The qualitative part of this is transformed into a MAPL expression to form part of the initial state for planning:

```
(pos obj_d0 : wp_d1) (pos obj_d2 : wp_d3)
(pos obj_d4 : wp_d5) (pos obj_d6 : wp_d7)
(pos obj_d8 : wp_d9)
(blue obj_d0) (blue obj_d2)
(green obj_d4) (green obj_d6) (red obj_d8)
```

Next, the mapping process uses potential fields to generate the spatial relationships between the waypoints for all of the visible objects. Only the relationships necessary to satisfy the goal state are considered, so in this case only the "left of" relationship is considered. Part of this process is presented in Figure 8(b), which shows the "left of" field for the top right cube. In this picture the camera is positioned directly in front of the red ball (hence the field being tilted). This results in the following information being added to the initial state:

```
(left_of wp_d5 wp_d1) (left_of wp_d9 wp_d1)
(left_of wp_d5 wp_d3) (left_of wp_d7 wp_d3)
(left_of wp_d9 wp_d3) (left_of wp_d5 wp_d9)
(left_of wp_d7 wp_d9)
```

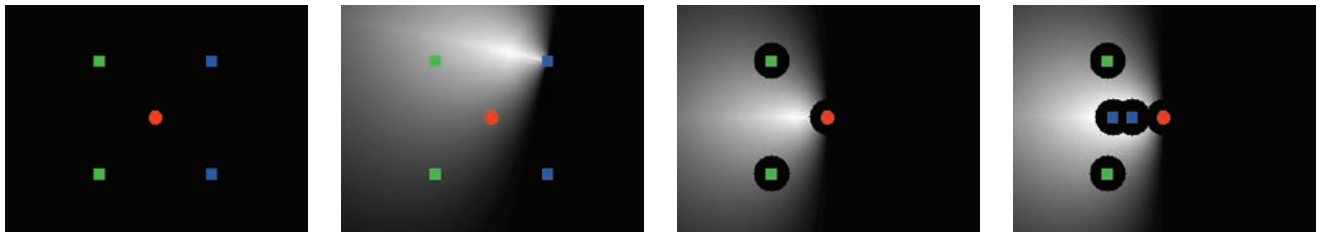
The next step is to generate new waypoints that can be used to satisfy the goal state. This is done by grounding the landmark and target elements of the goal state in the information from vision. The target group ("the blue cubes") is grounded by counting how many of the visible objects match this description. Because there are two objects that match the colour and shape of the objects described by the human, two new waypoints are generated at the specified spatial relationship to the landmark group. The waypoint for the landmark object is identified (in this case `wp_d9`), and then the new waypoints must be placed as dictated by the appropriate potential field. In this case a projective field is generated around the red ball's waypoint, with the non-target objects (the green cubes) as distractors. This field can be seen in Figure 8(c). The new waypoint positions are selected by picking the points in the field with the highest values (and inhibiting the area around the points selected). This final step is presented in Figure 8(d), and results in the following extra information being added to the mapping: `wp_d10 (172,200), wp_d11 (156,200).`

To complete the planning problem, its initial state is extended with `left_of` propositions describing the spatial relations of the newly generated empty waypoints to the already occupied ones. Finally the FF planner is run, returning:

```
0: pickup robot obj_d0 wp_d1
1: put robot obj_d0 wp_d11
2: pickup robot obj_d2 wp_d3
3: put robot obj_d2 wp_d10
```

Although the plan looks simple in this example, note that the referential constraints in the goal description (cf. Figure 3) are correctly resolved: the two *blue* blocks are picked up. Note further that even this problem contains non-trivial causal constraints between actions to which the planner automatically adheres: neither does it try to pick up several objects at once, nor does it place several objects on the same waypoint.

Before plan execution, the plan must be updated to include information about the current scene. This is done by querying the mediation process to determine the objects from vision referred to by the object IDs. Using this information the manipulation system acts out the human's command by picking up each blue cube in turn and placing them at the points indicated in Figure 8(d), resulting in the scene in Figure 7(b).



(a) The initial state. (b) The potential field for "left" for the top right cube. (c) The potential field for "left" with two target cubes removed. (d) The target cubes placed at the highest points in the field.

Figure 8: The progression of generated potential fields in for the processing of "put the blue cubes to the left of the red ball". The two squares on the left represent green cubes, whilst the two on the right represent blue ones.

9 Conclusions and Future Work

In this paper we presented a novel approach to mediating between quantitative and qualitative representations for a robot that must follow commands to perform manipulative actions. Within this approach we have demonstrated two novel techniques: a generic method for the interpretation of natural language action commands driven by planning operators that enables "lazy" resolution of referential expressions by a planner; and the task-orientated use of potential field models to both automatically generate waypoints in real space that the planner can use to solve an under-constrained problem, and to add spatial relationships between existing waypoints.

As this approach is still in its early stages there are a number of features we would like to add to it. These include optimisation functions for the planner, possibly based on spatial knowledge; more robust methods for placing waypoints in potential fields, perhaps using local search; and methods of detecting failure when no waypoints can be placed, or no plan can be found. In this latter case there are a number of alterations that can be made to the state generation process that may allow a plan to be found, even if it is not of a high quality. Future scenarios for our robot will consist of multi-step mixed-initiative interactions with humans. To this end we want to extend our mediation methods to support the generation of descriptions for spatial configurations and plans.

Acknowledgements

This work was supported by the EU FP6 IST Cognitive Systems Integrated project Cognitive Systems for Cognitive Assistants "CoSy" FP6-004250-IP. The authors would like to acknowledge the impact discussions with other project members have had on this work, and also thank Mark Roberts for contributing to the development of the integrated system.

References

[Allen *et al.*, 2001] J. Allen, D. Byron, M. Dzikovska, G. Ferguson, and L. Galescu. Towards conversational human-computer interaction. *AI Magazine*, 2001.

[Brenner, 2005] M. Brenner. Planning for multiagent environments: From individual perceptions to coordinated execution. In *ICAPS-05 Workshop on Multiagent Planning and Scheduling*, 2005.

[Costello and Kelleher, 2006] F. Costello and J. Kelleher. Spatial prepositions in context: The semantics of near in the presence of distractor objects. In *Proc. ACL-Sigsem WS on Prepositions*, 2006.

[Fox and Long, 2003] M. Fox and D. Long. PDDL 2.1: an extension to PDDL for expressing temporal planning domains. *JAIR*, 20:61–124, 2003.

[Hoffmann and Nebel, 2001] J. Hoffmann and B. Nebel. The FF planning system: Fast plan generation through heuristic search. *JAIR*, 14, 2001.

[Kelleher *et al.*, 2006] J. Kelleher, G.J. Kruijff, and F. Costello. Proximity in context: an empirically grounded computational model of proximity for processing topological spatial expressions. In *Proc. ACL-COLING '06*, 2006.

[Kruijff *et al.*, 2006] G.-J. M. Kruijff, J. D. Kelleher, and N. Hawes. Information fusion for visual reference resolution in dynamic situated dialogue. In *PIT '06*, 2006.

[Logan and Sadler, 1996] G.D. Logan and D.D. Sadler. A computational analysis of the apprehension of spatial relations. In M. Bloom, P.and Peterson, L. Nadell, and M. Garrett, editors, *Language and Space*. MIT Press, 1996.

[Mavridis and Roy, 2006] N. Mavridis and D. Roy. Grounded situation models for robots: Where words and percepts meet. In *Proc. IROS '06*, 2006.

[Mcguire *et al.*, 2002] P. Mcguire, J. Fritsch, J. J. Steil, F. Rothling, G. A. Fink, S. Wachsmuth, G. Sagerer, and H. Ritter. Multi-modal human-machine communication for instructing robot grasping tasks. In *Proc. IROS '02*, 2002.

[Olivier and Tsujii, 1994] P. Olivier and J. Tsujii. Quantitative perceptual representation of prepositional semantics. *Artificial Intelligence Review*, 8(147-158), 1994.

[Regier and Carlson, 2001] T Regier and L. Carlson. Grounding spatial language in perception: An empirical and computational investigation. *Journal of Experimental Psychology: General*, 130(2):273–298, 2001.

[Sidner *et al.*, 2003] C.L. Sidner, C.H. Lee, and N. Lesh. The role of dialog in human robot interaction. In *Int. Workshop on Language Understanding and Agents for Real World Interaction*, 2003.