Classical Planning with Communicative Actions

Citation for published version:
Marques, T & Rovatsos, M 2016, Classical Planning with Communicative Actions. in ECAI 2016: 22nd European Conference on Artificial Intelligence, 29 August - 2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016). vol. 285, Frontiers in Artificial Intelligence and Applications, vol. 285, The Hague, The Netherlands, pp. 1744-1745, 22nd European Conference on Artificial Intelligence, The Hague, Netherlands, 29/08/16. DOI: 10.3233/978-1-61499-672-9-1744

Digital Object Identifier (DOI):
10.3233/978-1-61499-672-9-1744

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
ECAI 2016

Classical Planning with Communicative Actions

Tânia Marques and **Michael Rovatsos**[1]

Abstract. Explicit communication planning is an increasing necessity for agent systems. Furthermore, there has also been a renewed interest in using classical planning to perform this planning in addition to physical actions in a goal-directed way. Existing approaches, however, are not applicable to a broad spectrum of domains. We present several generic pre-processing strategies and adaptations of the Fast-Forward (FF) planner that we compare in two different domains in terms of time complexity.

1 Introduction

Recently, there has been an increasing interest in planning communication explicitly, contrasting with traditional approaches where communication is disregarded [3] or implicit [7, 12]. In fields where agents communicate with humans, the relevance of explicit planning communication over the task is obvious, but even in more artificial settings, this has been shown to be increasingly important, especially for tasks where information exchange is not naturally implicit [2] or tasks where the agents have different sets of beliefs [11].

Symbolic approaches to explicitly plan communication make it easy to incorporate the semantics of speech acts commonly used in agent communication languages [4], and allow for expressing the mechanics of communication in a domain-independent way that avoids enumerating all possible messages at design time. As a consequence, there has been renewed interest in symbolic approaches for dialogue planning (e.g. [9, 13]), and epistemic planning that can be employed in communicative situations to model uncertainty over other agents' mental states, which determine their expected behaviour towards the planning agent [5, 11].

In this paper, we investigate how a symbolic planner can be used to incorporate explicit communicative actions, while keeping the common assumptions of classical planning and its great strength of being generic and applicable to a broad range of domains. To focus on the fundamental problem of modeling communicative actions in such a way that they can be seamlessly combined with domain actions, we assume a deterministic setting where the other agent will always accept any request coming from the planning agent. While ignoring the uncertainty inherent to communication, this allows us to expose the complexity problems that arise even in this simple case without being biased by additional layers of complexity raised by contingency and probabilities. We focus on two broad classes of methods that extend classical planning to support communicative actions: pre-processing methods that can be easily implemented without changing the planner itself; and an integration of communication inside a classical planners that helps us exploit the machinery it provides. We present different variations of these methods and show how using the planner machinery is important to reduce the complexity of search, independently of the domain.

2 Integrating Communication

Our planning approach resembles speech act theory [1, 4], where utterances are actions whose effects are transfers of beliefs. If we assume two agents $a, b \in Agents$ and a set of actions A, divided into physical ground actions $Acts$ and communicative ground actions $Coms$. Then, a communicative action is a planning action defined by its name and a set of predicates as preconditions and effects taken from a set of *communicative fluents* such as `intends` (agent a has intention of performing act *act*) and `knows_wants` (agent b knows that agent a wants *act* to be performed) which allows us to build "mediating actions" following Cohen and Perrault's [4] concept, stating that a "mental" action will cause an intention in the hearer, and consequently to the actual performance of the action intended by the speaker. As arguments the communicative actions will take agents $requester, receiver \in Agents$ and any grounded physical action $request \in Acts$. We assume that only grounded physical actions can be taken as arguments. This makes our communicative actions *second-order actions*, but with a depth limit of one. Assuming this construction, actions *request* and *accept* would look as follows in PDDL [10]:

```
(:action request
:parameters   (?requester - agent ?receiver - agent ?request - act)
:precondition (and (not (= ?requester ?receiver))  (me ?requester))
:effect       (knows_wants ?receiver ?requester ?request))

(:action accept
:parameters   (?receiver - agent ?requester - agent ?request - act)
:precondition (and (not (= ?receiver ?requester))  (not (me ?recceiver))
(knows_wants ?receiver ?requester ?request))
:effect       (intends ?receiver ?request))
```

The integration of communicative actions can be done in a pre-processing phase or can be achieved by incorporating them inside the planning algorithm. The latter is more complex, because it assumes knowledge of the planner, but it allows the use of the machinery already provided by the planner.

2.1 Pre-Processing Methods

We present three possible pre-processing methods to add communication to classical planning.

- **Specific Communicative Actions:** This method automatically creates *request* and *accept* communicative actions that are tailored specifically to each physical actions contained in the domain file.
- **Grounded Physical Actions:** Here, the physical actions and the objects are used to generate all the possible objects of the type *act* that can be used in the communicative actions. An action is also generated which is an interface between the static object created and the possibility of using it in planning as a physical action.
- **Conditional Actions:** Similar to the previous one, but instead of using an action for every static object of the type *act*, we use an encoding in a single action that has *when* in its effects. Basically it works as a *switch* to decide which effects as an action are associated to a specific act object.

---

[1] The University of Edinburgh, email: {tmarques,mrovatso}@inf.ed.ac.uk

2.2 Adapting the planning algorithm

Two adaptations of the FF planner [8] were implemented, each based on a specific way of using the original planner:

- **Action Templates:** FF has already machinery in place to verify which objects can fill the schematas or not based on several factors. This method integrates communication into the FF code to allow the use of this machinery to define the communicative. actions to be created.
- **Relevant facts:** FF has also a reachability analysis to choose the relevant facts (facts that are reached with the actions provided). If an action does not reach any relevant fact, then we disregard it in this method by not creating the respective communicative action.

3 Experimental Results

3.1 Domains

We explored two domains: the Colored Trails [6] and the Deliver Letters domain. In the Colored Trails domain, agents play a game on a grid-like board with coloured squares. Each player needs to move to a goal location, but can only move to an adjacent square if it owns a chip of the same colour as that square, and the agent "spends" that chip. In ou scenario, we consider single row grid, where the requester is in the leftmost location and its goal is located at the opposite end. The requester has no chips, which are all with the receiver, so the agent will have to request the exchange of a number of chips equal to the length of the row. In the Deliver Letters domain, two agents are in a initial storage place and receive letters to deliver to the post offices. Every agent can only deliver letters to a certain post office, but may receive letters directed to any of them. For simplicity, we assume the requester only receives letters it cannot deliver, so it has to request their delivery from the other agent. All instances have two agents, three locations and the number of letters is a power of 2. Both the domains seem similar, but they are conceptually different. While the importance of a physical action in Colored Trails is defined by their potential use during planning, in Deliver Letters, the relevance of an action is completely defined by the object (the letter *ID*). There are eight instances for each domain that vary in terms of the size of the world (squares and letters) by powers of 2 from 4 to 512. These instances were run on a Intel Core i7-4770 CPU @ 3.40GHz.

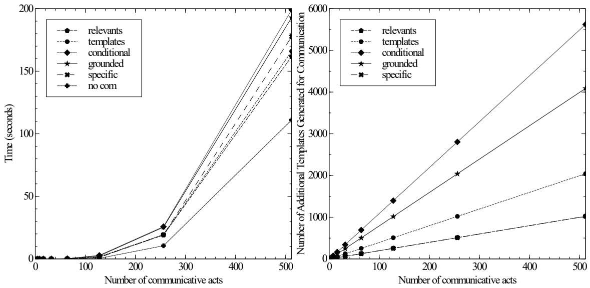

Figure 1. Time performance in Colored Trails, when the number of communicative actions equals the number of chips. Left: runtime; Right: number of additional templates created compared to using no communication.

The plots in figure 1 and 2 present the time taken for each approach and the number of additional grounded actions (templates) created when compared with having no communication (*no com*).

The results are consistent across all scenarios. The *grounded* physical actions and *conditional* cations approaches always produce the worst performance (and higher number of templates), while the *specific* communicative actions and *relevant* facts approach had the best performance. Time was mainly spent on the search, and adding communicative actions had the greatest effect on time when the actions can only be identified as relevant in the search as in Colored Trails.

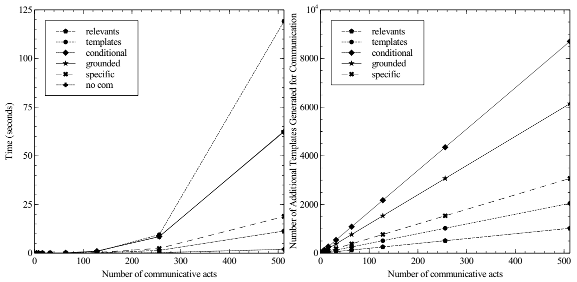

Figure 2. Plot with the time performance and additional templates obtained in the Delivering Letters domain.

4 Conclusions

We have shown that using specific actions or generating well-informed templates are the best approaches for integrating communication in classical planning. This is due to the link between communicative and physical actions which makes the search process critical in reducing the number of communicative actions to be considered. In the future, it will be interesting to find solutions to fully integrate communication in the plan search, and to test different methods in more complicated plan settings such as contingent planning.

ACKNOWLEDGEMENTS

The research presented in this paper has been funded by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 607062 *ESSENCE: Evolution of Shared Semantics in Computational Environments* (`http://www.essence-network.com/`).

REFERENCES

[1] J. L. Austin, *How to do things with words*, Oxford, Clarendon, 1962.
[2] T. Balch and R. C. Arkin, 'Communication in reactive multiagent robotic systems', *Autonomous Robots*, **1**(1), 27–52, (1994).
[3] R. I. Brafman and C. Domshlak, *From One to Many: Planning for Loosely Coupled Multiagent Systems*, 28–35, ICAPS'08, AAAI, 2008.
[4] P.R. Cohen and C.R. Perault, 'Elements of a plan-based theory of speech acts', *Cognitive Science*, **3**, 177–212, (1979).
[5] T. Engesser et al., *Cooperative epistemic multiagent planning with implicit coordination*, 68–77, DMAP'15, 2015.
[6] Y. Gal et al., *Colored Trails: a Formalism for Investigating Decision-Making in Strategic Environments*, 25–30, Workshop on Reasoning, Representation, and Learning in Computer Games, AAAI, 2005.
[7] M. R. Genesereth et al., 'Cooperation without communication', *Heuristic Programming Project*, 51–57, (1984).
[8] J. Hoffman, 'The fast-forward planning system', *AI*, **22**(3), 57–62, (2001).
[9] A. Koller and R. P. Petrick, 'Experiences with planning for natural language generation', *Computational Intelligence*, **27**(1), 23–40, (2011).
[10] D. McDermott et al., 'The PDDL planning domain definition language', *AIPS98 Planning Competition Comitee*, (1998).
[11] C. Muise et al., *Towards Team Formation via Automated Planning*, COIN'15, 2015.
[12] E. Pagello et al., 'Cooperative behaviors in multi-robot systems through implicit communication', *Robot Auton Syst*, **29**(1), 65–77, (1999).
[13] M. Steedman and R. P. A. Petrick, *Planning dialog actions*, 265–272, SIGdial'07, 2010.