Multiagent Planning with Partially Ordered Temporal Plans

Michael Brenner
Institut für Informatik, Universität Freiburg, 79110 Freiburg, Germany
brenner@informatik.uni-freiburg.de

Abstract

This paper* discusses the specifics of planning in multiagent environments. It presents the formal framework MAPL ("maple") for describing multiagent planning domains. MAPL allows to describe both qualitative and quantitative temporal relations among events, thus subsuming the temporal models of both PDDL 2.1 and POP. Other features are different levels of control over actions, modeling of agents' ignorance of facts, and plan synchronization with communicative actions. For single-agent planning in multiagent domains, we present a novel forward-search algorithm synthesizing MAPL's partially ordered temporal plans. Finally, we present a general distributed algorithm scheme for solving MAPL problems with several coordinating planners. These different contributions are intended as as step towards a simple, yet expressive standard for the description of multiagent planning domains and algorithms. Such a standard could in the future allow cross-evaluation of Multiagent Planning algorithms on standardized benchmarks.

In this paper, we discuss the specific properties of planning in Multiagent Systems (MAS). With the term Multiagent Planning (MAP), we denote any kind of planning being performed in multiagent environments, meaning on the one hand that the planning process itself may be distributed among several *planning* agents, but also that individual plans can (and possibly must) take into account concurrent actions by several *executing* agents.

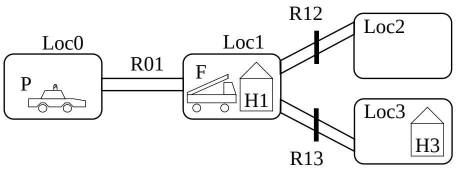

Figure 1: A multiagent planning problem

As a motivating example, Fig. 1 shows a simple MAP problem as appearing in the RoboCupRescue challenge [7]. Two autonomous agents, police force P and fire brigade F, are working in a city devastated by an earthquake. While F's goal is to extinguish all burning houses, it is P's capability and goal to clear the blocked roads. P's position being Loc0 we will assume him being unaware of R12 and R13 being blocked. The agents' actions have specific durations that may be exactly known only at execution time, sometimes because of specific execution parameters of the agents, sometimes because of intrinsic unpredictability of the environment: while, for example, moving through the town may take between 2 and 4 minutes and depend only on the map distance and speed of the agent, extinguishing a fire may take 1 to 4 hours depending on conditions unknown to the agents.

This trivial example shows several general features of MAP problems and plans: (1) Agents may be *unaware* of parts of the world state (P does not know whether R13 is blocked). (2) *Concurrent acting* is central to MAP (P can move to Loc1 and start clearing R13 while F is extinguishing H1, although both agents using the same road at the same time may be prohibited to avoid collisions). Modeling concurrency necessitates (2a) a description of which events may occur concurrently and which not, (2b) *metric time* for realistic descriptions of action durations and their relations, but (2c) synchronizing on actions of unknown (at least to some agent) duration demands *qualitative* use of time (e.g. "after P has cleared R13"). A specific usage of qualitative time is (3) synchronization on *communicative acts*, as in "F moves to Loc3 after P has informed him that R13 is clear".

In their plans, agents must take other agents actions into account: F may "exploit" P's clearing of R13 in his own plan but must also assure that he does not try to use a road that is also used by P at the same time. Especially, (4) cannot *control* occurence or duration of other agents' actions.

To address the representation problems (1)–(4) we introduce the Multiagent Planning Language MAPL ("maple"). Instead of propositional state representations MAPL allows non-boolean state variables (cf. also [5]). To model feature (1) each state variable may have the special value *unknown*, thereby avoiding representation of belief states as sets of possible states. A number of other advantages comes with the introduction of state variables; especially, for feature (2a), an intuitive definition of mutual exclusivity (i.e. the impossibility to execute some actions concurrently, cf. [1]) can be given that describes mutexes as *read-write locks* on state variables. According to this perspective, distributed planning can then be seen as detection or, even better, prevention of possible read-write locks *before* execution.

MAPL's temporal model allows to combine (2b) quantitative and (2c) qualitative temporal information in plans, thus subsuming both the purely metric temporal model of PDDL 2.1 [4] and the purely qualitative model of Partial Order Planning [8]. At its core MAPL represents a multiagent plan as a Simple Temporal Network [3] in which each durative action is modeled by start and end events, possibly extended by invariant conditions. In the STN, both action durations and qualitative ordering relations are treated as constraints represented by closed, semi-open or open intervals. In so doing, not only can imprecisely known action durations be represented as intervals of the form $[\delta_1, \delta_2]$, but qualitative constraints like "after" can be described by (semi-)open intervals

*Definitions of the formal semantics of our Multiagent Planning Language as well as the algorithms for single and multi-agent planning are given in the long version of this paper [2].

like $(0, \infty)$. Fig. 2 shows F's plan for the problem of Fig. 1, the interval $(0, \infty)$ being represented as $<$.

Another new concept shown in Fig. 2 is (3) the use of speech acts as reference events for plan synchronization. On the one hand, this allows agents to refer to facts (especially those achieved by others) the change of which they do not influence or witness themselves. On the other hand, as explained later, speech acts allow agents to reveal only the minimum of information about their plans needed for coordination.

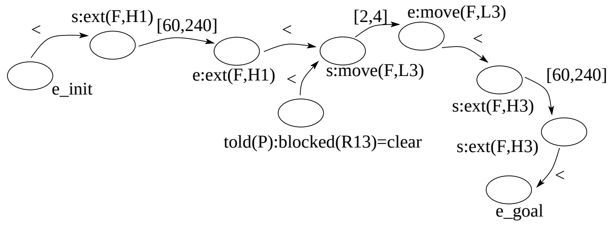

Figure 2: F's plan (including a communicative action by P)

A plan is only fully specified with (4) a *control function* describing which of the agents (or the environment) controls the occurence of each event. With this function we can describe, e.g., that a specific agent is allowed to add and remove an action from his plan (control of the start event), but has no influence on it its duration (end event controlled by the environment). During planning, having control of an event or not fundamentally changes its possible use and evaluation. For example, being able or unable to control the duration of an action will lead the planner to a fundamentally different heuristic evaluation of its use.

For a plan to be executable, it must be both temporally and logically consistent. The former criterion is reducible to consistency of the underlying STN. The latter, logical consistency, can be defined similarly to POP as the plan having no open conditions and no unsafe links, with the additional criterion that the plan must ensure that no mutex events may occur concurrently. For a plan to solve a certain agent's problem it must achieve his goals and also be consistent with the control function, i.e. only constraints involving events controlled by the respective agent must have been tightened by the planner.

How is planning in MAS carried out? It is obvious that the easiest way is to find a plan alone: assumed that F knows about P's capabilities, F can find a plan that solves his problems. Even if F does not know about P's concrete actions, this plan will provide clues about where help is needed and thus triggers cooperation. We see that (5) the capability for single-agent synthesis of multiagent plans is a basic requirement for MAP. We have developed a plan-space forward-search algorithm that can be used with any standard forward branching scheme and arbitrary plan metrics. We present two such metrics, the well-known makespan and the new minMaxMakespan, the latter of which extends the former by assigning maximal possible duration to uncontrolled durative actions. We also describe how heuristic forward planning in the style of FF [6] can be extended to find MAPL plans. The current simple algorithm is sound, but not complete, i.e. there is a set of clearly distinguished MAPL problems it cannot solve yet. We are working on a sound, yet more complex version of the algorithm.

When several agents are planning and acting individually in a common environment, they will probably run into one of the following problems: (6) They won't be able to find individual plans solving their problems or (7) the plans found will conflict at execution time. MAP Literature has mostly treated only problem (7), implicitly assuming that plans can be found and that therefore separating planning from coordination is possible. In our opinion, coordination *during* the planning phase is indispensable in the case of problem (6) and advantageous for problem (7). We have therefore developed a general distributed planning algorithm that uses single-agent planning to synthesize partial plans and to trigger cooperation and coordination efforts as early as possible.

A key concept is the use of a *responsibility function* that assigns to each state variable an agent managing and controlling its changes over time. This agent will detect read-write conflicts in the agents' plans. i.e. possible execution conflicts, but will also provide information when another agent cannot achieve a (sub)goal involving that variable. In the basic form of the algorithm, the responsibility is static, but similar to approaches in Distributed CSP research [9] we will relax this assumption in future work. The idea of the algorithm is simple: in a reachability analysis the planning agent detects goals involving state variables he does not know about, cannot manipulate or could if only some earlier condition were satisfied. He contacts the responsible agents to receive more information or delegate a subgoal concerning the variable. The responsible agent answers the question or adopts a temporary goal to help the asking agent.

All contributions of this paper aim at clarifyfing the specific representational and algorithmic needs of MAP research. We hope that our representation will allow to conveniently describe diverse MAP domains for which researchers can propose and cross-evaluate algorithmic approaches just as diverse, thus promoting the field of Multiagent Planning.

References

[1] A. Blum and M. Furst. Fast planning through planning graph analysis. *AIJ*, 90(1-2), 1997.

[2] M. Brenner. Multiagent planning with partially ordered temporal plans. Technical report, Institut f ̈ur Informatik, Univerisit ̈at Freiburg, Germany, 2003.

[3] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *AIJ*, 49, 1991.

[4] M. Fox and D. Long. *PDDL 2.1: an Extension to PDDL for Expressing Temporal Planning Domains*, 2002.

[5] H. Geffner. Functional STRIPS: a more flexible language for planning and problem solving. In *Logic-Based Artificial Intelligence*. Kluwer, 2000.

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

[7] H. Kitano et al. RoboCupRescue: Search and rescue in large-scale disasters as a domain for autonomous agents research. In *IEEE Intl. Conf. on Systems, Man and Cybernetics*, 1999.

[8] Daniel Weld. An introduction to least commitment planning. *AI Magazine*, 15(4), 1994.

[9] M. Yokoo and K. Hirayama. Algorithms for distributed constraint satisfaction: a review. *Autonomous Agents and Multi-Agent Systems*, 3(2), 2000.