



HAL
open science

An Implemented System for Cognitive Planning

Jorge Luis Fernandez Davila, Dominique Longin, Emiliano Lorini, Frédéric Maris

► **To cite this version:**

Jorge Luis Fernandez Davila, Dominique Longin, Emiliano Lorini, Frédéric Maris. An Implemented System for Cognitive Planning. Journées Francophones Planification, Décision et Apprentissage (JF-PDA 2022), Plate-Forme Intelligence Artificielle, Jun 2022, Saint-Etienne, France. pp.1-4. hal-03763068

HAL Id: hal-03763068

<https://ut3-toulouseinp.hal.science/hal-03763068v1>

Submitted on 29 Aug 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

An Implemented System for Cognitive Planning

Jorge Luis Fernandez Davila¹, Dominique Longin², Emiliano Lorini², and Frédéric Maris¹

¹IRIT, Toulouse University, France

²IRIT, CNRS, Toulouse University, France

{Jorge.Fernandez, Dominique.Longin, Emiliano.Lorini, Frederic.Maris}@irit.fr

Abstract

We present a system that implements a framework for cognitive planning¹. The system allows us to represent and reason about the beliefs, desires and intentions of other agents using an NP-fragment of a multiagent epistemic logic. The system has three components: the belief revision, the planner and the translator modules. They work in an integrated way to firstly capture new information about the world, secondly to perform the planning process extracting operators from a set of actions that will lead to achieving a goal and, finally, to verify satisfiability of the formulas generated at each step of the process. We illustrate how our system can be used to implement an persuasive artificial agent interacting with a human user.

1 Introduction

Automated planning is at the center of AI research with a variety of applications ranging from control traffic and robotics to logistics and services. Epistemic planning extends automated planning incorporating notions of knowledge and beliefs [Bolander and Andersen, 2011; Löwe *et al.*, 2011; Kominis and Geffner, 2015; Muise *et al.*, 2015; Cooper *et al.*, 2020]. Cognitive planning is a generalization of epistemic planning, where the goal to be achieved is not only a belief state but a cognitive state of a target including not only beliefs but also intentions. Moreover, we are particularly interested in persuasive goals of the planning agent, aimed at influencing another agent’s beliefs and intentions.

The increasing number of applications in social robotics, social networks, virtual assistants together with sentiment analysis techniques allow us to collect data related to humans’ beliefs and intentions. In [Akimoto, 2019] a framework for modeling mental attitudes of an agent, based on her narratives, is proposed. In addition, cognitive models can be used to predict agents’ decision-making by taking psychological factors like motivation and emotions into account [Prezenski *et al.*, 2017]. Nonetheless, few approaches exist which leverage this information about humans’ cognitive states for changing their attitudes and behaviors through persuasion.

Our work aims to fill this gap by introducing a system based on a simple framework detailed in [Fernandez *et al.*, 2021] in which we can represent an agent’s cognitive state in a compact way, reason about it and planning a sequence of speech acts aimed at changing it. Our approach is based on an epistemic logic introduced in [Lorini, 2018; Lorini, 2020], which allows us to represent an agent’s explicit beliefs, as the information in the agent’s belief base, and the agent’s implicit beliefs, as the information which is deducible from the agent’s belief base. Given that the satisfiability problem for the full logic is PSPACE-hard, we focus on an NP-fragment that makes the logic suitable for implementing real-world applications.

The core components of the system are the belief revision, the planner and the translator modules. The formulas representing the rules and constraints of a specific problem are loaded into the system. We encode these rules using the NP-fragment presented in [Fernandez *et al.*, 2021]. The system takes this information as the initial state and some actions — which are of type speech act — to build a plan that leads to the goal. An important feature is that actions have preconditions that impose constraints on their execution order. We illustrate the implementation of our system in a human-machine interaction (HMI) scenario in which an artificial agent has to persuade a human agent to practice a sport based on her preferences.

2 A Language for Explicit and Implicit Belief

This section describes the basics of the Logic of Doxastic Attitudes (LDA) introduced in [Lorini, 2018; Lorini, 2020]. It is a multiagent epistemic logic which supports reasoning about explicit and implicit beliefs. Assume a countably infinite set of atomic propositions Atm and a finite set of agents $Agt = \{1, \dots, n\}$. We define the language in two steps.

We first define the language $\mathcal{L}_0(Atm, Agt)$ by the following grammar in Backus-Naur Form (BNF):

$$\alpha ::= p \mid \neg\alpha \mid \alpha_1 \wedge \alpha_2 \mid \alpha_1 \vee \alpha_2 \mid \Delta_i\alpha,$$

where p ranges over Atm and i ranges over Agt . $\mathcal{L}_0(Atm, Agt)$ is the language for representing agents’ explicit beliefs. The formula $\Delta_i\alpha$ is read “ i explicitly believes that α ”. The language $\mathcal{L}(Atm, Agt)$ extends the language $\mathcal{L}_0(Atm, Agt)$ by modal operators of implicit belief and is defined by the following grammar:

¹<https://www.irit.fr/CoPains/software/>

$$\mathcal{L}_{\text{Frag}}^+ \xrightarrow{\text{red}} \mathcal{L}_{\text{Frag}} \xrightarrow{\text{nnf}} \mathcal{L}_{\text{Frag}}^{\text{NNF}} \xrightarrow{\text{tr}_1} \mathcal{L}_{\text{Mod}} \xrightarrow{\text{tr}_2} \mathcal{L}_{\text{Prop}}$$

Figure 1: Summary of reduction process

$$\varphi ::= \alpha \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \Box_i\varphi \mid \Diamond_i\varphi,$$

where α ranges over $\mathcal{L}_0(\text{Atm}, \text{Agt})$ and i ranges over Agt . For notational convenience we write \mathcal{L}_0 instead of $\mathcal{L}_0(\text{Atm}, \text{Agt})$ and \mathcal{L} instead of $\mathcal{L}(\text{Atm}, \text{Agt})$, when the context is unambiguous. The formula $\Box_i\varphi$ is read “ i implicitly believes that φ ” and $\Diamond_i\varphi$ is read “ φ is compatible (or consistent) with i ’s explicit beliefs”. The other Boolean constructions \top , \perp , \rightarrow and \leftrightarrow are defined in the standard way.

The language is interpreted with respect to a formal semantics using belief bases whose details are given in [Lorini, 2018; Lorini, 2020]. Checking satisfiability of \mathcal{L} formulas relative to this semantics is a PSPACE-hard problem. For that reason, in [Fernandez *et al.*, 2021], we looked for an interesting NP-fragment of \mathcal{L} that we called $\mathcal{L}_{\text{Frag}}$:

$$\varphi ::= \alpha \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \Box_m\alpha \mid \Diamond_m\alpha,$$

where α ranges over \mathcal{L}_0 and m is a special agent in Agt called the ‘machine’. In $\mathcal{L}_{\text{Frag}}$, all agents have explicit beliefs but only agent m has implicit beliefs, and moreover the latter are restricted to \mathcal{L}_0 formulas of type α . Agent m is the artificial planning agent. In order to represent agents’ belief dynamics, language $\mathcal{L}_{\text{Frag}}$ is extended by belief expansion operators. Such an extension will allow us to represent the actions of the planning agent in the cognitive planning problem. Specifically, we introduce the following language $\mathcal{L}_{\text{Frag}}^+$:

$$\varphi ::= \alpha \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \Box_m\alpha \mid \Diamond_m\alpha \mid [+_i\alpha]\varphi,$$

where α ranges over \mathcal{L}_0 and i ranges over Agt . The formula $[+_i\alpha]\varphi$ is read “ φ holds after agent i has privately expanded her belief base with α ”. Event of type $+_i\alpha$ are generically called informative actions.

3 Cognitive Planning

The planning problem in the context of the logic $\mathcal{L}_{\text{Frag}}^+$ is to find a sequence of informative actions for agent m of type $+_m\alpha$ which guarantees that agent m will knowingly achieve its goal. Let $\text{Act}_m = \{+_m\alpha : \alpha \in \mathcal{L}_0\}$ be agent m ’s set of informative actions and let the elements of Act_m be noted $\epsilon, \epsilon', \dots$. Agent m ’s informative actions have executability preconditions that are specified by the following function: $\mathcal{P} : \text{Act}_m \rightarrow \mathcal{L}_{\text{Frag}}$. So, we can define the following operator of successful occurrence of an informative action:

$$\langle\langle\epsilon\rangle\rangle\varphi \stackrel{\text{def}}{=} \mathcal{P}(\epsilon) \wedge [\epsilon]\varphi$$

with $\epsilon \in \text{Act}_m$. The formula $\langle\langle\epsilon\rangle\rangle\varphi$ has to be read “agent m ’s informative action ϵ can take place and φ holds after its occurrence”.

Informative actions of type ‘speech act’ are of interest here. In particular, we consider speech acts of type ‘to inform’, where m is assumed to be the speaker and $j \in \text{Agt}$ such that $j \neq m$ is assumed to be the hearer. We define the speech act “agent m informs agent j that α ” as follows:

$$\text{inform}(m, j, \alpha) \stackrel{\text{def}}{=} +_m \Delta_j \alpha.$$

In [Fernandez *et al.*, 2021] the planning problem is defined as a tuple $\langle \Sigma, \text{Op}, \alpha_G \rangle$ where:

- $\Sigma \subset \mathcal{L}_0$ is a finite set of agent m ’s available information,
- $\text{Op} \subset \text{Act}_m$ is a finite set of agent m ’s operators,
- $\alpha_G \in \mathcal{L}_0$ is agent m ’s goal.

The planning problem has a solution if the formula $\neg((\bigwedge_{\alpha \in \Sigma} \Box_m \alpha) \rightarrow \langle\langle\epsilon_1\rangle\rangle \dots \langle\langle\epsilon_k\rangle\rangle \Box_m \alpha_G)$ is unsatisfiable. The translator module converts the $\mathcal{L}_{\text{Frag}}^+$ -formula into its equivalent formula in propositional logic following the sequence of reductions detailed in Figure 1. In a second step, the translator interfaces with the SAT encoding tool TouIST [Fernandez *et al.*, 2020] to verify the validity of the propositional formula. TouIST will encode the formula in its DIMACS CNF format and send it to the MiniSAT solver for checking satisfiability.

4 Example and Implementation: Artificial Assistant

In order to probe the potential of our implemented system for cognitive planning we test it in a HMI scenario detailed in [Fernandez *et al.*, 2021]. In this scenario agent m is the artificial assistant of the human agent h . Agent h has to choose a sport to practice since her doctor recommended her to do a regular physical activity to be in good health. Agent m ’s aim is to help agent h to make the right choice, given her actual beliefs and desires. Agent m has to provide information about the possible options (Opt) the user can choose and their properties (Var). For each pair (Opt, Var) we assign a valuation Val . In this example, we suppose that Opt is composed of the following eight elements: swimming (sw), running (ru), horse riding (hr), tennis (te), soccer (so), yoga (yo), diving (di) and squash (sq). Moreover, there are exactly six variables in Var which are used to classify the available options: environment (env), location (loc), sociality (soc), cost (cost), dangerousness (dan) and intensity (intens). The variable assignments are showed in Table 1:

Opt	Var					
	env	loc	soc	cost	dan	intens
sw	<i>water</i>	<i>mixed</i>	<i>single</i>	<i>med</i>	<i>low</i>	<i>high</i>
ru	<i>land</i>	<i>outdoor</i>	<i>single</i>	<i>low</i>	<i>med</i>	<i>high</i>
hr	<i>land</i>	<i>outdoor</i>	<i>single</i>	<i>high</i>	<i>high</i>	<i>low</i>
te	<i>land</i>	<i>mixed</i>	<i>mixed</i>	<i>high</i>	<i>med</i>	<i>med</i>
so	<i>land</i>	<i>mixed</i>	<i>team</i>	<i>med</i>	<i>med</i>	<i>med</i>
yo	<i>land</i>	<i>mixed</i>	<i>single</i>	<i>med</i>	<i>low</i>	<i>low</i>
di	<i>water</i>	<i>mixed</i>	<i>single</i>	<i>high</i>	<i>high</i>	<i>low</i>
sq	<i>land</i>	<i>indoor</i>	<i>mixed</i>	<i>high</i>	<i>med</i>	<i>med</i>

Table 1: Variable assignments. For every option $o \in \text{Opt}$ and variable $x \in \text{Var}$, we denote by $v_{o,x}$ the corresponding entry in the table. For instance, we have $v_{\text{sw}, \text{env}} = \text{water}$.

Formulas representing the rules and constraints are loaded as part of agent m ’s belief base. For example, the implementation of the formula representing the fact that agent h explicitly believes that a sport cannot have two different values for

a given property is formalized as follows:

$$\bigwedge_{\substack{o \in Opt \\ x \in Var \\ v_1, v_2 \in Val_x: v_1 \neq v_2}} \left(\Delta_h \text{val}(o, x \mapsto v_1) \rightarrow \Delta_h \neg \text{val}(o, x \mapsto v_2) \right)$$

```
bigand
$o, $x, $v1, $v2 in $Opt, $Var, $Val($x), $Val($x)
when $v1 != $v2:
{h}val($o, ass($x, $v1))=>{h}not val($o, ass($x, $v2))
end
```

The set of actions are generated from table 1:

```
inform(m, h, val_so_ass_env_land)
```

The previous informative action is interpreted as the speech act used by agent m to inform agent h that the valuation of the property:environment for the option:soccer is land. In order to help agent h to select an activity, agent m also needs information about h 's set of actual desires. Information about agent h 's desires is gathered by agent m during its interaction with h . The interaction interface between agents h and m is showed in figure 2. The belief revision module is called after each agent h 's feedback and it restores consistency of the agent m 's belief base, in case the incoming information is inconsistent with agent m 's pre-existent beliefs.

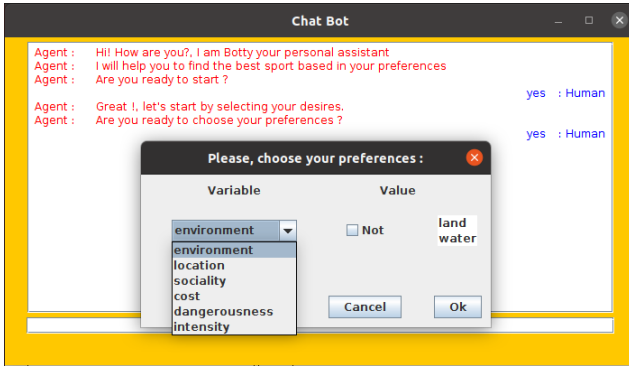


Figure 2: Collecting agent h 's preferences

In the example, agent h would like to practice a land activity, with medium intensity, which is not exclusively indoor, and which can be practiced both in single and team mode, if its cost is high. The next rule for precondition states that agent h must be informed by agent m about the dangerousness level of a sport, before presenting other properties for an option. For $a \notin Assign_{dan}$:

$$\mathcal{P}(\text{inform}(m, h, \text{val}(o, a))) = \Box_m (\text{val}(o, a) \wedge \bigwedge_{v \in Val_{dan}} (\text{val}(o, \text{dan} \mapsto v) \rightarrow \Delta_h \text{val}(o, \text{dan} \mapsto v)))$$

In the next lines we illustrate how the precondition is assigned by the planner module together with its $+_m\alpha$ operator in order to specify the successful occurrence of an informative action:

```
[m]((val_te_ass_intens_med) and
(val_te_ass_danger_med =>
{h}val_te_ass_danger_med)) and
plus({h}val_te_ass_intens_med, ...
```

The planner reads the Initial State, the Actions and the Goal files. The planning module generates plans with the elements contained in the Action file. It starts with plans of length 1, and enters in a loop. At each interaction the planner asks the SAT solver to verify whether the plan allows to achieve the Goal. If no plan of length k is found, the program will increase the counter in one and look for a plan of length $k+1$.

```
plus({h}(val_te_ass_danger_med)
plus({h}(val_te_ass_intens_med)
plus({h}(val_te_ass_soc_mixed)
plus({h}(val_te_ass_loc_mixed)
plus({h}(val_te_ass_env_land)
plus({h}(ideal_h_te)
```

The order of speech acts is determined by the preconditions. Specifically, the planner informs firstly about the dangerousness level of the sport. Secondly, it provides explanation of why the user's desires are satisfied. Finally, it indicates the ideal sport for the user, in this case tennis.

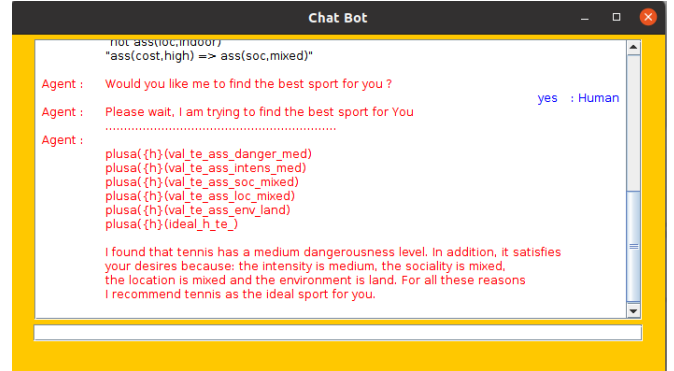


Figure 3: Plan showed by the chatbot to the human

The chatbot writes both the formal actions sequence and its translation in natural language.

5 Conclusion

The implementation demonstrates that the NP-complete episodic logic presented in [Fernandez *et al.*, 2021] and the cognitive planning problem formulated in this logic are suitable for real-world applications in the domain of human-machine interaction. In future work, we plan to extend the implemented system by speech acts of type question to capture both sides of interaction, from agent m to agent h (handled by the actual implementation) and from agent h to agent m . We also plan to combine our implementation of cognitive planning with machine learning and data mining techniques, as presented in [Krzywicki *et al.*, 2016], which are aimed at extracting information about the human user from real data.

References

- [Akimoto, 2019] Taisuke Akimoto. Narrative structure in the mind: Translating genette’s narrative discourse theory into a cognitive system. *Cognitive Systems Research*, 58:342–350, 2019.
- [Bolander and Andersen, 2011] T. Bolander and M. B. Andersen. Epistemic planning for single- and multi-agent systems. *Journal of Applied Non-Classical Logics*, 21(1):9–34, 2011.
- [Cooper *et al.*, 2020] Martin C. Cooper, Andreas Herzig, Faustine Maffre, Frédéric Maris, Elise Perrotin, and Pierre Régnier. A lightweight epistemic logic and its application to planning. *Artificial Intelligence*, page 103437, 2020.
- [Fernandez *et al.*, 2020] Jorge Fernandez, Olivier Gasquet, Andreas Herzig, Dominique Longin, Emiliano Lorini, Frédéric Maris, and Pierre Régnier. Touist: a friendly language for propositional logic and more. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 5240–5242. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Demos.
- [Fernandez *et al.*, 2021] Jorge Fernandez, Dominique Longin, Emiliano Lorini, and Frédéric Maris. A simple framework for cognitive planning. In *Proceedings of the Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*. AAAI Press, 2021. To appear.
- [Kominis and Geffner, 2015] F. Kominis and H. Geffner. Beliefs in multiagent planning: from one agent to many. In Ronen I. Brafman, Carmel Domshlak, Patrik Haslum, and Shlomo Zilberstein, editors, *Proceedings of the 25th International Conference on Automated Planning and Scheduling (ICAPS 2015)*, pages 147–155. AAAI Press, 2015.
- [Krzywicki *et al.*, 2016] Alfred Krzywicki, Wayne Wobcke, Michael Bain, John Calvo Martinez, and Paul Compton. Data mining for building knowledge bases: Techniques, architectures and applications. *The Knowledge Engineering Review*, -1:1–27, 03 2016.
- [Lorini, 2018] E. Lorini. In praise of belief bases: Doing epistemic logic without possible worlds. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, pages 1915–1922. AAAI Press, 2018.
- [Lorini, 2020] E. Lorini. Rethinking epistemic logic with belief bases. *Artificial Intelligence*, 282, 2020.
- [Löwe *et al.*, 2011] B. Löwe, E. Pacuit, and A. Witzel. DEL planning and some tractable cases. In *Proceedings of the 3rd International International Workshop on Logic, Rationality and Interaction (LORI 2011)*, pages 179–192. Springer Berlin Heidelberg, 2011.
- [Muise *et al.*, 2015] C. Muise, V. Belle, P. Felli, S. A. McIlraith, T. Miller, A. R. Pearce, and L. Sonenberg. Planning over multi-agent epistemic states: A classical planning approach. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI 2015)*, pages 3327–3334. AAAI Press, 2015.
- [Prezenski *et al.*, 2017] Sabine Prezenski, André Brechmann, Susann Wolff, and Nele Russwinkel. A cognitive modeling approach to strategy formation in dynamic decision making. *Frontiers in Psychology*, 8:1335, 2017.