Abduction of Plans and Intentions in Dialogues

Paulo Quaresma* pq@fct.unl.pt
José Gabriel Lopes gpl@fct.unl.pt

Artificial Intelligence Center, UNINOVA
2825 Monte da Caparica, Portugal
July 7, 1993

Abstract

A robust man-machine interaction implies the capability to infer the beliefs and intentions of each active agent. In this article it will be proposed a framework that supports the recognition of plans and intentions through abductive inferences over discourse sentences. The possible actions, world knowledge, events and states are represented by extended logic programs (LP with explicit negation) and the abductive inference process is modeled by the framework proposed by Pereira ([9]) which is based on the Well Founded Semantics augmented with explicit negation (WFSX) and contradiction removal semantics (CRSX). It will be shown how this framework supports abductive planning with Event Calculus ([3]) and some classical examples will be shown ([6, 10]) in the domain of information-seeking dialogues. Finally, some open problems and future work will be pointed out.

1 Introduction

A robust man-machine interaction implies the capability to infer the beliefs, intentions and plans of each active agent.

In order to deal with these problems there has been done some work with different approaches. One major approach follows the classical planning scheme developed in the STRIPS ([4]) and NOAH ([12]) model. In this model each plan is defined as a sequence of actions and each action is composed by an head, pre-conditions, constraints, effects and sub-actions. The inference of plans (a list of user actions) is done through the use of a library of plans and actions, some heuristic rules and the user possible goals. This approach has been used by Litman and Allen ([6, 5]) in order to infer plans behind speech acts in dialogues. A different approach was followed by Pollack ([10, 1]) which models plan as mental states and tries to abduce the mental attitudes behind each speech act.

In this paper we follow a general approach which will allow us to handle both models. This approach uses non-monotonic reasoning, namely default

*Owns a scholarship from JNICT, reference n^o BD/1766/IA

and abductive reasoning, as the basic inference process. In order to support this kind of reasoning we have used the event calculus to represent events, time and actions and a logic programming framework with a defined semantics, Well Founded Semantics of eXtended Logic Programs (WFSX) augmented with Contradiction Removal Semantics (CRSX) from the work of Pereira ([9]). This framework extends logic programming and alllows the modelling of several kinds of non-monotonic reasoning, namely default, abductive and hypothetical reasoning. Furthermore, it allows the removal of contradictions allowing the desambiguation of dialogues. In section 2 a description of the framework showing how non-monotonic reasoning is dealt with is given. In section 3 the process of abductive planning with event calculus is described and in section 4 it is shown how this framework is able to handle the same kind of problems that Litman and Pollack handle. Finally in section 5 some open problems and future work will be pointed out.

2 Logic Programming Framework

In order to reason about plans and attitudes we need to model actions, events, states and world knowledge. In this framework they are modeled by extended logic programs which are a set of rules and integrity rules having the form:

- $H \leftarrow B_1, \ldots, B_n, notC_1, \ldots, notC_m (m \geq 0, n \geq 0)$

where H, B_1, ..., B_n, C_1, C_m are classical literals. A classical literal is either an atom A or its explicit negation $\neg A$. *not* stands for the negation by failure (NAF). In integrity rules H is the symbol $\bot$ (contradiction).

Default reasoning can be modeled adding to the program rules of the form:

- Normally $A(X)$ implies $B(X)$

which can be written as:

1. B(X) $\leftarrow$ A(X), not ab(X)

which states that if it's not possible to prove the abnormality of X then B should hold.

With a slight change it is possible to handle hypothetical reasoning:

- Quakers might (or not) be pacifists

which can be written as:

1. pacifist(X) $\leftarrow$ quaker(X), hypqp(X)
2. hypqp(X) $\leftarrow$ not $\neg$ hypqp(X)
3. $\neg$ hypqp(X) $\leftarrow$ not hypqp(X)

which states that quakers are pacifists if it's not possible to prove (by NAF) explicitly that they are not (and vice-versa).

Abductive reasoning is modeled with rules:

- F or its negation can be assumed

which can be written as:

1. $F \leftarrow not \neg F$
2. $\neg F \leftarrow not F$

which state that if it's not possible to prove $\neg F$ then F should hold (and vice-versa).

Using this approach it's possible to create an abductive program from an abductive theory (P, Ab) by adding to the program P for all literals L in the abducible list Ab two rules of the form:

1. $L \leftarrow not \neg L$
2. $\neg L \leftarrow not L$

3 Abductive Planning with Event Calculus

In order to represent and reason about actions and events it was used the event calculus with some changes proposed by Shanahan ([13], Eshgi ([3]) and Missiaen ([8]). In this framework events are time-points with no duration and are represented by the time at which they occur. Time-points are ordered by the precedence relationship $<$. Events are action instances and they are related to the actions by the predicate *act/2*. There are also other predefined predicates, namely the predicate *initiates/2* which means that an event initiates a given property, *terminates/2* meaning that the property was terminated by the event and *succeeds/1* meaning that an event can happen (its pre-conditions are satisfied). Within this framework states are represented by the properties that hold at a given time.

The following logic program is proposed by Missiaen in order to describe what properties hold at a given time:

$$\begin{aligned}
holds_at(P,T) &\leftarrow happens(E), initiates(E,P), && (1)\\
&\quad\ succeeds(E), E < T, persists(E,P,T).\\
persists(E,P,T) &\leftarrow notclipped(E,P,T). && (2)\\
clipped(E,P,T) &\leftarrow happens(C), terminates(C,P), && (3)\\
&\quad\ succeeds(C), notout(C,E,T).\\
out(C,E,T) &\leftarrow (T = C; T < C; C < E). && (4)
\end{aligned}$$

which states that the property P holds at time T if there was an event E that happened before T, if that event initiates P and if P persists until T. Property

P persists until T if it is not possible to prove (by NAF) the existence of an event C that terminates the property P before T. Note that if it is not possible to prove that the event C is outside $[E, T[$ then it is assumed to be inside. This rule is different from the one proposed by Shanahan (if it can not be proved to be inside $[E, T[$ then assume it is outside) but, as shown by Missiaen, it prevents incorrect time orderings when there are partial time ordering between time-points.

In order to infer the user plans it's necessary to abduce events (related with actions) and their temporal ontology because plans in this framework are described by sequences of actions. The set of abducible predicates in our framework is composed by:

Ab = {happens/1, act/2, </2}

This set if abducibles allow the framework to abduce the events that explain the observed effects (properties that hold at a given time).

The representation of actions in this framework can be done by a translation mechanism that allows each action to be described by a set of statements in the form:

- A causes F if P_1, ..., P_n

This statement means that action A causes the effect F if the pre-conditions P_1, ..., P_n hold. The statement can be translated by the following rules:

1. succeeds(E) $\leftarrow$ act(E, A), holds-at(P_1, E), ..., holds-at(P_n, E).
2. initiates(E, F) $\leftarrow$ act(E, A).

These rules state that an event E associated with an action A at a time T succeeds if the pre-conditions hold at that time and as a consequence of this event the property F will hold in the future.

4 Example

In the following two subsections it will be presented two examples solved with the unified extended logic framework proposed. In the first example there is missing information that was not conveyed by the user. In the second example it will be pointed out how to solve dialogue situations where incorrect knowledge is assumed by the speaker.

4.1 Train Dialogue

Suppose the following example (from [6]):

- Passenger: The eighty-fifty to Montreal?
- Clerk: Eighty-fifty to Montreal? Gate Seven.

- Passenger: Where is it?
- Clerk: Down this way to the left. Second one on the left.
- Passenger: Ok.

In order to be able to interpret this example we have first to define the library of actions (domain actions, speech acts) needed to make inferences. We have used the actions defined in [6] and the translation mechanism described in the previous section.

Namely the following actions were defined:

- First rule:

goto(agent, location, time) *causes* at(agent, location, time).

This rule means that if an agent goes to a specific place at a given time then he will be at that place.

Accordding to with the translation mechanism described, this rule would originate the following logic programming rules:

$$succeeds(E) \leftarrow act(E, goto(agent, location, time)) \quad (5)$$

$$initiates(E, at(agent, location, time)) \leftarrow act(E, goto(agent, location, time)) \quad (6)$$

- Second rule:

meet(agent, arriveTrain) *if* goto(agent, gate(arriveTrain), time(arriveTrain)).

This rule means that if an agent goes to a specific gate of an arrival train at the arriving time then s/he will meet that train.

This rule would originate:

$$succeeds(E) \leftarrow act(E, meet(agent, arriveTrain)), \quad (7)$$
$$holds\_at(goto(agent, gate(arriveTrain), time(arriveTrain)), E).$$

- Third rule:

board(agent, departTrain) *if*

goto(agent, gate(departTrain), time(departTrain)), geton(agent, departTrain).

This rule means that if an agent goes to a gate of a departing train and gets on the train then s/he will board the train.

Using the translation mechanism we would have:

$$succeeds(E) \leftarrow act(E, meet(agent, arriveTrain)), \quad (8)$$
$$holds\_at(goto(agent, gate(arriveTrain), time(arriveTrain)), E).$$

- Forth rule:

take-train-trip(agent, departTrain, destination) *if*

buy-ticket(agent, clerk, ticket), board(agent, departTrain).

This rule means that an agent takes a train trip if he buys a ticket to that trip and boards the departing train.

Using the translation mechanism we would have:

$$\begin{aligned} succeeds(E) \leftarrow\ & act(E, take\_train\_trip(agent, departTrain, destination)), \\ & holds\_at(buy\_ticket(agent, clerk, ticket), E) \\ & holds\_at(board(agent, departTrain), E). \end{aligned} \tag{9}$$

It was also needed to define rules for some speech acts:

- First rule:

inform(speaker, hearer, proposition) *causes*

know(hearer, proposition), know(hearer, know(speaker, proposition))

if

know(speaker, proposition), surface-inform(speaker, hearer, proposition).

If a speaker knows a specific proposition and informs the hearer about that proposition then the hearer will learn the proposition and he will know that the speaker knows it.

Using the translation mechanism we would have:

$$\begin{aligned} succeeds(E) \leftarrow\ & act(E, inform(S, H, P), && (10) \\ & holds\_at(know(S, P), E), \\ & holds\_at(surface\_inform(S, H, P), E). \\ initiates(E, know(H, P)) \leftarrow\ & act(E, inform(S, H, P)), && (11) \\ initiates(E, know(H, know(S, P)) \leftarrow\ & act(E, inform(S, H, P)). && (12) \end{aligned}$$

- Second rule:

informref(speaker, hearer, term, proposition) *causes*

knowref(hearer, term, proposition),

if

knowref(speaker, term, proposition), know(hearer, proposition).

If a speaker knows about a property of a proposition then, if he informs the hearer about that property, the hearer will learn that fact.

Using the translation mechanism we would have:

$$\begin{aligned} succeeds(E) \quad &\leftarrow \quad act(E, informref(S,H,T,P)), \\ &\qquad holds\_at(knowref(S,T,P),E), \\ &\qquad holds\_at(know(hearer, proposition),E). \end{aligned} \tag{13}$$

$$initiates(E, knowref(H,T,P)) \quad \leftarrow \quad act(E, informref(S,H,T,P)). \tag{14}$$

- Third rule:

request(speaker, hearer, action) *causes*

want(hearer, action), know(hearer, want(speaker, action)),

if

want(speaker, action),

(surface-request(speaker, hearer, action) or

surface-request(speaker, hearer, informif(hearer, speaker, cando(hearer, action)) or

surface-inform(speaker, hearer, ¬ cando(speaker, action)) or

surface-inform(speaker, hearer, want(speaker, action))).

If a speaker wants an action to be done he can perform one of the following four speech acts:

- Request the hearer to do the action;
- Request the hearer to inform her/him if the hearer is able to do the action;
- Inform the hearer that s/he is not able to do the action;
- Inform the hearer that s/he wishes the action to be done.

causing the hearer to want to do the action (it is a cooperative dialogue) and to know that the hearer wants the action to be done.

Using the translation mechanism we would have:

$$\begin{aligned} suceeds(E) \quad &\leftarrow \quad act(E, request(s,h,a)), \\ &\qquad holds\_at(want(s,a),E), \\ &\qquad (holds\_at(surface\_request(s,h,a),E) or \\ &\qquad holds\_at(surface\_request(s,h,informif(h,s,cando(h,a)),E) or \\ &\qquad holds\_at(surface\_inform(s,h,\neg cando(s,a)),E) or \\ &\qquad holds\_at(surface\_inform(s,h,want(s,a)),E)). \end{aligned} \tag{15}$$

$$initiates(E, want(h, a)) \leftarrow act(E, request(s, h, a)). \quad (16)$$
$$initiates(E, know(h, want(s, a)) \leftarrow act(E, request(s, h, a)). \quad (17)$$

There were also defined meta-actions, allowing the inference of meta-plans and allowing the system to reason about the structure of the dialogue: *continue-plan*, *identify-parameter*, *correct-plan* and *modify-plan* ([6, 5]).

After this process the first user utterance can generate the following facts:

1. happens(e1)
2. act(e1, surface_request(p, c, informref(c, p, T, train(montreal))))
3. holds-at(knowref(p, time(8:50), train(montreal)), e1)

describing that the passenger requested to be informed by the clerk about a given reference of the Montreal train. The passenger also knows the time of the departure or arrival of that train.

It is possible to make inferences about the present, past and future situations. For instance it can be inferred what action might be the consequence of the first speech act event:

1. happens(e2), e1 < e2, act(e2, A),
2. A = act(e2, request(p, c, informref(c, p, T, train(montreal))))

In order to make these inferences it was used the rule 15 and there were abduced the following facts:

1. happens(e0), e0 < e1,
2. act(e0, want(p, informref(c, p, T, train(montreal))))

It can also be inferred what properties and actions will hold as a consequence of that action:

1. happens(e3), e2 < e3,
2. act(e3, A), holds-at(X, e3)
3. A = act(e3, infref(c, p, gate(7), train(montreal)))
4. X = want(c, informref(c, p, T, tr(montreal))) and
5. X = know(c, want(p, informref(c, p, T, tr(montreal))))

This inference was done using the facts already inferred and the rules 13, 16 and 17 and it represents the fact that the clerk wants to inform the passenger and that he knows that the passenger wants to be informed. It can also be inferred that the clerk will inform the passenger about the missing property, the gate where the train departs (it is a cooperative information-seeking dialogue).

Furthermore it's possible to repeat this process and to infer what might happen next:

1. happens(e4), e3 < e4,
2. holds-at(X, e4), act(e4, A)
3. A = goto(p, gate(7))
4. X = knowref(p, gate(7), tr(montreal))

In this situation it's possible that the passenger will go to the Montreal gate.

If we continue to repeat this process, it's possible to abduce that the passenger will go to the gate in order to take a train trip (the desambiguation between meeting and boarding can be done through the access of a train knowledge base with information about trains, departure and arrival hours and destinations) and that the action is a first step of the meta-plan *introduce-plan*.

4.2 Computer mail messages

In this example (adapted from [10]) a dialogue with an error situation is pointed out. In fact the user plan is incorrect in order to achieve the user goal.

Example:

- Q: I want to talk to Kathy? Do you know the phone number at the hospital?
- A: She's already been discharged. Her home number is 555-8321.

Accordingly with Pollack's approach, the plan ascription is made through the inference of the agents beliefs and intentions. As epistemic operators (describing the agents mental states) there are defined the following operators ([1, 2]):

INT(a, α): agent a intends to do α

BEL(a, p): agent a beliefs that p is true

ACH(a, p):agent a beliefs p will be true as a consequence of its actions

EXP(a, p) $\leftarrow$ BEL(a, p) or ACH(a, p)

More complex actions can be constructed from the operators TO and BY:

TO(α, p): the plan of performing α in order to make p true

BY(α, β, p): the plan of making β by doing α, while p is true

It is also needed to define some rules that connect these epistemic operators:

$$INT(a, TO(\alpha, p)) \leftarrow BEL(a, TO(\alpha, p)), INT(a, \alpha), ACH(a, p) \qquad (18)$$

These rule means that if an agent believes that doing α makes p true, if he intends to do α and if he wants p to become true, then he intends to do α in order to make p true.

There is also the corresponding rule for the relation BY:

$$INT(a, BY(\alpha, \beta, p)) \leftarrow BEL(a, BY(\alpha, \beta, p)), INT(a, \alpha), INT(a, \beta), EXP(a, p) \quad (19)$$

This rule means that if an agent believes that by doing α β is done and if he intends to do α and β and if he expects p to be true, then he intends to do α in order to do β while p is true.

It's also needed an integrity constraint, showing the inconsistency between BEL e ACH:

$$\perp \leftarrow BEL(a, p), ACH(a, p) \quad (20)$$

In fact, it's impossible to believe simultaneously that p is true and to wish p to become true. Using these rules and the Event Calculus it's possible to interpret the example presented.

In the example the user question might generate the following facts:

1. happens(e1)
2. act(e1, surface_request(u, s, infref(s, u, phone(N), hospital)))
3. initiates(e1, int(u, talk(kathy)))

In order to make inferences over this domain we need rules linking the attitudes and the domain actions:

$$INT(s, phone(p)) \leftarrow request(a, h, informref(h, s, phone\_number(n), p)) \quad (21)$$

$$BEL(s, by(phone(p), talk(a), at(p, a))) \leftarrow EXP(s, at(p, a)) \quad (22)$$

With these facts it's possible to infer that:

happens(e2), e1 < e2,

holds-at(e2, int(u, BY(phone(hospital), talk(kathy), at(hospital, kathy))))

This means that the user intention by phoning to the hospital is to talk to Kathy (assuming she is at the hospital). In order to make this inference there were abduced the following facts:

1. exp(u, at(hospital, kathy))
2. bel(user, by(phone(hospital), talk(kathy), at(hospital, kathy)))

connected with an event $e0 < e1$ and there were used rules 19, 21 and 22.

If the system knows that Kathy is at home because there was an event $e0$:

1. happens(e0), e0 < e1,

2. act(e1, BEL(s, at(home, kathy)))

then the user plan is incorrect (from the system's point of view), and it's possible for the system to infer:

1. happens(e2), e0 < e2,

2. holds-at(e2, bel(s, by(home(hospital), talk(kathy), at(home, kathy))))

using the same rules.

With this process it's possible to support dialogues with ill-formed user-plans.

5 Conclusions and Future Work

We have presented in this paper a framework that supports the abductive inference of actions and events through the use of extended logic programs and the event calculus. This framework allows the implementation of a contradiction removal mechanism that is able to remove possible contradictions from the logic program. Using this mechanism it's also possible to handle ambiguous situations in dialogues and to choose the best interpretation for the utterances.

The results obtained show that this framework is able to handle some of the problems that arise in dialogues (non-specified goals, clarification sub-dialogues, error situations) and it can support the inference of plans and attitudes in a more general natural language processing system such as the one described by Lopes and Quaresma ([7, 11]) where a multi-headed architecture coordinates several independent modules with the shared objective of supporting a robust natural language interaction.

References

[1] Douglas E. Appelt and Martha E. Pollack. Weighted abduction for plan ascription. *User Modeling and User-Adapted Interaction*, 2(1), 1992.

[2] P. Cohen, J. Morgan, and M. Pollack. *Intentions in Communication.* MIT Press, Cambridge, MA, 1990.

[3] Kave Eshghi. Abductive planning with event calculus. In *Proceedings of the International Conference on Logic Programming*, 1988.

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

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

[6] Diane J. Litman. *Plan Recognition and Discourse Analysis: An Integrated Approach for Understanding Dialogues.* PhD thesis, Dep. of Computer Science, University of Rochester, 1985.

[7] J. G. Lopes. Architecture for intentional participation of natural language interfaces in conversations. In C. Brown and G. Koch, editors, *Natural Language Understanding and Logic Programming III.* North Holland, 1991.

[8] Lode Missiaen. *Localized Abductive Planning with the Event Calculus.* PhD thesis, Univ. Leuven, 1991.

[9] Luís Moniz Pereira, José Júlio Alferes, and Joaquim Nunes Aparício. Contradiction removal semantics with explicit negation. In M. Masuch and L. Pólos, editors, *Knowledge Representation and Reasoning Under Uncertainty, Volume 808 of LNAI*, pages 91–106. Springer-Verlag, 1994.

[10] Martha E. Pollack. *Inferring Domain Plans in Question-Answering.* PhD thesis, Dep. of Computer and Information Science, University of Pennsylvania, 1986.

[11] P. Quaresma and J. G. Lopes. 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, 1992.

[12] Earl D. Sacerdoti. *A Structure for Plans and Behavior.* American Elsevier, New York, 1977.

[13] Murray P. Shanahan. Prediction is deduction but explanation is abduction. In *Proceedings of the IJCAI*, 1989.