Modeling agents in dialogue systems

Paulo Quaresma and José Gabriel Lopes

Departamento de Informática, Universidade Nova de Lisboa, Portugal

Abstract. In this paper we present an extended logic programming framework that allows to model dialogues between agents with different levels of sincerity and credulity. An agent is modeled by a set of extended logic programming rules and the agents mental state is defined by the well founded model of the extended logic program. Using this modeling process an agent is able to participate in dialogues, updating and revising its mental state after each sentence.

1 Introduction

In order to participate in dialogues, an agent needs the capability to model its mental state. Namely, it is necessary to represent the agent attitudes (believes, intentions, and objectives), world knowledge and temporal, reasoning and behavior rules. In this paper, we propose a logic programming framework that allows the representation of agent models. Agents are defined as logic programs extended with explicit negation and the semantics of the programs is given by the well founded semantics of logic programs with explicit negation (from Alferes and Pereira, 1996). The well founded semantics has a complete and sound top-down proof procedure with polynomial complexity and there is an implemented prototype (Damásio et al., 1994) which allows us to obtain experimental results.

2 Logic programming framework

Logic programs extended with explicit negation are finite set of rules of the form

- $H \leftarrow B_1, \ldots, B_n, \text{not } C_1, \ldots, \text{not } C_m \quad (m \geq 0, n \geq 0)$

where H, B_1, ..., B_n, C_1,..., C_m are objective literals. An objective literal is an atom A or its explicit negation $\neg A$; *not* stands for negation by default; *not* L is a default literal. Literals are objective or default and $\neg\neg L \equiv L$.

The set of all ground objective literals of a program P designates the extended Herbrand base of P and it is represented by $\mathcal{H}(P)$. An interpretation I of an extended program P is represented by $T \cup not\ F$, where T and F are disjoint subsets of $\mathcal{H}(P)$. Objective literals of T are true in I; objective literals of F are false by default in I; objective literals of $\mathcal{H}(P) - I$ are undefined in I. Moreover, if $\neg L \in T$ then $L \in F$. An interpretation I of an extended logic program P is a partial stable model of P iff $\Phi_P(I) = I$ (see Alferes and Pereira, 1996 for the definition of the Φ operator). The well founded model of the program P is the F-least partial stable model of P. The well founded semantics of P is determined by the set of all partial stable models of P.

2.1 Events

As time formalism we use a variation of the Event Calculus (Quaresma, 1997;Missiaen, 1991) that allows events to have an identification and a duration. As a consequence events may occur simultaneously.

The predicate $holds\_at(P,T)$ defines the properties P that are true at a specific time T; the predicate $happens(E,T_i,T_f)$ means that the event E occurred between T_i and T_f; $initiates(E,T,P)$ means that the event E initiates P at time T; $terminates(E,T,P)$ means that the event E terminates P at time T; $persists(T_i,P,T)$ means that P persists since T_i until T (at least); $succeeds(E,T_i)$ means that the event E may occur at time T_i (its pre-conditions are satisfied).

3 Agents mental state

In our proposal, agents are modeled by the well founded model of an extended logic program representing the agents behavior rules, rationality, actions description, world knowledge, and temporal formalism. In this paper we will only analyze the behavior rules for credulity and sincerity. The agent attitudes are: $bel(A,P)$ meaning that agent A believes in proposition P; $int(A,\alpha)$ meaning that agent A intends action α to be executed; $ach(A,P)$ meaning that proposition P is a goal of agent A.

Credulity and sincerity define how an agent accepts/transmits information from/to other agents. The main process defines how believes are transferred:

$$\begin{aligned} holds\_at(bel(A_1,P),T) \leftarrow\ & holds\_at(bel(A_1,bel(A_2,P)),T), \\ & holds\_at(bel(A_1,credulous(A_1)),T), \\ & holds\_at(bel(A_1,sincere(A_2)),T). \end{aligned} \tag{1}$$

This rule defines that an agent believes in a proposition if he believes that another agent believes in it, and if he believes he is credulous and the other agent is sincere. This rule can be changed for more sceptical agents adding more pre-conditions (check if the belief is not contradictory with some previous believes).

4 Example: Non-sincere dialogue

In this section we show an example of a dialogue between sincere and credulous agents (Astérix and the modeled agent Obélix) and a non-sincere agent (Lierix). Our agent — Obélix — as the goal to be happy and he may execute one action to achieve is goal: to beat romans.

The dialogue we want to model is the following:

1. Lierix to Obélix: There are romans at Babaorum!
2. Obélix does not believe because he believes Lierix is not sincere.
3. Astérix to Obélix: There are romans at Petibonum!
4. Obélix believes and he may act in order to go to Petibonum and to beat the romans.

The initial Obélix objective is to be happy ($holds\_at(ach(obelix, happy(obelix)), t_0)$). The domain specific action is (we are using the A-language as action description language):

$$to\_beat\_romans(obélix)\ causes\ happy(obélix)\ if\ at(obélix, Place), at(romans, Place) \quad (2)$$

This action defines that Obélix may beat the romans if he is at a place where the romans are.

In the first sentence, Lierix says to Obélix that there are romans at Babaorum. The following facts are created from the analysis of the sentence (in the rules $lierix$ will be abbreviated by l, $asterix$ by a and $obelix$ by o):

$$happens(e_1, t_1, t_2), act(e_1, inform(l, o, at(romans, babaorum))). \quad (3)$$

However, the information is not transferred because Obélix believes Lierix is not sincere (rule 1).

The second sentence, from Astérix, creates the following facts:

$$happens(e_2, t_3, t_4), act(e_2, inform(a, o, at(romans, petibonum))). \quad (4)$$

In this situation, the belief is transferred and Obélix starts to believe that there are Romans at Petibonum. In this situation, the planning process could identify the need to go to Petibonum as a pre-condition of the action of beating romans. However, the planning process is not analyzed in this paper (in Quaresma and Lopes, 1995 , we propose an abducting procedure that allows the inference of plans)

5 Conclusions

We have proposed an agent modeling process defined over a logic programming framework with a specific semantic (well founded semantics of extended logic programs). This process allows the definition of behavior rules enabling us to model non-well behaved agents.

As future work we intend to integrate this agent modeling framework in a more general architecture allowing a complete representation of dialogues.

References

Alferes, J. J., and Pereira, L. M. (1996). Reasoning with Logic Programming, volume 1111 of Lecture Notes in Artificial Intelligence. Springer.

Damásio, C., Nejdl, W., and Pereira, L. M. (1994). Revise: An extended logic progamming system for revising knowledge bases. In Kaufmann, M., ed., KR'94.

Missiaen, L. (1991). Localized Abductive Planning with the Event Calculus. Ph.D. Dissertation, Univ. Leuven.

Quaresma, P., and Lopes, J. G. (1995). Unified logic programming approach to the abduction of plans and intentions in information-seeking dialogues. Journal of Logic Programming (54).

Quaresma, P. (1997). Inferência de Atitudes em Diálogos. Ph.D. Dissertation, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa. In Portuguese. To appear.