INTENTION RECOGNITION WITH EVENT CALCULUS GRAPHS AND WEIGHT OF EVIDENCE

Fariba Sadri
Department of Computing, Imperial College London, Queens Gate, London, U.K.

Keywords: Event calculus, Ambient intelligence, Intention recognition.

Abstract: Intention recognition has significant applications in ambient intelligence, for example in assisted living and care of the elderly, in games and in crime detection. In this paper we describe an intention recognition system based on a formal logic of actions and fluents. The system, called WIREC, exploits plan libraries as well as a basic theory of actions, causality and ramifications. It also exploits profiles, contextual information, heuristics, the actor's knowledge seeking actions, and any available integrity constraints. Whenever the profile and context suggest there is a usual pattern of behaviour on the part of the actor the search for intention is focused on existing plan libraries. But, when no such information is available or if the behaviour of the actor deviates from the usual pattern, the search for intentions reverts to the basic theory of actions, in effect dynamically constructing possible partial plans corresponding to the actions executed by the actor.

1 INTRODUCTION

Intention recognition is the task of recognizing the intentions of an agent by analyzing their actions and/or analyzing the changes in the state (environment) resulting from their actions. Research on intention recognition has been going on for the last 30 years or so. Early applications include story understanding and automatic response generation, for example in Unix help facilities. Examples of early work can be found in Scmidt et al. (1978) and Kautz and Allen (1986). More recently new applications of intention recognition have attracted much interest.

These applications include assisted living and ambient intelligence (e.g. Pereira and Anh, 2009, Roy et al., 2007, Geib and Goldman, 2005), increasingly sophisticated computer games (e.g. Cheng and Thawonmas, 2003), intrusion and terrorism detection (e.g. Geib and Goldman, 2001, Jarvis et al., 2004) and more militaristic applications (e.g. Mao and Gratch, 2004 and Suzic and Svenson, 2006). These applications have brought new and exciting challenges to the field. For example assisted living applications require recognizing the intentions of residents in domestic environments in order to anticipate and assist with their needs. Applications in computer systems intrusion or terrorism detection require recognizing the intentions of the would-be-attackers in order to prevent them.

Cohen, et al. (1981) classify intention recognition as either *intended* or *keyhole*. In the former the actor wants his intentions to be identified and intentionally gives signals to be sensed by other (observing) agents. In the latter the actor either does not intend for his intentions to be identified, or does not care; he is focused on his own activities, which may provide only partial observability to other agents. Our approach is applicable to both classes, but here we describe it for the first only.

The intention recognition problem has been cast in different formalisms and methodologies. Prominent amongst these are logic-based, case-based and probabilistic approaches. Regardless of the formalism, much of the work on intention recognition is based on using pre-specified plan libraries that aim to predict the intentions and plans of the actor agent. Use of the plan libraries has obvious advantages, amongst them managing the space of possible hypotheses about the actor's intentions. But it also has a number of limitations. For example anticipating, acquiring and coding the plan library are not easy tasks, and if intention recognition relies entirely on plan libraries then it cannot deal with cases where the actor's habits are not well-known or if the actor exhibits new, unanti-

Sadri F..
INTENTION RECOGNITION WITH EVENT CALCULUS GRAPHS AND WEIGHT OF EVIDENCE.
DOI: 10.5220/0003275104700475
In *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence* (ICAART-2011), pages 470-475
ISBN: 978-989-8425-40-9

cipated behaviour.

The contributions of this paper are as follows. We propose a new logic-based approach to intention recognition based on deduction and the Event Calculus (Kowalski and Sergot, 1986) which is a formalism for reasoning about events, causality and ramifications. The system we propose is called WIREC (Weighted Intention Recognition based on Event Calculus). It exploits any available information about the actor and the context, including the actor's context-based usual behaviour, and constraints, for example his inability to perform certain tasks in certain circumstances.

WIREC takes into account the actor's physical actions as well as any knowledge-seeking actions, and reasons with what it infers about the actor's knowledge. It can exploit plan libraries if any plans correspond to the known profile of the actor, and it can revert to a basic theory of causality if no such plans are available or if the actor's behaviour deviates from his known profile. WIREC incorporates a concept of "weight-of-evidence" to focus the search for intentions and to rank the hypotheses about intentions.

Chen et al. (2008) also use the event calculus for reasoning about intentions in a framework for assisted living, but in their work they know the intention of the actor a priori, and use the event calculus to plan for the intention in order to guide the actor through the required actions. Hong (2001) shares with us concerns about the limitations of intention recognition based entirely on plan libraries. In his work he does not use plan libraries and uses a form of graph search through state changes. But his aim is to identify fully or partially achieved goals, by way of explaining executed actions rather than to predict future intentions and actions.

2 MOTIVATING EXAMPLES

Example 1: A simple example for ambient intelligence at home may be based on the following scenario. John is boiling some water. There are multiple possible intentions, beyond the immediate intention of having boiled water, for example to make a hot drink (tea or coffee), to make a meal or to use the hot water to unblock the drain. Several factors can help us narrow the space of possible hypotheses and to rank them. One set of factors involves any information about John's usual habits (John's profile) and constraints, and the current context, such as time of day, and temperature (e.g. John usually has tea during the day if it is cold, and he does not drink coffee).

Another set of factors involves "weight of evidence", which can be used if John's profile is not known, or in conjunction with his profile, or if John is behaving in a way unanticipated by his known profile. Weight of evidence can be based on what we know about what John knows based on his "knowledge seeking" actions (e.g. John has already looked in the cupboard and our RFID tag readers indicate there is no tea). It can also take into account John's other "physical" actions, and the accumulated effort towards one intention or another (e.g. John gets the pasta sauce jar, strengthening the hypothesis that he intend to make a meal, or John takes the boiled water to the sink strengthening the hypothesis that he intends to pour the water down the sink to unblock the drain).

Example 2: An example with a game flavor is as follows. Located on a grid are towns, treasures, keys to treasures, weapons, monsters, and agents. The agents can move through the grid stepping through adjoining locations, can pick up weapons, enter towns, kill monsters, and pick up treasures. They may have some prior knowledge about the locations of these various entities. Each agent has one or more intentions (goals), including killing monsters, collecting treasures or arriving at towns. The actions that the agents can perform have preconditions, for example to kill a monster, the agent must have a weapon and be co-located with the monster, and to collect a treasure the agent must have a treasure key and be co-located with the treasure. We have no prior knowledge of the "profiles" of the agents. We can guess their intentions only from their actions (and sometimes from lack of actions), for example if their progress through the grid gets them closer to a weapon or to a treasure key, if they seek to move to a grid position with a monster on it, or if despite being co-located with a weapon they do not pick it up.

3 BACKGROUND

The approach we take in this paper is based on the Event Calculus (EC). This formalism has been used for planning (Mancarella et al., 2004, for example), and has an ontology containing a set of action operators, symbolized by A, a, a1, a2, b, c, etc, a set of fluents (time-dependent properties), symbolized by P, p, p1, p2, .., q, r, neg(p), etc, and a set of time points. There are two types of fluent, primitive and ramification.

The semantics of actions are specified in terms of their preconditions and the primitive fluents they initiate and terminate. Initiation, termination and preconditions are domain-dependent rules of the form:

Initiation:

$initiates(A,P,T) \leftarrow holds(P_1,T) \wedge \ldots \wedge holds(P_n,T)$

Termination:

$terminates(A,P,T) \leftarrow holds(P_1,T) \wedge \ldots \wedge holds(P_n,T)$

Precondition: $precondition(A,P)$

The conditions $holds(P_1,T) \wedge \ldots \wedge holds(P_n,T)$, above, are called *qualifying conditions*. The first two rule schemas state that at a time when $P_1, \ldots, P_n$ hold, action A (if executed) will initiate, or terminate, respectively, fluent P. The last rule schema states that for action A to be executable fluent P must hold. An action may have any number of preconditions.

Primitive fluents hold as a result of actions:

$holds(P,T_2) \leftarrow do(A,T_1) \wedge initiates(A, P, T_1) \wedge T_2=T_1+1$

$holds(neg(P),T_2) \leftarrow do(A,T_1) \wedge terminates(A, P, T_1) \wedge T_2 = T_1+1$

Ramifications hold as a result of other fluents (primitive or ramification) holding:

$holds(Q, T) \leftarrow holds(P_1,T) \wedge \ldots \wedge holds(P_n,T)$

In the rules above all the variables are assumed universally quantified in front of each rule.

As an example of EC specification consider the following (self-explanatory) domain-dependent rules:

Example 3:

initiates(pushOnButton(Actor, radio), on(radio), T) ←holds(hasBattery(radio),T)∧ holds(neg(on(radio)),T)

terminates(pushOnButton(Actor,radio), on(radio), T) ← holds(on(radio),T)

precondition(pushOnButton(Actor,radio), co-located(Actor, radio))

holds(co-located(X,Y), T) ← holds(loc(X,L), T) ∧ holds(loc(Y,L),T).

4 INTENTION RECOGNITION: OUR APPROACH

We make the following assumptions. There are two agents, the observer (which is the WIREC system), and the actor, who is assumed to be a rational agent, and may have multiple (concurrent) intentions. We observe all the actions of the actor and in the order they take place, and the actions are successfully executed.

As well as actions, we also observe fluents. In an ambient intelligence assisted living scenario, for example, the house will have a collection of sensors, and readings from these can periodically update the representation of state kept by the system. Such observed fluents will typically be properties that can change without the intervention of the actor, for example, whether the actor is alone or has company, and whether it is a hot day.

An intention may be an action or a fluent. In the former case, the actor's actions are directed towards achieving the preconditions of the intended action, thus making the action executable. In the latter case the actor's actions are directed towards achieving the intended fluent.

4.1 Architecture of WIREC

Figures 1 and 2 illustrate the architecture of WIREC.

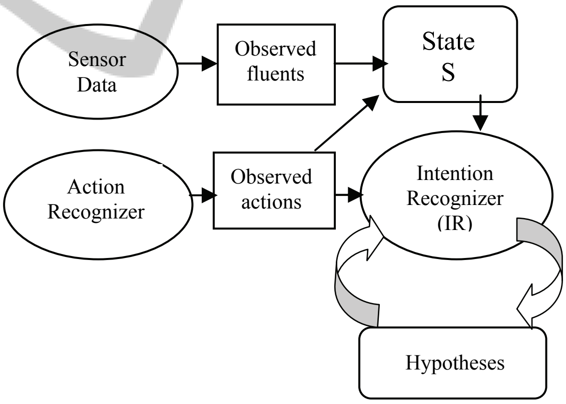

Figure 1: Architecture of WIREC.

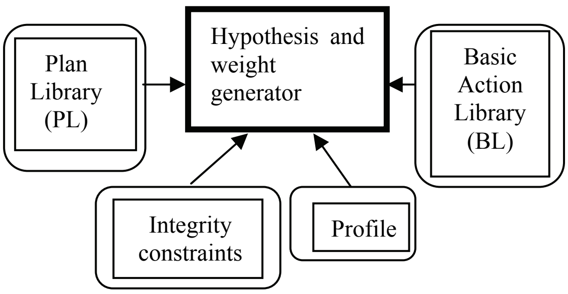

Figure 2: Architecture of the Intention Recognizer (IR) in WIREC.

The Action Recognizer can be based on some form of activity recognition (e.g. Philipose et al. 2005), and is beyond the scope of this paper. Observed fluents and actions update a database S representing the current state of the environment. The updating is done according to the semantics of actions and fluents given by EC. S contains only primitive fluents, the ramifications remaining implicit. Observed actions and state S are then used by the Intention Recognizer (IR).

IR first consults Profile to see if, in the current context, there is any information about the actor's profile identifying possible intentions and plans. If so then appropriate plans are selected from the Plan Library PL, providing an (initial) focus for the search. If not, or if the sequence of actions observed thus far does not correspond to any plans that may be selected from PL, then the search uses the Basic Action Library, BL. Both PL and BL are based on a graph representation of the Event Calculus.

4.2 Graph Representation of the Event Calculus

We adopt a graph-like representation of the Event Calculus axioms (and plans). This representation is introduced in Table 1. Each instance of a graph given in the last column is called a *graph fragment*. This graphic representation allows our intention recognition algorithm to be interpreted both in terms of reasoning and in terms of graph matching or traversal.

Plans (and thus plan libraries) can be constructed using this graph-like representation. For example Fig. 3(i) shows a plan for achieving intention *r* by doing actions *a1, a2, a3* in any order, and doing *a4* after *a1* and *a2*. Fig. 3(ii) gives a more conventional representation of the same plan used by other intention recognition systems. The approach in Fig. 3(i) compared to Fig. 3(ii) and to other approaches such as the Hierarchical Task Network models (Erol et al., 1994) has a number of advantages.

The representation in Fig. 3(i) provides information about qualifying conditions (*p1* and *p2* for the initiation of *q1*), preconditions (*q1* and *q2* for the executability of action *a4*) and ramifications (*r* holding as a result of *r1* and *r2*). All this information can be useful in intention recognition. For example if the observer knows that the actor knows that *p1* does not hold, then if the actor performs action *a1* he certainly does not intend *q1*, nor *a4*, and thus is very unlikely to intend *r*.

Also the observer may not see actions *a1* and *a2* executed, but sees *a4*. The plan makes it clear that *a1* and *a2* are needed only to establish the preconditions for the executability of *a4*. So not having observed them does not distract from the possibility of *r* being an intention. The preconditions of *a4* may have already held and the actor opportunistically executed *a4*.

Table 1: EC graph representation.

EC Axiom Name	EC Axiom schema	Graph Representation
Initiation	initiates(A,P,T) ← holds(P_1,T)∧ … ∧ holds(P_n,T)	A, P_1, ., ., P_n → P
Termination	terminates(A,P,T) ←holds(P_1,T)∧ … ∧ holds(P_n,T)	A, P_1, ., ., P_n → neg(P)
Precondition	precondition(A,P_1) precondition(A,P_2) . . precondition(A,P_n) being all the precondition axioms for A	P_1, ., ., P_n → A
Ramification	holds(Q,T) ← holds(P_1,T)∧ … ∧ holds(P_n,T)	P_1, ., ., P_n → Q

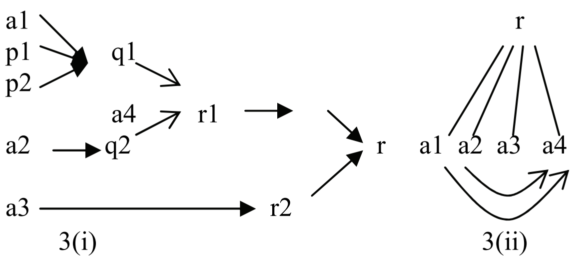

Figure 3(i): An EC plan for achieving an intention r 3(ii): A conventional representation of the plan.

4.3 Generating Hypothesis by Graph Traversal

Plan libraries (PL) in WIREC consist of "joined-up" graph fragments such as the one in fig. 3(i), and basic action libraries (BL) in WIREC consist of graph fragments such as instances of those in table 1. Whether the intention recognizer uses PL or BL, the search for hypothesis about intentions focuses on the executed actions, propagating them through graph matching (which can also be thought of as forward reasoning) and propagating the "weight of evidence”. Weight of evidence, which is a number between 0 and 1, takes into account several factors, amongst them how many actions the actor has executed so far towards an intention, and what the actor knows because of his knowledge-seeking actions.

Note that when the search uses BL, it amounts to dynamically constructing new partial plans matching the executed actions. We illustrate the algorithm by an example.

Example 4: Suppose BL consists of the fragments in table 2, where *a,b,c,d,e* are actions, and *p, p1, p2, p3, q, q1, .., q4, r, r1, t* are fluents. Fragment 2i and 2iii represent action preconditions, 2viii represents a ramification and the others represent fluent initiations.

Table 2: An Example of Part of BL.

2i	2ii	2iii	2iv	2v
p → a	a → q	q → b	c → p1	a → t
2vi	**2vii**	**2viii**	**2ix**	**2x**
b → q1 p1 ↗	b → q2 p2 ↗	q2 → r q3 ↗	d → r1 q4 ↗	e → p3 q1 ↗

Suppose we observe that action *a* has been executed. Reasoning forward from *a* amounts to traversing (some of) the paths starting at *a*. We assign weights as we do the traversal: <*q,1*> and <*t, 1*> (because of 2ii and 2v, *q* and *t* actually hold now because of *a*), <*b,1*> (2iii, action *b* is enabled - i.e. its precondition(s) now hold because of *a*), <*q1,1/2*> (2vi, action *b* is enabled by the actor but he has made no effort towards *p1* yet, so only one half of the conditions for achieving *q1* are in place), <*q2,1/2*> (2vii, similar to 2vi), <*r,1/4*> (2viii, the actor has made some effort towards *q2* but none towards *q3* yet), <*p3, 1/4*> (2x, similar to 2viii).

Notice that we ignore 2i, 2iv, 2ix; this is because we focus on the changes that are brought about by the actor. Now suppose the actor does action *c* next. This gives a weight of 1 to *p1*, and increases the weights of *q1* to 1, and *p3* to 1/2. The other weights remain the same.

Our approach has a flavour of GraphPlan (Blum and Furst, 1997), but with two significant differences. Firstly in GraphPlan in each state all actions whose preconditions are satisfied are considered. In our approach we consider only those actions whose preconditions are (fully or partially) satisfied because of the actor’s actions. Secondly GraphPlan completely constructs all states as it computes paths into possible futures. We simply partially “skim” paths into the future.

4.4 Controlling the Search for Hypotheses

We make use of several features to control the search for hypotheses:

(1) Intentions versus Consequences: An action can have several effects, some of which may be incidental and side-effects (e.g. increasing the water thermostat increases the heating bill). These we call *consequences*. Other effects may be the (immediate) intentions behind the execution of the action (e.g. having hot water) and possibly paving stones towards further actions and longer term intentions (e.g. having a shower and going to work). We use consequences to update the state S, but we ignore them in the graph traversal.

(2) Integrity Constraints: We represent and use any available information about what the actor is not capable of doing (e.g. he cannot climb ladders), as well as any constraints known about the environment (e.g. it is not possible to open the attic door or it is not possible to enter a room without being seen by a sensor). Integrity constraints can be context-dependent (e.g. John never watches TV when he has company). In example 4 if we can infer that action *e* is not possible for the actor then we will ignore fragment 2x and any other paths originating from *e*.

(3) Weight of Evidence Heuristic Threshold: We specify cut-off points, beyond which the Intention recognizer does not look further into possible futures. Currently we use a numerical Threshold, such that when the weight of a fluent/action falls below it no further reasoning (propagation) is done using it.

(4) Knowledge-Seeking Actions: Observing the actor’s knowledge-seeking actions (e.g. opening and looking inside a cupboard) gives us information about what he knows. This information is used to increase or reduce weight of evidence of the hypotheses, thus affecting further graph traversals, sometimes cutting off propagation, much as with integrity constraints.

(5) Profiles: This includes any information available (or acquired through learning) about the actor’s usual behaviour in given contexts, in terms of what his intentions may be and how he may go about achieving them. We use such information to highlight plans in PL to focus the search on. Typically PL’s plans are “connected” subsets of the

(6) fragments in BL, and thus they allow more efficient (but less exhaustive) search for hypotheses about intentions.

5 IMPLEMENTATION

We have a prototype implementation of WIREC, which has been tested by test corpora generated automatically via the planning functionalities of the Event Calculus. We have conducted empirical studies regarding the impact of factors (2), (3), and (5) from the list in sub-section 4.4. The tests have confirmed expectations regarding reduction of search. However, larger scale tests and a realistic application are needed and are part of future work.

6 CONCLUSIONS

In this paper we proposed an approach to intention recognition based on the Event Calculus. The approach has been implemented and we are currently conducting systematic testing and empirical studies in performance and scalability.

WIREC allows many further extensions and enhancements, amongst them a more sophisticated notion of weight of evidence, possibly combined with probabilities, as well as extensions to deal with scenarios involving partial observability or cognitively impaired actors, or groups of actors. Formal analysis of complexity and soundness of the approach are also subjects of current research.

REFERENCES

Blum, A. L., Furst, M. L., 1997. Fast planning through planning graph analysis, *Artificial Intelligence*, 90, 281-300.

Chen, L., Nugent, C., Mulvenna, M., Finlay, D., Hong, X., Poland, M., 2008. A logical framework for behabviour reasoning and assistance in a smart home, *International Journal of Assistive Robotics and Mechatronics*, Vol. 9 No.4, 20-34.

Cheng, D. C., Thawonmas, R., 2004. Case-based plan recognition for real-time strategy games. In Proceedings of the 5th Game-On International Conference (CGAIDE) 2004, Reading, UK, 36-40.

Cohen, P. R., Perrault, C. R., Allen, J. F., 1981. Beyond question answering. In Strategies for Natural Language Processing, W. Lehnert and M. Ringle (Eds.), Lawrence Erlbaum Associates, Hillsdale, NJ, 245-274.

Erol, K., Hendler, J., Nau, D., 1994. Semantics for hierarchical task-network planning. CS-TR-3239, University of Maryland.

Geib, C. W., Goldman, R. P., 2001. Plan recognition in intrusion detection systems. In the Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX), June.

Geib, C. W., Goldman, R. P., 2005. Partial observability and probabilistic plan/goal recognition. In Proceedings of the 2005 International Workshop on Modeling Others from Observations (MOO-2005), July.

Hong, Jun, 2001. Goal recognition through goal graph analysis, *Journal of Artificial Intelligence Research* 15, 1-30.

Jarvis, P., Lunt, T., Myers, K., 2004. Identifying terrorist activity with AI plan recognition technology. In the Sixteenth Innovative Applications of Artificial Intelligence Conference (IAAI 04), AAAI Press.

Kautz, H., Allen, J. F., 1986. Generalized plan recognition. In Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86), 32-38.

Kowalski, R. A., Sergot, M. J., 1986. A logic-based calculus of events, In *New Generation Computing*, Vol. 4, No.1, February, 67-95.

Mancarella, P., Sadri, F., Terreni, G., Toni, F., 2004. Planning partially for situated agents, 5th Workshop on Computational Logic in Multi-Agent Systems (CLIMA V), 29-30, September, J.Leite and P.Torroni, eds, 132-149.

Mao, W., Gratch, J., 2004. A utility-based approach to intention recognition. AAMAS Workshop on Agent Tracking: Modelling Other Agents from Observations.

Pereira, L. M., Anh, H. T., 2009. Elder care via intention recognition and evolution prospection, in: S. Abreu, D. Seipel (eds.), Procs. 18th International Conference on Applications of Declarative Programming and Knowledge Management (INAP'09), Évora, Portugal, November.

Philipose, M., Fishkin, K. P., Perkowitz, M., Patterson, D. J., Hahnel, D., Fox, D., Kautz, H., 2005. Inferring ADLs from interactions with objects. *IEEE Pervasive Computing*.

Roy, P., Bouchard B., Bouzouane A, Giroux S., 2007. A hybrid plan recognition model for Alzheimer's patients: interleaved-erroneous dilemma. IEEE/WIC/ ACM International Conference on Intelligent Agent Technology, 131- 137.

Scmidt, C., Sridharan, N., Goodson, J., 1978. The plan recognition problem: an intersection of psychology and artificial intelligence. *Artificial Intelligence*, Vol. 11, 45-83.

Suzić, R., Svenson, P., 2006. Capabilities-based plan recognition. In Proceedings of the 9th International Conference on Information Fusion, Italy, July.