Multi-Agent Assumption-Based Planning

Damien Pellier, Humbert Fiorino
Laboratoire Leibniz, MAGMA
46, avenue Félix Viallet, 38000 Grenoble, France
Damien.Pellier@imag.fr, Humbert.Fiorino@imag.fr

Abstract

The purpose of this poster is to introduce a dialectical theory for plan synthesis based on a multi-agent approach. This approach is a promising way to devise systems able to take into account partial knowledge and heterogeneous skills of agents. We propose to consider the planning problem as a defeasible reasoning where agents exchange proposals and counter-proposals and are able to conjecture *i.e.,* formulate plan steps based on hypothetical states of the world.

1 Introduction

The problem of plan synthesis achieved by autonomous agents in order to solve complex and collaborative tasks is still an open challenge. Increasingly new application areas can benefit from this research domain. From our point of view, multi-agent planning can be likened to the process used in automatic theorem proving. In a sense, a plan can be considered to be a particular proof based on specific rules, called actions. In this poster, we draw our inspiration from the proof theory described by [Lakatos, 1976]. The plan synthesis problem is viewed as a dialectical and collaborative goal directed reasoning about actions. Each agent can refine, refute or repair the ongoing team plan. If the repairing of a previously refuted plan succeeds, it becomes more robust but it can still be refuted later. If the reparation of the refuted plan fails, agents leave this part of the reasoning and explore another possibility: finally "bad" sub-plans are ruled out because no agent is able to push the investigation process further. As in an argumentation with opponents and proponents, the current plan is considered as an acceptable solution when the proposal/counter-proposal cycles end and no more objection remains.

The originality of this approach relies on the agent capabilities to elaborate plans under partial knowledge and/or to produce plans that partially contradict its knowledge. In other words, in order to reach a goal, such an agent is able to provide a plan *which could be executed if certain conditions were met*. Unlike "classical" planners, the planning process does not fail if some conditions are not asserted in the knowledge base, but rather proposes an Assumption-Based Plan or *conjecture*. Obviously, this conjecture must be *reasonable*: the goal cannot be considered "achieved" and the assumptions must be as few as possible because they become new goals for the other agents. For instance, suppose that a door is locked: if the agent seeks to get into the room behind the door and the key is not in the lock, the planning procedure fails even though the agent is able to fulfill 100% of its objectives behind the door. Another possibility is to suppose for the moment that the key is available and then plan how to open the door etc. whereas finding the key might become a new goal to be delegated. To that end, we designed a planner that relaxes some restrictions regarding the applicability of planning operators.

2 Assumption-Based Planning Model

In order to produce conjectures, we propose an assumption-based planning model based on a domain independent planning mecanism, HTN. In HTN planning model, the objective is not to achieve a set of goals but to perform some sets of *tasks*. The agent's input includes a set of operators and also a set of *methods*, each of which is a prescription on how to decompose some tasks into some sets of subtasks. The agent proceeds by decomposing *non-primitive tasks* recursively into smaller and smaller subtasks, until *primitive tasks*, that can be performed directly by planning operators, are reached.

Assumption-Based Planning is defined as $\langle \mathcal{G}, \mathcal{E}, \mathcal{A} \rangle$: $\mathcal{G}$, is a goal description (*i.e.,* a set of world states), $\mathcal{E}$ is a partial description of the world (*i.e.,* the agent's beliefs) and $\mathcal{A}$ is a description of the actions that an agent can execute. $\mathcal{E}$ and $\mathcal{G}$ are described in propositional logic.

A *primitive operator* α is a tuple $\langle name_\alpha, Pre_\alpha, Del_\alpha, Add_\alpha \rangle$ where $name_\alpha$ is the name of the primitive action, Pre_α is the preconditions set needed to execute α, Del_α and Add_α define respectively the set of effects to delete and to add to the agent's beliefs.

A *compound operator* α is described by a method $\langle n_\alpha, Pre_\alpha, Act_\alpha \rangle$ where n_α is the name of α, Pre_α is the preconditions set needed to apply α, Act_α defines a list of actions to execute (*i.e.,* the method body).

A *planning problem* is defined by a tuple $\langle \mathcal{E}, \mathcal{O}, \mathcal{G} \rangle$: $\mathcal{E}$ defines the agent's beliefs; $\mathcal{O} = \{\langle Pre_\alpha, Del_\alpha, Add_\alpha \rangle \mid \alpha \in \mathcal{A}\}$ defines the description of the operators that an agent can execute; $\mathcal{G}$ defines the goal of an agent.

A *conjecture* χ is defined as an ordered list of couples $\chi = (\langle \mathcal{H}_{\alpha_1}, \alpha_1 \rangle, \ldots, \langle \mathcal{H}_{\alpha_n}, \alpha_n \rangle)$ where α_i is an action in $\mathcal{A}$; $\mathcal{H}_{\alpha_i}$ describes the assumptions that must hold before executing α_i. If $\mathcal{H}_{\alpha_i}$ is an empty set, no assumption is needed to apply α_i.

Given a planning problem $\langle \mathcal{E}, \mathcal{O}, \mathcal{G} \rangle$, a conjecture χ is equivalent to an ordered list of $n+1$ world states $\chi = \mathcal{E}_0, \mathcal{E}_1, \ldots, \mathcal{E}_n$ with $\mathcal{E}_0 = \mathcal{E} \cup \mathcal{H}_{\alpha_1}$ and $\mathcal{E}_i = (\{\mathcal{E}_{i-1} \cup \mathcal{H}_{\alpha_{i-1}}\} - Del_{\alpha_i}) \cup Add_{\alpha_i}$ $(1 \leq i \leq n)$. The set of solution conjectures $\mathcal{C}(\mathcal{E}, \mathcal{O}, \mathcal{G})$ of a planning problem can be recursively defined:

- If $\mathcal{G}$ is an empty set, the empty conjecture is returned.
- Otherwise, let α be the first task or goal of $\mathcal{G}$, and $\mathcal{R}$ be the remaining goals: i) If α is a primitive action and there is a conjecture χ_1 to reach α then $\mathcal{C}(\mathcal{E}, \mathcal{O}, \mathcal{G}) = \{append(\chi_1, \chi_2) \mid \chi_2 \in \mathcal{C}(\mathcal{E}, \mathcal{O}, \mathcal{R})\}$; ii) If α is a primitive action and there is no conjecture χ to reach α then $\mathcal{C}(\mathcal{E}, \mathcal{O}, \mathcal{G}) = \emptyset$; and finally, iii) If α is a composed action then $\mathcal{C}(\mathcal{E}, \mathcal{O}, \mathcal{G}) = \mathcal{C}(\mathcal{E}, \mathcal{O}, append(Act_\alpha, \mathcal{R}))$ where Act_α defines the actions list to be executed in order to realize α.

To reach its goal, an agent must check all assumptions made by the conjecture. It can rely on its teammates competences to make those assumptions become true. In other words, assumptions made by one agent become additional goals to be recursively satisfied. The assumption computation distinguishes two kinds of assumptions: i) *Hypothesis* represents literals that do not belong to the current agent's beliefs. This means that expressions missing from the beliefs are not considered as false but rather as unknown; ii) *Fact negation* represents literals that are negations of facts in the current agent's beliefs: this fact is withdrawn and replaced by its negation.

3 Dialectical Controller

In a multi-agent context, the plan synthesis relies on dialectical exchanges between agents as expected in a debate. This is convenient for dialogue where participants have no predetermined role (*i.e.,* proponent, opponent). Agents interact collaboratively in the dialogue so as to construct a plan without assumption fulfilling the assigned goals. Agents have a dialogue based on speech acts. At the informational layer, speech acts can contain conjectures or propositions needed to elaborate the global shared plan. The conjectures and the refutations suggested by the agents during the dialogue are stored in a structure called *proof board*. Each agent has its own proof board initialized at the dialogue beginning. The proof board defines the public part of the agents' interactions and records the agents' reasoning. Each time an agent uses a speech act to refine, refute or repair a conjecture, the proof board is updated. Through agents' dialogue, only one specific part of the shared plan is considered. We call this particular conjecture, the *current conjecture*. This conjecture defines the dialogue context. In order to structure the dialectical plan synthesis, we propose to define a set of dialogue conventions described according to the states of the proof board. These conventions are organized in two layers:

- an *informational layer* that defines the rules to exchange refinements, refutations and repairings about the current conjecture. Each new conjecture suggested by an agent produces new goals to achieve for the other agents.
- a *contextualization layer* in which agents decide to start, when a set of goals is assigned to the system, or stop interacting when they believe that a solution was found or is not reachable. Finally, agents can decide to change the dialogue context by modifying the current conjecture if it has been refuted or none of the agents can refine its assumptions.

The proof board of each agent can be viewed as a proof tree. Each node defines a conjecture suggested by an agent. The edges of the tree express refinement relations. Each child node defines a sub-conjecture proving an assumption of the conjecture contained in the father node. In order to test if a solution plan *i.e.,* a plan without assumption is contained in the proof board, agents try to extract recursively a sub-tree of the proof board from the root node. For each assumption in the root node, a sub-tree which nodes are not refuted and do not contain assumptions must be extracted. This test is recursively triggered on the alternative refinements proposed for each assumption. The performatives list used are shown in Tab. 3. We consider two performatives levels. The performatives used to refine, refute or repair conjectures that have an informational content and the performatives used to modify the dialogue context. Each time a performative is received or sent, the proof board is modified

Levels	**Performatives**
Informational	refine, refute, repair
Contextualization	prop.enter, prop.leave, prop.change, ack.enter, ack.leave, ack.change, rej.enter, rej.leave, rej.change

Table 1: Performatives list ordered by level.

4 Conclusion

The advantage of the dialectical plan synthesis is to merge in the collaborative plan generation, the composition and the coordination steps. It also includes the notion of uncertainty in the agents' reasoning and allows agents to make conjectures and to compose their heterogeneous competences. Moreover, we apply conjecture/refutation to structure the multi-agent reasoning as a collaborative investigation process. However, former works on synchronization, coordination and conflict resolution are integrated through the notions of refutation/repairing. From our point of view, this approach is suitable for applications in which agents share a common goal and in which the splitting of the planning and the coordination steps (when agents have independent goals, they locally generate plans and then solve their conflicts) becomes difficult due to the agents strong interdependence.

References

[Lakatos, 1976] I. Lakatos. *Proofs and Refutations: The Logic of Mathematical Discovery*. Cambridge University Press, Cambridge, England, 1976.