Social Continual Planning in Open Multiagent Systems: a First Study

Matteo Baldoni, Cristina Baroglio, Roberto Micalizio

Università degli Studi di Torino — Dipartimento di Informatica
firstname.lastname@unito.it

Abstract. We describe a Multiagent Planning approach, named Social Continual Planning, that tackles *open* scenarios, where agents can join and leave the system dynamically. The planning task is not defined from a global point of view, setting a global objective, but we allow each agent to pursue its own subset of goals. We take a *social* perspective where, although each agent has its own planning task and planning algorithm, it needs to get engaged with others for accomplishing its own goals. Cooperation is not forced but, thanks to the abstraction of *social commitment*, stems from the needs of the agents.

Keywords: Continual Planning, Multi-agent Planning, Social Commitments

1 Introduction

The ability to plan one's own activities, even in dynamic and challenging scenarios such as Multiagent Systems (MAS), represents a key feature in many real-world applicative domains (see e.g., logistics, air traffic control, rescue missions, and so on). Not surprisingly, planning in MAS is drawing the attention of an ever growing number of researchers, as witnessed by the new series of *Distributed and Multi-Agent Planning Workshops* hosted by ICAPS.

The term Multiagent Planning (MAP) refers to a planning task in which a set of planning agents, each equipped with its own tools and capabilities, has to synthesize a *joint solution* (i.e., a joint multiagent plan). The planning task usually involves a number of interdependent subgoals, so that some form of coordination among the agents is necessary to solve the problem. Different methodologies have been proposed in the literature. Besides centralized approaches (e.g., [3]), which fall outside the above notion of MAP, the other distributed solutions can be categorized into three main families, depending on when the coordination among the agents is actually performed: *after* the planning phase [7], *interleaved* with the planning process [8,9,13], or *before* the planning search [6].

In all the above approaches, the planning task defines a global objective to be achieved by means of a "joint solution" involving the capabilities of the agents. Moreover, the set of agents to be involved is known in advance and cannot change during the planning process; the system is therefore closed. In this paper we deal with a *different planning problem*, and propose a methodology named *Social Continual Planning* (SCP), to tackle it. We consider the planning problem of an agent situated in an open multiagent system. The agent may resort on other agents for solving a task of its own interest. The agent plans both its own actions, and its interactions with others whenever it is not capable, or it deems as not convenient, to execute certain steps in autonomy. The focus is not on negotiation, but on the framework through which an agent seeks the help by the others, and on the *engagements* that bind agents to supporting each other. Interaction is not limited to communication but it is a process through which the involved agents progress each in the solution of its own task. Engagements are binary social relationships, that are established dynamically and that create expectations on the involved agents behavior. An agent autonomously decides (plans) when to bind to another one to do something.

More precisely, we take a *social* perspective in the sense that, even though each agent has its own planning task and uses its own planning algorithm, the agent has still to get engaged with others in order to accomplish its own goals. The interactions that an agent has with others will, in general, allow both parties to get closer to their own goals. Cooperation is not forced to the agents just because their are part of the system, but rather cooperation stems from the needs of the agents within the system, and endures as far as the parties take advantage of it. In other terms, we propose a form of (agent) planning which is *situated* in a multiagent system, where an agent not only has to plan its own actions, but has also to plan its social relationships with other agents. Since the coordination has to be planned, it must be supported by a proper abstraction that enables one agent to create expectations about the behaviors of others. To this end, in this paper we adopt *social commitments* [14]. Interestingly, a recent work by Telang et al. [16] shows how goals and commitments are strongly interrelated by means of a set of practical rules. This supports our intuition that commitments may play a central role, together with beliefs and goals, in the synthesis of a plan in a multiagent setting. [2] describes an early implementation, and exemplifies the approach in a logistic scenario.

2 Related Work and Background

Multiagent Planning. To the best of our knowledge, the SCP problem has not been tackled in the literature, so far. SCP is however rooted in multiagent and distributed planning that, since the seminal work by Boutilier et al. [3], has addressed the problem of finding a coordinated, joint solution to a given planning task. In the early, centralized methodologies to multiagent planning, agents are seen as resources to be managed so as to achieve the global goal. More recently, distributed approaches allow the planning search to be distributed among the agents; however, the definition of the planning task is still centralized; see for instance the MA-STRIPS formalization [4]. Distributed approaches can be distinguished on how the planning and coordination phases are actually carried on. First attempts to coordinating plan *after* the planning phases [7] suffered from a sever drawback: whenever conflicts were detected between any two plans, the agents had to revise their plans accordingly. Thus, the domain knowledge about conflicts and constraints was not used actively during the planning phase, but only *a posteriori* to verify the correctness of the joint solution. This drawback is overcome by approaches (see e.g., [8,9,13]) in which the coordination and planning phases are *interleaved*. These approaches rely on the exchange of various kinds of information, such as partial plans, or states inferred during the search, so that conflicts are discovered as soon as possible, and corrections can be made while the planning phase is still in progress. A last family of approaches set the coordination phase *before* the planning one (see e.g., [6]). Such solutions, however, assume that all the possible conflicts are known in advance and globally defined.

Commitments. As stated in the introduction, SCP is a novel methodology of planning driven by social engagements; in particular, in this paper we focus on social engagements that can be modeled as *social commitments* (simply commitments below), first introduced in [14]. Commitments arise, exist, are satisfied, revoked, or otherwise manipulated, all in a social context (i.e., *social state* below); commitments have therefore a *life cycle* that evolves as a consequence of the operation performed by agents on them [11,15,12]. More formally, a commitment $\mathsf{C}(x, y, s, u)$ formalizes a relationship between an agent x, playing the role of *debtor*, and another agent y, playing the role of *creditor*: the debtor is committed towards the creditor to bring about a consequent condition u, whenever an antecedent condition occurs s. Antecedent and consequent conditions are conjunctions or disjunctions of events and commitments and they concern only the *observable behavior* of the agents.

Notably, there have been some recent attempts to integrate commitments in planning problems, see [11,15,12] which as well as our work rely on the rules proposed in [16]. The idea of translating pragmatic rules into a planning language is first proposed in [15], where the Hierarchical Task Network (HTN) formalization is used. HTNs, however, are used at design time to model and verify commitment protocols [11]; thus, the point of view of these works is still centralized. In this work we will consider a STRIPS-like representation of the pragmatic rules, and use them for generative planning in a context where a centralized point of view is missing. In other terms, in this paper the interactions via commitments are not outlined within pre-designed HTNs, but have to be discovered at execution time by the planning search.

Goal Formalization. The notion of *goal* plays an important role not only from the point of view of planning, but also in general whenever one has to design and develop intelligent agents. In this paper, we take advantage of the formalization initially proposed in [17], and subsequently revised in [16]; specifically, a goal G is a tuple $\mathsf{G}(x,$ $p,$ $r,$ $q,$ $s,$ $f)$, where x is the agent pursuing G, p is a precondition that must be satisfied before G can be considered active, r is an invariant condition that holds until the achievement of G, q is a post-condition (effect) that becomes true when G is successfully achieved, and finally, s and f are the success and failure conditions, respectively. As well as commitments, goals have a life cycle in which state transitions are triggered by the execution of proper goal actions [16].

Pragmatic Rules. The relation between goals and commitments has been studied in [16], and it has been formalized in terms of practical rules, which capture patterns of pragmatic reasoning, in terms of changes to the configuration of an agent. Specifically, the configuration of an agent x is the tuple $S_x = \langle \mathcal{B}, \mathcal{G}, \mathcal{C} \rangle$ where $\mathcal{B}$ is its set of beliefs about the current snapshot of the world, $\mathcal{G}$ is the set of agent's goals, and $\mathcal{C}$ its set of commitments; i.e., commitments in which x is involved either as debtor or as creditor.

The operational semantics of pragmatic rules is given via guarded rules in which S_i are configurations having form $\frac{S_1 \longrightarrow S_2}{guard}$; where $guard$ is a condition over the current agent's beliefs and commitments; whereas $S_1 \longrightarrow S_2$ is a state transition involving a change in the state of commitments or goals; usually it corresponds to an operation on goals or commitments. Pragmatic rules are distinguished into: (1) rules from goals to commitments, they involve commitments that are used as a means to achieve some goal; and (2) rules from commitments to goals, they involve goals that are used as a means to achieve either the antecedent (if the agent at issue is debtor) or the consequent (if creditor) condition of a commitment. Some rule examples are reported in Table 1 (subscripts denote the state of commitments and goals as discussed in [16]).

For instance, the ENTICE rule tackles the situation in which (only) by creating the commitment can the agent satisfy its goal: If G is active and C is null, x creates an offer to another agent. The DELIVER rule, on the other hand, allows an agent to discharge one of its commitments by activating the goal appearing in the consequent condition: If C becomes detached (i.e., goal G_2 has been satisfied), then debtor x activates a goal G_1 to bring about the consequent.

3 The Social Continual Planning Problem

A *Social Continual Planning (SCP) system* is an open environment inhabited by heterogeneous and independent agents. Each agent has its own planning task, and can perform a specific set of actions. A *Social Continual Planning Problem* is a planning problem of an agent, situated within an SCP system, which, for being solved, requires the agent to plan also a set of engagements, realized as social commitments, with other agents in the system. Agents can join and leave the system dynamically; however, we assume that no agent leaves the system as long as there are active commitments involving it either as debtor or as creditor. More formally, an SCP system is a tuple $\langle \mathcal{U}, \mathcal{A}, \mathcal{S} \rangle$ where:

- $\mathcal{U}$ is a finite set of propositional atoms, whose truth value can be observed by all the agents in the SCP; $\mathcal{U}$ represents a sort of common language through which agents can interact. Atoms in this set are used to describe the state of the environment shared by the agents. In addition, these are the atoms that can appear as antecedents and consequents of the commitments.
- $\mathcal{A}$ is a set of agents; each agent $i \in \mathcal{A}$ is associated with a configuration which extends the agent configuration we have already introduced. Specifically, the agent configuration for agent i is a tuple $\langle \mathcal{B}^i, \mathcal{G}^i, \mathcal{C}^i, Acts^i, Socs^i \rangle$: $\mathcal{B}^i$, $\mathcal{G}^i$, and $\mathcal{C}^i$ are as before; whereas:
  - $Acts^i$ is a set of actions agent i can perform; it is partitioned into:
    - Φ^i is a set of "physical" actions; as usual, these actions are defined in terms of preconditions and effects, which can be both conditions on environment

Table 1. Examples of practical rules from [16]

Goals-to-Commitments	$\frac{\langle G^A, C^N \rangle}{\mathsf{create}(C)}$	ENTICE	$\frac{\langle G^{T \vee F}, C^A \rangle}{\mathsf{cancel}(C)}$	WITHDRAW OFFER
Commitments-to-Goals	$\frac{\langle G_1^N, C^D \rangle}{\mathsf{consider}(G_1) \wedge \mathsf{activate}(G_1)}$	DELIVER	$\frac{\langle G_2^N, C^C \rangle}{\mathsf{consider}(G_2) \wedge \mathsf{activate}(G_2)}$	DETACH

atoms (i.e., in $\mathcal{U}$) or on internal (agent-dependent) atoms that are not globally traced (i.e., the internal state of an agent is private).

    - $*$ Σ^i is a set of *social actions*; preconditions and effects are defined in terms of goals in $\mathcal{G}^i$ and commitments in $\mathcal{C}^i$. More precisely, each social action corresponds to a pragmatic rule from goals to commitments. Indeed, we consider these pragmatic rules as actions because, as we discuss below, they can be used by an automated planner to plan interactions with other agents. Note that while goals in $\mathcal{G}^i$ are *private* (only agent i can see and manipulate them), commitments in $\mathcal{C}^i$ have a *social value*: whenever i changes the state of a commitment in $\mathcal{C}^i$, this change becomes visible to all the other agents in the system (see $\mathcal{S}$).
  - • $Socs^i$ is a the set of pragmatic rules from commitments to goals adopted by an agents; from our point of view these rules define the *social strategy* of agent i. Thus, these rules are not used during the planning search, but rather to decide which goals should be pursued.
- $\mathcal{S}$ is the social state shared by all the agents in the SCP system at hand. The social state can be partitioned into two subsets:
  - • $\mathcal{S}^{\mathcal{C}}$ is the set of all the active commitments defined between any two agents in $\mathcal{A}$; in particular, for each agent $i \in \mathcal{A}$, $\mathcal{C}^i \subseteq \mathcal{S}$, $\mathcal{C}^i$ is the projection of $\mathcal{S}$ over all the commitments in which i appears either as debtor or as creditor.
  - • $\mathcal{S}^{\mathcal{E}}$ is the set of all the propositional atoms describing the environment that hold at a given time; in particular, $\mathcal{S}^{\mathcal{E}} \subseteq \mathcal{U}$.

Given an SCP system $\langle \mathcal{U}, \mathcal{A}, \mathcal{S} \rangle$, let $i \in \mathcal{A}$ be an agent, that is described by the tuple $\langle \mathcal{B}^i, \mathcal{G}^i, \mathcal{C}^i, Acts^i, Socs^i \rangle$. An SCP problem for i amounts to finding a plan, composed by $Acts^i$ and $Socs^i$, to achieve $\mathcal{G}^i$ starting from $\mathcal{B}^i$. In particular:

- $\mathcal{B}^i$ is the initial state of the planning task i is responsible for; such a state is a set of atoms possibly occurring in $\mathcal{U}$, but also occurring in a private set of atoms describing the internal state of i, and hence these atoms are not traced within the SCP system. We only assume that i joins the SCP system iff $\mathcal{S} \cup \mathcal{B}^i \not\models \bot$.
- $\mathcal{G}^i$ is a list of goals the agent has to achieve; each goal can be an atom or a conjunction of atoms in $\mathcal{U}$ and possibly in the private set of agent's atoms. Note that, differently from classical planning, it is not required that all the goals in $\mathcal{G}^i$ hold in a unique system state.
- $\mathcal{C}^i$ is initially empty.
- Φ^i is a set of domain-dependent actions agent i can directly perform whenever their preconditions hold. For instance, in a logistic domain, a truck-agent can perform action `drive`, whereas a plane-agent can `fly`.
- Σ^i can be initialized in different ways; in fact, differently from Φ^i, this set needs not to be static; on the contrary, it could change over time according to contextual conditions. In our preliminary implementation, we have adopted a very simple solution. Let us consider the ENTICE rule above[1]. The objective of this rule is to create a commitment of the form $C(i, j, s, u)$, in order to "entice" another agent j to bring about s, which is of interest for i. At this initial stage, however, i cannot

[1] Other rules are treated consequently.

Algorithm 1 Social Continual Planning Strategy

```
SCP-Strategy($\mathcal{B}^i, \mathcal{G}^i, \mathcal{C}^i, Acts^i, Socs^i$)
1. while $\mathcal{G}^i \neq \emptyset \vee \mathcal{C}^i \neq \emptyset$ do
2.     on $\mathcal{S}$ change update $\mathcal{G}^i$ using $Socs^i$
3.     $g \leftarrow$ pick up a goal from $\mathcal{G}^i$
4.     $\pi \leftarrow$ plan to $g$
5.     $status \leftarrow$ execute $\pi$
6.     if $status$ equals $success$ then
7.         $\mathcal{G}^i \leftarrow \mathcal{G}^i \setminus \{g\}$
8.     end if
9. end while
```

know which condition u is of interest for j. Surely enough, i knows which atoms it can directly achieve by performing its physical actions. Thus, for each atom $s \in \mathcal{U}$ such that s never appears as an effect of any action in Φ^i, agent i creates a template `entice-`s whose effect is the creation of a commitment $C(i, \_, s, u)$, where $\_$ denotes any agent willing to satisfy s, and u is any atom in $\mathcal{U}$ that appears in the effects of at least one physical action in Φ^i. Of course, since the `entice-`s template can be instanced in different ways, depending on the actual u condition, agent i will offer first the conditions, that from its point of view, are the cheapest to achieve.

- $Socs^i$ is a static set of rules, decided at design time, that defines the social behavior of i; namely, how an agent is reliable for bringing about the consequent and antecedent conditions of the commitments in $\mathcal{C}^i$.

Social Continual Planning: the Strategy. The SCP strategy we propose, sketched in Algorithm 1, is a form of continual planning (see e.g., [5]) in which generative planning is interleaved with plan execution. The main difference with other approaches is that to achieve a goal, an agent plans not only its own actions, but also its engagements with others, and depending on how these interactions carry through, the agent may decide to perform some replanning or to pursue a different goal.

An agent i follows the SCP strategy as far as there are goals in $\mathcal{G}^i$ to be achieved or $\mathcal{C}^i$ is not empty. This second condition assures that an agent does not leave the system when it is still involved in some active commitments.[2] At each iteration, the agent checks for updates in the social state $\mathcal{S}$ (line 2); any change occurring in $\mathcal{S}$, in fact, can have an impact on the set $\mathcal{G}^i$ of goals. For instance, a new commitment $C(j, \_, s, u)$ appearing in $\mathcal{S}^{\mathcal{C}}$ could draw the attention of agent i when u is a condition that i needs but it cannot achieve on its own, and at the same time i knows how to obtain s. In such a case, i could accept to be the *creditor*: s is added to $\mathcal{G}^i$ (i will eventually bring about s). On the other hand, the occurrence of a new atom in $\mathcal{S}^{\mathcal{E}}$ could make the achievement of

[2] In principle, an agent may remaining situated within the system indefinitely, waiting for agents to cooperate with. For example, in a logistic domain, a shipper has the high-level objective of earn money by offering its transportation facilities. This objective does not immediately translate into an initial goal $\mathcal{G}$, but rather it is better modeled in terms of pragmatic rules (i.e., both social actions in Σ, and behavioral rules in $Socs$), so as the shipper is willing to accept requests from other agents, but also offers itself shipment services to others.

a goal g in $\mathcal{G}^i$ no longer necessary, so g is dropped. Of course, these agent's decisions are driven by the $Socs^i$ behavioral rules.

Once $\mathcal{G}^i$ has been updated, agent i selects one goal g from $\mathcal{G}^i$ (line 3); and synthesizes a plan π reaching g (line 4). It is worth noting that any off-the-shelf planner can be used to synthesize π since from the point of view of the planner there is no distinction between social and physical actions (both kinds of actions are translated into PDDL, see below). We only assume that in case the used planner produces a partial-order plan (POP), π is one of the possible linearizations of such a POP.

After the planning step, the agent can start the execution of π (line 5), which contains both physical actions in Φ^i, and social actions in Σ^i. The execution of π proceeds one action a at a time and in the order. If a is a physical action, it is immediately executed, and its effects on atoms in $\mathcal{U}$ are made available to all the other agents via $\mathcal{S}^{\mathcal{E}}$. If a is a social action, e.g., an entice-s action, the action execution affects $\mathcal{S}^{\mathcal{C}}$ with the addition of a new commitment $C(i, \_, s, u)$, which has to be picked up by some other agent. The execution of π is therefore suspended; indeed, the entice-s action is part of π only in case the atom s is a precondition for some subsequent action, and hence the plan execution cannot proceeds without s. In case an agent j is interested in u, it accepts the offers by finalizing the commitment in $C(i, j, s, u)$, and eventually it will bring about s. As soon as s is satisfied, i proceeds with the execution of its plan (u will be added to $\mathcal{G}^i$ the next time i checks for changes in S). When all the actions in π are performed, the execution phase terminates in *success* state (i.e., g has been achieved), and hence g is removed from $\mathcal{G}^i$ (line 7).

However, it is also possible that no agent is interested in the service u offered by i. To avoid an indefinite wait, i sets up a timer. As soon as the time runs out, the commitment is canceled from $\mathcal{S}^{\mathcal{C}}$, and the plan execution terminates with a *failure* state. Since g has not been achieved, it is not removed from $\mathcal{G}^i$. At the next iteration of the strategy, i first checks whether g is still required (line 2), and then tries to find an alternative plan reaching it (line 4) that may require a different instantiation for the entice-s action (i.e., with a different condition offered as consequent of the commitment).

Intuitively, the correctness of the approach relies on the coherence and convergence properties discussed in [16]. In particular, the goal convergence property states that in the situation in which agent i has a goal G_1=$\mathsf{G}(i, p_1, r_1, q_1, s, f_1)$, another agent j has a goal G_2=$\mathsf{G}(j, p_2, r_2, q_2, s, f_2)$, and there exists a commitment $C_1 = C(i, j, s, u) \in \mathcal{S}^{\mathcal{C}}$, then, there is a finite sequence of pragmatic rules that leads to $G2$'s state equaling G_1's state. This means that whenever agent j brings about s, satisfying its internal goal G_2, then, also agent i has its own goal G_1 indirectly satisfied. This demonstrates the correctness of the SCP strategy in the sense that whenever a plan π, synthesized by i, contains an entice action entice-s, which actually creates the commitment C_1, then the plan is:

1. *feasible*: no action a in π has open preconditions (i.e., atoms that are neither provided by the initial state nor by any previous action); this implies that the preconditions that agent i cannot directly produce, are obtained via cooperation with others;
2. *correct*: if each action is performed successfully, g holds in $\mathcal{S}^{\mathcal{E}}$ at the end of π; as noted above, the execution of social action implies the cooperation with other agents.

4 Discussion and Conclusions

In this paper we addressed the SCP problem, and proposed the SCP strategy as a possible solution. Differently from MAP approaches, where a predefined team of agents has to find a *joint* plan solving a given planning task, here we deal with situations in which each agent is given a planning task which is independent of the others' ones. The challenge, thus, is not to find a joint plan, but to find a plan for each agent that solves the agent's planning task taking advantage of the cooperation with other agents. Moreover, agents are free to join and leave the system dynamically.

The novelties of our proposal are not limited to the openness of the agent team. While in approaches to MAP agents can be thought of as resources used for solving the given planning task, in SCP agents are seen as *autonomous* entities. This change implies that an agent cannot order another agent to do a job, but the agent can just make an offer, and as we have seen, social commitments come at handy to model this kind of relations. More importantly, however, we have to observe that an agent receiving an offer, being an autonomous entity, can accept or reject the offer depending on its contextual conditions and its local goals. A rational agent, in fact, should accept an offer only if the offer brings along some advantages, otherwise the offer should be put aside. It is worth noting how the SCP strategy supports the *decoupling* of agents, that just share environment objects, whereas they are independent for all the other respects. In particular, each agent can implement its social strategy (i.e., pragmatic rules in $Socs$ and Σ) according to local criteria. Moreover, the planning algorithm each agent uses can be tailored to meet optimization functions that are relevant for the agent itself. Note also how the cooperation among the agents do not require that an agent knows the action templates of others (as for instance happens in [13]), and, hence, also the agents' privacy is preserved.

Many lines of research and improvement are possible. In the near future we aim at engineering the implementation of the SCP strategy by exploiting one of the many agents platforms available. In particular, the JaCaMo+ platform [1] seems to be a good candidate since it naturally supports the notions of commitments and social states. In addition, the social behavioral rules in $Socs$ could find an easy implementation as Jason plans (used to program JaCaMo+ agents). Also the integration with a planner does not seem to raise to much troubles; as demonstrated in [10] where Jason plans have been integrated with generative planning.

Acknowledgments

This work was partially supported by the *Accountable Trustworthy Organizations and Systems (AThOS)* project, funded by Università degli Studi di Torino and Compagnia di San Paolo (CSP 2014).

References

1. Matteo Baldoni, Cristina Baroglio, Federico Capuzzimati, and Roberto Micalizio. Programming with Commitments and Goals in JaCaMo+. In *Proc. of the International Conference on*

Autonomous Agents and Multiagent Systems (AAMAS'15), pages 1705–1706. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

2. Matteo Baldoni, Cristina Baroglio, and Roberto Micalizio. Exploring Multiagent Cooperation via Social Continual Planning. In G. Cortellessa, D. Magazzeni, M. Maratea, and I. Serina, editors, *Proc. of 6th Italian Workshop on Planning and Scheduling, IPS 2015*, Ferrara, Italy, September 2015. CEUR-WS.org Workshop Proceedings.
3. C. Boutilier, R. Dearden, and M. Goldszmidt. Stochastic dynamic programming with factored representations. *Artificial Intelligence*, 121(1-2):49–107, 2000.
4. Ronen I. Brafman and Carmel Domshlak. From one to many: Planning for loosely coupled multi-agent systems. In *Proceedings of the Eighteenth International Conference on Automated Planning and Scheduling, ICAPS 2008, Sydney, Australia, September 14-18, 2008*, pages 28–35, 2008.
5. Michael Brenner and Bernhard Nebel. Continual planning and acting in dynamic multiagent environments. *Autonomous Agents and Multi-Agent Systems*, 19(3):297–331, 2009.
6. Pieter Buzing, Adriaan Ter Mors, Jeroen Valk, and Cees Witteveen. Coordinating self-interested planning agents. *Autonomous Agents and Multi-Agent Systems*, 12(2):199–218, 2006.
7. Jeffrey S Cox, Edmund H Durfee, and Thomas Bartold. A distributed framework for solving the multiagent plan coordination problem. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 821–827. ACM, 2005.
8. Edmund H Durfee and Victor R Lesser. Partial global planning: A coordination framework for distributed hypothesis formation. *Systems, Man and Cybernetics, IEEE Transactions on*, 21(5):1167–1183, 1991.
9. Victor Lesser, Keith Decker, Thomas Wagner, Norman Carver, Alan Garvey, Bryan Horling, Daniel Neiman, Rodion Podorozhny, M Nagendra Prasad, Anita Raja, et al. Evolution of the gpgp/taems domain-independent coordination framework. *Autonomous agents and multi-agent systems*, 9(1-2):87–143, 2004.
10. Felipe R. Meneguzzi and Michael Luck. Leveraging new plans in agentspeak(pl). In *Declarative Agent Languages and Technologies VI, 6th Int. Workshop, DALT 2008, Revised Selected and Invited Papers*, volume 5397 of *Lecture Notes in Computer Science*, pages 111–127. Springer, 2008.
11. Felipe R. Meneguzzi, Pankaj R. Telang, and Munindar P. Singh. A first-order formalization of commitments and goals for planning. In *Proc. of the 27th AAAI Conference on Artificial Intelligence*. AAAI Press, 2013.
12. Felipe R. Meneguzzi, Pankaj R. Telang, and Neil Yorke-Smith. Towards planning uncertain commitment protocols. In *Proc. of the 2015 Int. Conf. on Autonomous Agents and Multiagent Systems, AAMAS*, pages 1681–1682. ACM, 2015.
13. Raz Nissim and Ronen I. Brafman. Distributed heuristic forward search for multi-agent planning. *Journal of Artificial Intelligence Research (JAIR)*, 51:293–332, 2014.
14. Munindar P. Singh. An ontology for commitments in multiagent systems. *Journal of Artificial Intelligence in Law*, 7(1):97–113, 1999.
15. Pankaj R. Telang, Felipe R. Meneguzzi, and Munindar P. Singh. Hierarchical planning about goals and commitments. In *Int. conf. on Autonomous Agents and Multi-Agent Systems, AAMAS '13*, pages 877–884. IFAAMAS, 2013.
16. Pankaj R. Telang, Munindar P. Singh, and Neil Yorke-Smith. Relating Goal and Commitment Semantics. In *Post-proc. of ProMAS*, volume 7217 of *LNCS*. Springer, 2011.
17. Michael Winikoff, Lin Padgham, James Harland, and John Thangarajah. Declarative & procedural goals in intelligent agent systems. In Dieter Fensel, Fausto Giunchiglia, Deborah L. Mc Guinness, and Mary-Anne Williams, editors, *Proc. of the 8th Int. Conf. on Principles and Knowledge Representation and Reasoning (KR-02)*, pages 470–481. Morgan Kaufmann, 2002.