Updating and revising the agents mental state in dialogues

Paulo Quaresma and José Gabriel Lopes
{pq,gpl}@di.fct.unl.pt
Departamento de Informática,
Faculdade de Ciências e Tecnologia,
Universidade Nova de Lisboa,
2825 Monte da Caparica
Portugal

January 25, 2000

Abstract

In this paper we present an extended logic programming framework that allows to model dialogues between different kinds of agents. Namely, it will be shown how this framework is able to handle dialogues between agents with different levels of sincerity, cooperativeness, credulity, and activity.

In this framework an agent/computational system is modeled by a set of extended logic programming rules representing its mental state. These rules describe the agent behavior, attitudes (believes, intentions, and objectives), world knowledge, and temporal and reasoning procedures. The complete mental state is defined by the well founded model of the extended logic program that models the agent.

Using this modeling process an agent is able to participate in dialogues, updating and revising its mental state after each sentence. The revision process includes the capability to remove contradictions in the agent mental state.

1 Introduction

In order to participate in dialogues, an agent/computational system 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 and the definition of update and revise procedures. In a dialogue, these procedures are executed after each event (sentence) and they update the agent model using the information associated with the different speech acts.

Agent models 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 Pereira et al. Alferes and Pereira 1996; Alferes et al. 1995; Alferes 1993). 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.

At each time instant, the agent mental state is given by the well founded model of the logic program that models the agent. In a dialogue, after each sentence, it is necessary to update the agent model with the new information. This process is done through the update of the logic program with the facts that describe the events: identification of the time and speech acts associated with each event.

However, the update process may create a contradictory mental state. For instance, it is possible that an event initiates a belief that is contradictory with some previous believes. In these situations, it is necessary to revise the agent mental state, terminating the attitudes that supported the contradiction.

The updated and, eventually, revised agent mental state may be used as the input of a planning procedure that tries to satisfy the agent objectives.

The proposed framework has some advantages over many classical dialogue systems: Litman and Allen (Litman 1985; Litman and Allen 1987; Allen et al. 1991), Carberry (Carberry 1985; Carberry 1988), Pollack (Pollack 1986; Pollack 1990). In fact, it supports the recognition of attitudes using a formal framework with a specific semantic. Moreover, it allows the representation of several kinds of users and it supports the existence of contradictory states, eliminating the contradiction when necessary. These characteristics allow this framework to handle a wider range of dialogues, dealing with error situations and non-well behaved agents.

In the next section, the logic programming framework is briefly described. In section 3 we present the agent modelling process, with a special focus on the capability to model agents with different levels of sincerity, cooperativeness, credulity, and activity. The procedures to update and revise the agents mental state after each event are described in section 4 and 5. The planning recognition process is described in section 6. Finally, in section 7 some conclusions and open problems are pointed out.

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 $H(P)$. An interpretation I of an extended program P is represented by $T \cup not\ F$, where T and F are disjoint subsets of $H(P)$. Objective literals of T are true in I; objective literals of F are false by default in I; objective literals of $H(P) - I$ 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.

Pereira et al. (Alferes and Pereira 1996; Alferes et al. 1995) showed that every non-contradictory program has a well founded model and they also presented a complete and sound top-down proof procedure for several classes of programs.

In their work, Pereira et al., proposed a revision process that restores consistency for contradictory programs, taking back assumptions of the truth value of negative literals. As it will be described in section 4, we also use this approach in order to revise the agents mental state.

2.1 Events

The agent modeling process must be able to deal with time and events. In fact, it is very important that agents have the capability to reason about their mental state at a given time point. They should also be able to change their mental state as a consequence of some external or internal events.

As a time formalism we propose a variation of the Event Calculus (Shanahan 1989; Eshghi 1988; Missiaen 1991) that allows events to have an identification and a duration. As a consequence events may occur simultaneously.

The predicate *holds_at* defining the properties that are true at a specific time is:

$$\begin{aligned} holds\_at(P,T) \leftarrow\ & happens(E,T_i,T_f), && (1)\\ & initiates(E,T_P,P), \\ & T_P < T, \\ & persists(T_P,P,T). \\ persists(T_P,P,T) \leftarrow\ & not\ clipped(T_P,P,T). && (2)\\ clipped(T_P,P,T) \leftarrow\ & happens(C,T_{ci},T_{cf}), && (3) \end{aligned}$$

$$
\begin{aligned}
& & & terminates(C, T_C, P), \\
& & & not\ out(T_C, T_P, T). \\
out(T_C, T_P, T) & \leftarrow & & T \leq T_C. & (4) \\
out(T_C, T_P, T) & \leftarrow & & T_C < T_P. & (5)
\end{aligned}
$$

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 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).

Note that a property P is true at a time T ($holds\_at(P, T)$), if there is a previous event that initiates P and if P persists until T. P persists until T if it can not be proved by default the existence of another event that terminates P before the time T.

We need additional rules for the relation between not holding a property and holding its negation and we also need to define the relation between the two kinds of negation:

$$\neg holds\_at(P, T) \leftarrow holds\_at(\neg P, T). \tag{6}$$

$$\neg holds\_at(P, T) \leftarrow not\ holds\_at(P, T). \tag{7}$$

The predicates need to be related by some integrity rules:

1. Events can not initiate and terminate a property at the same time:

$$\leftarrow initiates(E, T, P),\ terminates(E, T, P). \tag{8}$$

2. Events can not initiate/terminate a property and its negation:

$$\leftarrow initiates(E, T, P), initiates(E, T, \neg P). \tag{9}$$

$$\leftarrow terminates(E, T, P), terminates(E, T, \neg P). \tag{10}$$

3. Events can not be associated to different time intervals:

$$
\begin{aligned}
\leftarrow\ & happens(E, T_{1i}, T_{if}), \\
& happens(E, T_{2i}, T_{2f}), \\
& T_{1i} = T_{2i}, \\
& not(T_{if} = T_{2f}).
\end{aligned} \tag{11}
$$

4. Events can not have a negative duration:

$$\leftarrow happens(E, T_i, T_f),\ not(T_i \leq T_f). \tag{12}$$

5. Events must have an associated action:

$$\begin{aligned} \leftarrow \quad & happens(E, T_i, T_f), \\ & not(act(E, A)). \end{aligned} \tag{13}$$

6. Properties must be initiated by some event:

$$\begin{aligned} \leftarrow \quad & holds\_at(P, T), \\ & not(ev\_gen(P, T)). \\ ev\_gen(P, T) \quad \leftarrow \quad & happens(E, T_i, T_f), \\ & initiates(E, T_p, P), \\ & T_i \leq T_p \leq T, \\ & persists(T_p, P, T). \end{aligned} \tag{14}$$

7. Events can not occur if the pre-conditions are not satisfied:

$$\leftarrow happens(E, T_i, T_f),\ not\ succeeds(E, T_i). \tag{15}$$

3 Agents mental state

In our proposal, agents are modeled by the well founded model of an extended logic program with the following structure:

1. Rationality rules (RR). These rules describe the relation between the different attitudes (believes, intentions, and objectives).
2. Behavior rules (BR). These rules define the agent activity, cooperativeness, credulity, and sincerity.
3. Actions description (Ac). These rules describe the actions that may be executed by the agent. In the domain of dialogues, these rules describe the speech acts, their pre-conditions and effects.
4. A temporal formalism (T). These are the rules presented in the previous section.
5. World knowledge (WK).These rules describe the agent world knowledge: entities, taxonomies, ...

In the next subsections we will analyze the first two structures: rationality rules and behavior rules.

3.1 Rationality rules

These rules define relations between the agents attitudes: believes (*bel*), objectives (*ach*), and intentions (*int*).

The main relations are (for related work see Bratman 1990; Cohen and Levesque 1990a; Cohen and Levesque 1990b; Perrault 1990):

- Integrity

$$\bot \leftarrow holds\_at(bel(A,P),T),\ holds\_at(bel(A,\neg P),T). \tag{16}$$

$$\bot \leftarrow holds\_at(ach(A,P),T),\ holds\_at(ach(A,\neg P),T). \tag{17}$$

- Consistency

$$\neg holds\_at(bel(A,\neg P),T) \leftarrow holds\_at(bel(A,P),T). \tag{18}$$

$$\neg holds\_at(ach(A,\neg P),T) \leftarrow holds\_at(ach(A,P),T). \tag{19}$$

- Introspection

$$\bot \leftarrow holds\_at(bel(A,P),T), holds\_at(bel(A,\neg bel(A,P)),T). \tag{20}$$

$$\bot \leftarrow \neg holds\_at(bel(A,P),T), holds\_at(bel(A,bel(A,P)),T). \tag{21}$$

- Necessity

$$holds\_at(bel(A,P),T) \leftarrow holds\_at(P,T). \tag{22}$$

3.2 Behavior rules

These rules allow the definition of the agent behavior. As behavior properties we have considered the credulity, sincerity, activity, and cooperativeness.

The behavior properties are represented by agent believes (about himself and about the other agents). For instance, an agent a believes at a time t that the agent b is sincere, credulous, cooperative and reactive.

$$holds\_at(bel(a,sincere(b)),t).$$
$$holds\_at(bel(a,credulous(b)),t).$$
$$holds\_at(bel(a,cooperative(b)),t).$$
$$holds\_at(bel(a,reactive(b)),t).$$

Note that this approach allows an agent to have different behaviors depending of the time instant. Moreover, an agent may change his believes about the others behavior. However, in this paper we will not discuss the events that may contribute to these changes.

3.2.1 Credulity

Credulity defines how an agent accepts information from other agents. The main process defines how believes are transferred:

$$holds\_at(bel(H,P),T) \leftarrow holds\_at(bel(H,bel(S,P)),T), holds\_at(bel(H,credulous(H)),T). \tag{23}$$

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

The credulity property has also consequences over the description of the speech acts. In fact, the effect of a speech act depends on the model that the hearer has of the speaker.

In this paper, we will show only the effect of the *inform* speech act:

$$initiates(E,T_f,bel(H,bel(S,P))) \leftarrow act(E,inform(S,H,P)), happens(E,T_i,T_f), holds\_at(bel(H,sincere(S)),T). \tag{24}$$

3.2.2 Sincerity

Sincerity defines what is the relation between what agents speak and what they believe.

The effect of a *inform* speech act for a sincere agent is:

$$initiates(E,T_f,bel(S,bel(H,bel(S,P)))) \leftarrow act(E,inform(S,H,P)), happens(E,T_i,T_f), holds\_at(bel(S,sincere(S)),T), holds\_at(bel(S,P),T). \tag{25}$$

For a non sincere speaker, the effect will be (note that the speaker informs the incorrect truth value of P):

$$initiates(E,T_f,bel(S,bel(H,bel(S,\neg P)))) \leftarrow act(E,inform(S,H,\neg P)), happens(E,T_i,T_f), holds\_at(bel(S,sincere(S)),T), holds\_at(bel(S,P),T). \tag{26}$$

3.2.3 Cooperativeness

This property defines how intentions and objectives are transferred between agents.

For a cooperative agent:

$$holds\_a(int(H, A), T) \leftarrow holds\_at(bel(H, int(S, A)), T), \quad holds\_at(bel(H, cooperative(H)), T). \tag{27}$$

$$holds\_a(ach(H, P), T) \leftarrow holds\_at(bel(H, ach(S, P)), T), \quad holds\_at(bel(H, cooperative(H)), T). \tag{28}$$

For a non cooperative agent:

$$holds\_a(\neg int(H, A), T) \leftarrow holds\_at(bel(H, int(S, A)), T), \quad holds\_at(bel(H, non\_cooperative(H)), T). \tag{29}$$

$$holds\_a(\neg ach(H, P), T) \leftarrow holds\_at(bel(H, ach(S, P)), T), \quad holds\_at(bel(H, non\_cooperative(H)), T). \tag{30}$$

3.2.4 Activity

An agent may be pro-active or reactive. A pro-active agent has some objectives that were not transferred from other agents: the agent has his own motivations. On the contrary, a reactive agent acts only as a response to other agents acts.

A pro-active agent as the following property (he has his own objectives):

$$holds\_at(bel(A, proactive(A)), T) \leftarrow holds\_at(ach(A, P), T), \quad not\ holds\_at(bel(A, ach(B, P)), \quad not(A = B). \tag{31}$$

4 Updating an agent mental state

The agent mental state, as it was defined in the previous sections, must be updated after each event.

This process is defined in the following way:

Definition 1 *Let* m *be the agent model, :* $m = < RR, Ac, T, BR, WK >$*, where* RR *are the rationality rules,* Ac *are the rules defining the domain actions,* T *are the temporal axioms,* BR *are the behavior rules and* WK *are the works knowledge rules.*

The update function $update : M \times E^n \longrightarrow M$*, is defined such as:*

1. $update(m, e_1 \times ... \times e_n) = < RR_1, Ac_1, T_1, BR_1, WK_1 >$
2. $RR_1 = RR$
3. $Ac_1 = Ac$
4. $T1 = T$
5. $BR_1 = BR$
6. $WK_1 = WK \cup \{act(e_1, a_1), happens(e_1, t_1, t'_1)..., act(e_n, a_n), happens(e_n, t_n, t'_n)\}$, the world description is updated with the new events $e_1, ..., e_n$.

The new agent attitudes are the properties at ($bel/2$, $ach/2$, and $int/2$) that hold in the new model:

$$holds\_at(at, t) \in WFM(RR \cup Ac \cup T \cup BR \cup WK_1)$$

The update process may initiate some attitudes which are inconsistent with the previous mental state. In this situation, the model must be revised, and some attitudes should be terminated. In the next section this process will be described.

Using this update process it is possible to handle situations where the agent believes in one property at a given time point, and then he changes his belief. Note that the previous belief is not lost (the agent knows at which time interval it hold).

As an example, agent a believes at time t_1, that Kathy is at the hospital:

$$holds\_at(bel(a, at(hospital, kathy)), t_1).$$

At a greater time point, $t_2 > t_1$, he is informed that she is at home:

$$happens(e_1, t_2, t_2).$$
$$act(e_1, inform(b, a, at(home, kathy))).$$

If agent a is credulous, he will adopt the new information (using the speech act $inform$ presented previously):

1.

$$holds\_at(bel(a, at(hospital, kathy)), t_2).$$

2.

$$holds\_at(bel(a, at(home, kathy)), t_2).$$

However, there should be an integrity constraint stating that is contradictory to believe that an agent may be at two different places at the same time:

$$\begin{aligned}\bot \quad & \leftarrow holds\_at(bel(A, at(B, L_1)), T), \\ & holds\_at(bel(A, at(B, L_2)), T), \\ & L_1 \neq L_2.\end{aligned}$$

In this situation, the model must be revised and one of the non-contradictory solution must be chosen (see next section).

5 Revising Mental States

As it was shown, the update process may introduce contradiction in the agents mental state. In fact, the new events ($happens(e_i, t, t'), act(e_i, a_i)$, with $1 \leq i \leq n$), may introduce contradiction due to two different causes:

1. Contradiction caused by the new facts;
2. Contradiction caused by the effects of the new facts.

The first type of contradiction is caused by the violation of integrity constraints relating the description of facts (happens and act) and is analyzed at subsection 5.1. The second cause of contradiction is associated with the effects of the new events (see section 5.2).

5.1 Contradictory facts

Contradiction may be caused by the description of the new events. As an example, suppose the open window action (very simplified version):

$$\begin{aligned} enabled(E, T_i) \quad & \leftarrow \quad act(E, open\_window), \\ & \qquad holds\_at(closed\_window, T_i). \end{aligned} \tag{32}$$

$$\begin{aligned} initiates(E, T_f, opened\_window) \quad & \leftarrow \quad happens(E, T_i, T_f), \\ & \qquad act(E, open\_window), \\ & \qquad holds\_at(closed\_window, T_i). \end{aligned} \tag{33}$$

Suppose the window is open at time t_0:

$$happens(e_0, t_0, t_0). \tag{34}$$

$$act(e_0, start). \tag{35}$$

$$initiates(e_0, t_0, opened\_window). \tag{36}$$

The agent recognized the following event:

$$happens(e, t, t'). \tag{37}$$
$$act(e, open\_window), \tag{38}$$
$$t_0 \leq t \leq t'. \tag{39}$$

In these conditions, the model is contradictory because the integrity constraint 15 is violated:

$$\bot \ \leftarrow happens(E, T_i, T_f), not\ enabled(E, T_i).$$

In fact, it is not possible to infer $enabled(e, t)$, because the following property can not be inferred:

$$holds\_at(janela\_fechada, t).$$

Contradiction can be removed using two approaches:

1. Abducting one action that allows the satisfaction of some desired properties (for instance, a previous example that had closed the window);
2. Assuming that there was an incorrect event recognition.

The first approach is already supported by the proposed framework through the use of the rules that allow the abduction of events ($happens/3$ and $act/2$). These rules allow to avoid contradiction whenever is possible to abduce actions that create the desired conditions.

However this process does not guarantee that the model is always non-contradictory. In this case it is necessary to use the contradiction removal process described by Alferes and Pereira 1996. The revisable predicates are:

$$rev = \{not\ happens(E, T_i, T_f), not\ act(E, A)\}$$

This set of revisables mean that is possible to revise the existence of events and its associated actions, whenever necessary. If the new events can not be incorporated in the agent model, then its existence should be revised (the model should always be non-contradictory).

The revision process allows the definition of the preferred non-contradictory solutions (Damásio et al. 1994). We can define the revision process to revise the newest/oldest events that support contradiction.

5.2 Contradictory Mental State

Contradiction may also be cause by the effects of the new events (these effects may violate some integrity constraints).

This kind of contradiction can be detected through the calculus of the contradiction support set of an extended logic program (Alferes and Pereira 1996).

Contradictions caused by the effects of events are associated with integrity constraint rules of the following form:

$$\perp \leftarrow holds\_at(P_1, T), holds\_at(P_2, T).$$

In this situation, the revising process allows the definition of preference rules over the properties (preferring a non-contradictory solution where some properties hold). Suppose that an event initiated property P_2, and property P_1 is also valid; a possible approach could be to revise P_1 (or P_2) revising the assumption that it hasn't terminated.

As an example, suppose the situation of the previous section where agent a believes Kathy is at the hospital and he is informed that she is at home:

$$holds\_at(bel(a, em(hospital, kathy)), t_2).$$
$$holds\_at(bel(a, em(home, kathy)), t_2).$$

We have the integrity constraint:

$$\begin{aligned} \perp \leftarrow \quad & holds\_at(bel(X, at(L_1, Y)), T), \\ & holds\_at(bel(X, at(L_2, Y), T), \\ & not(L_1 = L_2). \end{aligned} \tag{40}$$

The revision process obtains the non-contradictory solutions and it also obtains the preferred solution (accordingly with a pre-defined order between properties):

1. Terminate the believe that Kathy is at the hospital;

2. Terminate the believe that Kathy is at home.

6 Plan Recognition

The update and revise processes presented in the previous sections allow the definition of the agent mental state, after each event. Using this model it is possible to try to recognize the other agents plans in order to participate actively in the interaction process.

An agent plan, at a given time, is the set of intentions about the actions he wants to be realized:

Definition 2 *From the agent a point of view, $P_a(b,t)$ is the agent's b plan at time t, and is defined by:*

$$P_a(b,t) = \{int(b, X) : holds\_at(bel(a, int(b, X)), t) \in WFM(M_a)\}$$

where M_a is the model of a.

After each event, an agent tries to recognize the other agents plans and he uses them in order to plan his own actions.

An agent plan is given by $P_a(a,t)$, representing the actions he wants to be performed:

$$P_a(a,t) = \{int(a,X) : holds\_at(bel(a,int(a,X)),t) \in WFM(M_a)\}$$

Plan recognition is, in this framework, the inference of the believes about intentions. This plan recognition strategy may be seen as a mixture of the classical-STRIPS approaches (Fikes and Nilsson 1971; Litman 1985) and the mental states approach of Pollack (Pollack 1990). Actions are described in terms of their pre-conditions and effects (as in classical planning), but the recognition process is based on a attitude theory defining the agents mental states and their relation with the speech acts.

The next step is to generate an agent plan. This process is done using an abductive planning strategy that abduces the actions needed to create the desired states. However this process is not described in this paper (see Quaresma and Lopes 1995; Quaresma 1997).

7 Conclusions

We have proposed an agent modeling process with the following characteristics:

1. It was defined over a logic programming framework with a specific semantic (well founded semantics of extended logic programs);

2. It has a complete and sound top-down proof procedure;

3. It allows the definition of reasoning and behavior rules. These rules allow the modeling of non-well behaved agents;

4. It has an update and revise procedure defined for any event that may occur;

5. It may be the base of a planning process that allows the participation of agents in dialogues.

This framework has some advantages over previous systems because it is formal, it may be implemented (in fact we have a working prototype), and it supports a wide range of dialogue situations.

However, there are many problems to be deal as future work.

First, and as it was pointed out in the previous section, we have not analyzed the integration of the modeling process with the planning process and the natural language generation phase. Moreover, it was not discussed the problem of the recognition of speech acts from natural language sentences. These tasks are pre-conditions for the construction of a robust natural language processing system.

As future work we also intend to integrate this agent modeling framework in a more general architecture allowing a complete representation of dialogues. Namely, the architecture should be able to deal with cycles of conversation and clarification dialogues (Lopes 1991; Quaresma and Lopes 1992).

References

Alferes, J. J.: 1993, *Semantics of Logic Programs with Explicit Negation*, Ph.D. thesis, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa

Alferes, J. J., Damásio, C., and Pereira, L. M.: 1995, A logic programming system for nonmonotonic reasoning, *Journal of Automated Reasoning* 14, 93–147

Alferes, J. J. and Pereira, L. M.: 1996, *Reasoning with Logic Programming*, Vol. 1111 of *Lecture Notes in Artificial Intelligence*, Springer

Allen, J., Kautz, H., Pelavin, R., and Tenenberg, J.: 1991, *Reasoning about Plans*, Morgan Kaufman Publishers, Inc.

Bratman, M.: 1990, *What is Intention?, in Intentions in Communication*, MIT

Carberry, S.: 1985, *Pragmatic Modeling in Information System Interfaces*, Ph.D. thesis, University of Delaware

Carberry, S.: 1988, Modelling the user's plans and goals, *Computational Linguistics* 14(3), 23–37

Cohen, P. and Levesque, H.: 1990a, Intention is choice with commitment, *Artificial Intelligence* 42(3)

Cohen, P. and Levesque, H.: 1990b, *Persistence, Intention, and Commitment, in Intentions in Communication*, pp 33–70, MIT

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

Eshghi, K.: 1988, Abductive planning with event calculus, in *Proceedings of the International Conference on Logic Programming*

Fikes, R. E. and Nilsson, N. J.: 1971, Strips: A new approach to the application of theorem proving to problem solving, *Artificial Intellligence* (2), 189–208

Litman, D. and Allen, J.: 1987, A plan recognition model for subdialogues in conversations, *Cognitive Science* (11), 163–200

Litman, D. J.: 1985, *Plan Recognition and Discourse Analysis: An Integrated Approach for Understanding Dialogues*, Ph.D. thesis, Dep. of Computer Science, University of Rochester

Lopes, J. G.: 1991, Architecture for intentional participation of natural language interfaces in conversations, in C. Brown and G. Koch (eds.), *Natural Language Understanding and Logic Programming III*, North Holland

Missiaen, L.: 1991, *Localized Abductive Planning with the Event Calculus*, Ph.D. thesis, Univ. Leuven

Perrault, R.: 1990, *An Application of Default Logic to Speech Act Theory, in Intentions in Communication*, Chapt. 9, pp 161–186, MIT

Pollack, M.: 1990, *Plans as Complex Mental Attitudes, in Intentions in Communication*, Chapt. 5, pp 77–104, MIT

Pollack, M. E.: 1986, *Inferring Domain Plans in Question-Answering*, Ph.D. thesis, Dep. of Computer and Information Science, University of Pennsylvania

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

Quaresma, P. and Lopes, J. G.: 1992, A two-headed architecture for intelligent multimedia man-machine interaction, in *B. de Boulay and V. Sgurev (eds). Artificial Intelligence V - methodology, systems, applications*, North Holland

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)

Shanahan, M. P.: 1989, Prediction is deduction but explanation is abduction, in *Proceedings of the IJCAI*