Distributed planning in dynamic, semi-trusted and opportunistic environment

Michal Pechoucek[1], **Gerhard Wickler**[2], **Austin Tate**[2] **and Jeff Dalton**[2]
[1]Czech Technical University, Department of Cybernetics, Agent Technology Group
[2]University of Edinburgh, School of Informatics, Artificial Intelligence Application Institute

Abstract: This paper presents formal model of the distributed planning problem in dynamic, semi-trusted and opportunistic environment, suggest an abstract distributed planning architecture, and discusses also the various planning techniques to be further deployed.

1 Introduction

In this paper we will discus the planning problem positioned in very specific environment formulated in parts by the project funding agency[1]. The environment:

- is to be **non-centralized** and with flat organization hierarchy [R1] – the existence of a central coordinating and planning process shall be brought to absolute minimum and the planing knowledge, information about actors skills, resource availability knowledge and goals perception shall be distributed,
- shall provide **partial knowledge sharing** [R2] – the actors in the environment are motivated to keep substantial part of their private planning knowledge and resource availability information undisclosed,
- shall allow **varying interaction availability** [R3] – based on communication infrastructure featuring partial and temporal inaccessibility due to e.g. ad-hoc networking, unreliability of the communication infrastructure or actors to change off-line/on-line status,
- is to be **very dynamic** [R4] where both resource availability as much as goals persistence is expect to be changing between the planning and execution phase, while also during the execution phase, and
- is to be **opportunistic** [R5] – allowing the actors reason about potential goal accomplishment opportunities that may arise in the environment and also consider opportunities of the collaborating actors in the environment.

[1]ERO - European Research Office of US Army

Such a set of requirements is typical for rescue operations, complex humanitarian missions, other OOTW large scale multi-national coalition operations as well as small size military combat ops. Such features are also typical for complete different set of application domains such as virtual organizations and social networking.

1.1 Example Scenario

⊛ [MP]$_1$:Gerhard to take a lead here

Inspired by [20]

1.2 Problem Formalized

⊛ [MP]$_2$:Gerhard to take a lead here

2 Social Knowledge

Prior to discussing the distributed planning architecture, let us briefly introduce the concept of **social knowledge** in distributed multi-agent systems [13]. Social knowledge is a computational model of agents mutual awareness. Social knowledge is usually stored in agents *acquaintance models* and collects the information about the other agents capabilities, resource availabilities, but also their planning restrictions and preferences.

Social knowledge represents both necessary and optional information which an agent needs for its efficient operation in the multi-agent community. Appropriate use of social knowledge improving the quality of collaboration in the community (e.g. interaction efficiency), reduces of communication traffic and accelerates of agents' internal reasoning processes. In competitive environments social knowledge provides the agents with competitive advantage and thus improving individual utility from the interaction, allows agents to reason about the other agents in environments with partial communication accessibility. In semi-trusted environment social knowledge allows for resources protection and improvement of collaboration safety and interaction security.

Social knowledge has been classified in [12] the four levels as follows:

- **minimal social knowledge** – implemented by means of *white-page* (WP) list, collection of information about all members of agent's total neighborhood e.g. information about physical IP addresses, port numbers, ACL language
- **first level social knowledge** – represented by means of the *yellow-page* (YP) list, contains information about services and capabilities provided by the given agent, information about members of agent's cooperative neighborhood and
- **higher level social knowledge** – information about members of agents' monitoring neighborhood:
  - *external social knowledge* – knowledge about agent's outer characteristics such as of agents awareness of agents reliability, maintain and manipulate trust, investigate each other communication, computational and operational load and
  - *meta-knowledge* allows agent private knowledge reconstruction, intentions, other mental models, models of future agents behavior.

Social knowledge can be administered in a centralized way. Such an approach is easy to implement in multi-agent systems. It avoids possible duplication and redundancy and is inevitable at least for the registration phase of the life-cycle of multi-agent systems. The trouble is that any central component may become *bottleneck* in large scale or real-time applications. Social knowledge can be also administered by various dedicated agents that are loosely coupled with the rest of the multi-agent community (such as middle-agents, brokers, matchmakers, mediators) and who provide the yellow pages type of information and negotiate effective forms of cooperation. Social knowledge can be also maintained individually by each particular agent. Such approach can be facilitated by means of *(i) communication* – based on acquaintance model maintenance by means of *pull model* (such as periodical revisions) or *push model* (such as `subscribe-inform` protocol) or *(ii)* by by means of *meta-reasoning* – based on independent monitoring and reasoning about the surrounding agents (different in cooperative and competitive environment).

Social knowledge can be either (*i*) private, (*ii*) public or (*iii*) semi-private [15]. Semi-private knowledge is viewed as available or confidential respectively among different groups of agents or in different interaction situations. Privacy of social knowledge can be violated either *(i) intentionally* by sending and `inform`-type of a message or *(ii)* as a *side-effect* of speech acts such a `cfp`, `propose`, etc.

In the following text we will be working with social knowledge in the form of the information about other agents capabilities, resource availabilities and their planning knowledge.

3 Abstract Distributed Planning Architecture

⊛ [MP]$_3$:all please contribute here, provide feedback, Gerhard include your point of view ...

The problem of distributed planning (DP) has been often discussed in the AI planning and multi-agent research communities recently (e.g. [6], [5], [7], [4]). Distributed planning has been viewed as either (*i*) planning for activities and resources allocated among distributed agents, (*ii*) distributed (parallel) computation aimed at plan construction or (*iii*) plan merging activity. The classical work of Durfee [6] divides the planning process into five separate phases, that will guide our further discussion. We intend to comment and update this DP architecture so that it the requirements listed in 1. The Durfee DP architecture consist of phases as follows:

- $\boxed{\textbf{1}}$ task decomposition
- $\boxed{\textbf{2}}$ subtask delegation
- $\boxed{\textbf{3}}$ conflict detection
- $\boxed{\textbf{4}}$ individual planning
- $\boxed{\textbf{5}}$ plan merging

In the following we will discuss these phases in more details.

3.1 Task Decomposition

Task decomposition (or *global task refinement*) is a classical domain dependent planning problem, that processes nontrivial background knowledge or the data collected or provided by the individual agents. If the right set of data is provided, classical single agent planning technique such as various non-linear planning approaches [11], [21], HTN planning [8] or GraphPlan-like planning [2] can be used.

Collecting and providing the data is a typical problem solved in the multi-agent community. In order to support the requirement for limited knowledge disclosure [R2], the algorithms used in the phase $\boxed{1}$ need to relate closely to the issues of *data/knowledge sharing and disclosure* [15] and *trust and reputation* [9], [14], [16]. A related problem would be of task decomposition in the situation where the *planning knowledge is incomplete* and needs to be communicated among the actors.

Due to the requirements for decentralization [R1] we will be less interested in the situations where the phase $\boxed{1}$ is to be initiated y a single agent (referred to as *centralized DP initiation*). Instead the primary focus of the investigated architecture will be in the situations:

– with highly distributed problem solving knowledge where the phase $\boxed{1}$ will be carried in collaboration among several agents (referred to as *collaborative DP initiation*) but mainly

– with no single centralized goal, while with several mutually interdependent goals coordinated and in parts shared among several collaborating actors collaboration (referred to as *coordinated DP initiation*).

An easier problem is to find whether there exist a task decomposition, given the various constraints represented by planning knowledge. More complex problem arises in the situations where there are various possible decompositions and the most suitable needs to be selected. The problem is that whatever is the utility functions (subject of optimization) associated with the particular partial plans, it will always be managed (and accessible) to the individual actors. It is not clear whether such information will be available for disclosure. In any cases inter-agent communication is inevitable during such a process. The protocols and interaction methods used here will be similar to those used in the phase $\boxed{2}$.

3.2 Subtask delegation

Subtask delegation is also a classical AI or OR problem that can be solved by various existing methods and techniques for resource allocation. The appropriate choice of the task delegation mechanism depends on availability and the quality of social knowledge in the following way:

– Should social knowledge be available in a good quality to the agent, who is charged with the subtask delegation process, the classical centralized scheduling and resource allocation methods shall be used (e.g. [1], [19]).

– Provided that social knowledge the subtask delegation problem is to be solved by multi-agent classical techniques such contract-net-protocol, various auctioning and combinatorial auctioning techniques (e.g. [18], [10], [3], [17])

Availability and quality of social knowledge does not affect only the choice of the subtask delegation methods but also possibility to optimize the resulting delegation. With little social knowledge potentials for optimization are obviously limited.

Delegating the subtasks may fail. In such circumstances another task decomposition needs to be suggested and processed for subtask allocation. From the conceptual point of view this was meant to be a backtrack situation. However, designers would try to design such a task decomposition mechanism that would comply with feasible subtask allocation. This would be possible if and only if the good quality social knowledge is available.

The claims presented in this paragraph are valid for centralized, collaborative and coordinated DP initiation.

3.3 Conflict detection

Conflict detection is a phase in which each agent analyzes the requests obtained during the phase $\boxed{2}$. The request is rejected, provided the it does not match with the agents capabilities, resource availabilities of collaboration preferences. Conflict may arise due to:

– usage of imprecise social knowledge (caused e.g. by confidentiality reasons, resources overestimation, etc.) used during $\boxed{1}$ and $\boxed{2}$ phases,

– agents deliberate overconstraining its responsibility during the negotiation process taking place within $\boxed{2}$ phase, or

– very frequent changes of agents availability (and thus changes of social knowledge) since the $\boxed{1}$ and $\boxed{2}$ phases (this can be the case of very dynamic, real-time domains).

We claim that social knowledge availability affects implementation of the proposed DP architecture. Let us investigate the two extreme cases:

- there is high quality of *higher level social knowledge* available, providing very precise information about available resources; in such situations splitting the phases $\boxed{1}$ and $\boxed{2}$ is inefficient and both processes will be be implemented by a single algorithm,
- only *minimal social knowledge* is available which results in the phase $\boxed{2}$ being implemented by means of negotiation and agents avoiding conflicting deals; if this true then the phase $\boxed{3}$ will be embedded within the phase $\boxed{2}$.

It looks DP planning architecture will be based on the separate $\boxed{1}$, $\boxed{2}$ and $\boxed{3}$ in all but the two extreme cases listed above. However, if the amount and quality of available social knowledge requires at least small amount of interaction in the phases $\boxed{2}$ it is unlikely that any sperate phase $\boxed{3}$ will be required as the conflicts will be avoided during the phase $\boxed{2}$. On the other hand, if there is no negotiation and interaction required during the phase $\boxed{2}$ there is no reason for splitting the phases $\boxed{1}$ and $\boxed{2}$. Consequently splitting the phases is reasonable only in the case where social knowledge availability is different in various situations and in different teams of agents.

3.4 Individual planning

Individual planning is a classical plan construction or plan selection activity for which existing planning approaches will be used. If a possible conflict has not been detected during the phase $\boxed{3}$, failure of individual planning is less likely (while can happen in nontrivial planning problems).

From the point of view of the individual agents, the phase $\boxed{4}$ results in a set of preliminary commitments for implementation of the tasks assigned during the phase $\boxed{2}$ or $\boxed{3}$ respectively. These commitments will be further refined during the phase $\boxed{5}$.

From the implementation point of view, arbitrary planning and resource allocation algorithms can be used in the phase $\boxed{4}$. If the individual planning does not require any further collaboration with the other agents, the methods used in the phase $\boxed{1}$ and $\boxed{2}$ under the assumption of full social knowledge availability can be used in a similarly way during the phase $\boxed{4}$. If individual planning requires further task decomposition and subtask delegation among the collaborating agents (*nested planning*), the overall distributed planning process is initiated here again.

Nested distributed planning may provide specific reasons for failing individual planning (and thus causing a backtrack in the Durfee's planning sequence).

3.5 Plan merging

Plan merging is a very challenging phase within the Durfee's architecture. However, we argue that in the DP problems there are little practical requirements for obtaining and maintaining a centralized plan within the knowledge structures of the DP initiator. That is why classical works on plan merging are dispensable in the DP context. Instead, we suggest the core of the phase $\boxed{5}$ to be in plan coordination (refereed to as Multi-agent Plan Coordination Problem – MPCP), so that resources are used exclusively used and partial goals shared appropriately.

The phase $\boxed{5}$ in the sense of MPCP is critical in the situations of coordinated DP initiation, where agents have their partial plans for which they need to find complete plans in coordination and collaboration with the other.

3.6 Replanning

Replanning is an obvious outcome from a possible failure of the phase $\boxed{5}$ operation. In the ideal case only the individual plans get replanned, while backtracking to phase $\boxed{1}$, $\boxed{2}$ and $\boxed{3}$ is possible. Replanning may also occur (and is very likely to) during the **plan execution** phase or in the idle times, prior plan execution starts. The key triggers of such replanning are (i) plan miscoordination during $\boxed{5}$ or (ii) deviation from the agents' individual plans. Consequently, one of the key requirements for the DP architecture is a mechanism for **committing** the agents to the individual plan and mechanism supporting coordinated release of the commitment. Also it is expected that planning knowledge availability during the time of planning and replanning may vary or may even become inaccessible.

4 Techniques Supporting Distributed Planning

⊛ [MP]$_4$:part of the literature review: Gerhard please help here, Austin?

The good choice of plan representation is important. We need to decide what degree of flexibility would be required for execution. Shall we allow nested planning with the same community of agents, or will that be treated as a separate planning process (both are non-trivial)?

The choice of the plan representation language shall comply with the requirements for the (i) specifics domain we may come-up with (ii) underlying reasoning/planning processes and (iii) openness to support agent-to-agent negotiation and knowledge sharing. Plan representation is likely to be based on HTN, PGP, GraphPlans or POGP. Plan representation and the choice of an appropriate planner supports mainly the phase $\boxed{4}$.

In the following text we present several hand-picked techniques and approaches that may provide a substantial value to the DP integrated architecture.

4.1 Iterative Query-based Acquaintance Model

IQBM is a special contracting protocol that facilitates efficient task decomposition and subtask allocation processes in the situations where little or no social knowledge is available (latter option from above). IQMB is an iterative process of running the phase $\boxed{1}$ based on rather imprecise social knowledge and than merged phases $\boxed{2}$ and $\boxed{3}$. A possible failure of $\boxed{3}$ provides $\boxed{1}$ with additional social knowledge that is used for a refined $\boxed{1}$ process.

- ⊕ very flexible and computationally efficient approach, works nicely in semi-trusted environment

- ⊖ known weaknesses in competitive environment, tested on very limited datasets

4.2 ECNP/PAP

ECNP – Extended Contract Net Protocol and PAP – provisional Agreement Protocol are specific approach to solving the combinatorial auction problems. As well as IQBM, they also contribute to solving the $\boxed{1}$, $\boxed{2}$ and $\boxed{3}$ DP phases. ECNP extends the protocol with a `provisional accepts` and `provisional rejects` and allows CNP backtracking. Planning here is searching through a dynamically constructed AND/OR graph. PAP also allows `provisional bid` and `withdraw bid`.

- ⊕ It has several applications in the military logistics, it has been connected with trust modeling

- ⊖ does not work in semi-trusted environment

4.3 Multiagent Opportunistic Planning

A very specific technique for collaborative planning and collaborative plan execution that is making the best use of sharing resources and sharing overlapping goals. In the DP architecture the MAOP contributes to phases $\boxed{2}$ $\boxed{4}$ and $\boxed{5}$.

The key idea is that each agent creates plans that also include opportunities for the other agents. If the opportunity goal becomes pending it can be achieved by other agents. Goals may became unachievable due to changes in the environment or were unachieveble from the very start.

They work with POPG as they provide a bigger deal of flexibility for execution. They have tested several different strategies for selecting the additional goals for which an opportunity may arise.

⊕ based on minimal knowledge sharing (they share information about other agents capabilities and assigned goals, which may be even too much in our domain)

⊖ current implementation does not allow for online replanning (in a sense of dropping plans and adopting new plans instead)

4.4 Commitments/Decommitments

There is a broad area specifying formal models of agents commitments as mental structures in their programmes. An inseparable part of each commitment is a specification of the conditions/postconditions under which the agents are allowed to drop their commitments.

There is different use of commitments in the collaborative and competitive environments. While in cooperative settings the commitments postconditions provide mainly notification functionality, in the competitive environment the commitment postcondition provide incentive for an agent to keep its commitment (mainly in the from of penalties). It is believed that the combination of both would be necessary for DP architecture.

This work supports to the phases $\boxed{3}$, $\boxed{4}$ and $\boxed{5}$ of DP architecture.

⊕ well founded theory

⊖ not used for an application yet

4.5 Multiagent Plan Coordination

That is a rather theoretical work that is working with partial order, causal link (POCL) definition of a plan. They provide formal definition of the multi-agent parallel POCL plan, where they introduce *parallel step thread flaw* and *plan merge step flaw*. Based on this they have designed a multi-agent plan coordination algorithm that is working with the space of complete plans of the individual agents. The algorithm is based on branch-and-bounds search.

This work clearly supports to the phase $\boxed{5}$ of DP architecture.

⊕ high relevance, good formal foundation, provides empirical comparison to classical work of Yang

⊖ centralized, not used for an application yet

4.6 Stand-in Agents

A very specific multiagent technique supporting interaction among the agents while inaccessible or off-line (due to intentional logging off from the network or due to inaccessibility caused by properties of e.g. an ad-hoc networking environment). Stand-in agent is a copied agent that either becomes on-line when the owner is off-line or migrates to such a part of the network that retains its connectivity with the other agents.

- Various distributed methods for optimal placement of the stand-in agents have been designed and investigated (such as *forward swarming* and *backward swarming*).

Stand-in agents are expected to support the DP architecture in the phases $\boxed{1}$, $\boxed{2}$, $\boxed{3}$ and $\boxed{5}$.

⊕ implemented and tested in ad-hoc networking environment

⊖ integration with DP architecture may not be straightforward

References

[1] Amotz Bar-Noy, Reuven Bar-Yehuda, Ari Freund, Joseph (Seffi) Naor, and Baruch Schieber. A unified approach to approximating resource allocation and scheduling. *J. ACM*, 48(5):1069–1090, 2001.

[2] Avrim L. Blum and Merrick L. Furst. Fast planning through planning graph analysis. *Artificial Intelligence*, 90:281–300, 1997.

[3] Craig Boutilier, Moisés Goldszmidt, and Bikash Sabata. Sequential auctions for the allocation of resources with complementarities. In *IJCAI '99: Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 527–523. Morgan Kaufmann Publishers Inc., 1999.

[4] Mathijs M. de Weerdt, André Bos, J.F.M. Tonino, and Cees Witteveen. A resource logic for multi-agent plan merging. *Annals of Mathematics and Artificial Intelligence, special issue on Computational Logic in Multi-Agent Systems*, 37(1–2):93–130, January 2003.

[5] Marie E. DesJardins and Michael J. Wolverton. Coordinating a distributed planning system. *AI Magazine*, 20(4):45–53, 1999.

[6] Edmund H. Durfee. Distributed problem solving and planning. In Gerhard Weiß, editor, *A Modern Approach to Distributed Artificial Intelligence*, chapter 3. The MIT Press, San Francisco, CA, 1999.

[7] Eithan Ephrati and Jeffrey S. Rosenschein. A heuristic technique for multiagent planning. *Annals of Mathematics and Artificial Intelligence*, 20(1–4):13–67, 1997.

[8] Kutluhan Erol, James Hendler, and Dana S. Nau. HTN planning: Complexity and expressivity. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, volume 2, pages 1123–1128, Seattle, Washington, USA, 1994. AAAI Press/MIT Press.

[9] N. Griffiths and M. Luck. Cooperative plan selection through trust. In F. J. Garijo and M. Boman, editors, *Multi-Agent System Engineering: Proceedings of the Ninth European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'99)*. Springer-Verlag, 1999.

[10] L. Hunsberger and B. J. Grosz. A combinatorial auction for collaborative planning. In *ICMAS '00: Proceedings of the Fourth International Conference on MultiAgent Systems (ICMAS-2000)*, pages 151–158. IEEE Computer Society, 2000.

[11] Subbarao Kambhampati. A comparative analysis of partial order planning and task reduction planning. *SIGART Bulletin*, 6(1):16–25, 1995.

[12] V. Maříkand M. Pěchouček. Social knowledge in multi-agent systems. In M. Luck, V. Mařík, and O. Štěpánková, editors, *In Conference Proceedings 2004 IEEE International Conference on Systems, Man & Cybernetics*, volume 1, pages 1950–1957. Piscataway: IEEE, 2004.

[13] V. Mařík, M. Pěchouček, and O. Štěpánková. Social knowledge in multi-agent systems. In M. Luck, V. Mařík, and O. Štěpánková, editors, *Multi-Agent Systems and Applications*, LNAI. Springer-Verlag, Heidelberg, 2001.

[14] A.S. Patrick. Building trustworthy software agents. *IEEE Internet Computing*, 6(6):46–53, 2002.

[15] Michal Pěchouček, Vladimír Mařík, and Jaroslav Bárta. Role of acquaintance models in agent's private and semi-knowledge disclosure. *Knowledge-Based Systems*, (19):259–271, 2006.

[16] Martin Rehák, Lukáš Foltýn, Michal Pěchouček, and Petr Benda. Trust model for open ubiquitous agent systems. In *Intelligent Agent Technology, 2005 IEEE/WIC/ACM International Conference*, number PR2416 in IEEE, 2005.

[17] T. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1-2):1–54, 2002.

[18] R. G. Smith. The contract net protocol: High level communication and control in a distributed problem solver. *In IEEE Transactions on Computers*, C-29(12):1104–1113, 1980.

[19] K.P. Sycara, S.F. Roth, N. Sadeh, and M.S. Fox. Resource allocation in distributed factory scheduling. *IEEE Intelligent Systems and Their Applications*, 6(1):29–40, 2 1991.

[20] A. Tate. The helpful environment: Geographically dispersed intelligent agents that collaborate. *Special Issue on The Future of AI, IEEE Intelligent Systems*, 27(3):57–61, May-June.

[21] Håkan L. S. Younes and Reid G. Simmons. VHPOP: Versatile heuristic partial order planner. *Journal of AI Research*, 20:405–430, 2003.