Intention Recognition in Agents for Ambient Intelligence:

Logic-Based Approaches

Fariba Sadri

Department of Computing, Imperial College London, UK

fs@doc.ic.ac.uk

Abstract. In this paper we discuss the contribution of intention recognition in agents in ambient intelligence, and the role of logic in intention recognition. We consider the relationship between causal theories used for planning and the knowledge representation and reasoning used for intention recognition. We look at the challenges and the issues, and we explore several case studies.

Keywords. Intention recognition, plan library, logic, Event Calculus, Situation Calculus, BDI, abduction, keyhole, intended, adversarial

Introduction

An Ambient intelligence (AMI) environment is one that is engineered with embedded, unobtrusive and interconnected, digital devices to support its inhabitants. The environment is expected to be intelligent, context-aware and adaptive. Also, importantly from the point of view of this paper, the environment is expected to be capable of anticipating the needs, desires and behaviour of the inhabitants [Gaggioli, 2005, Aarts 2004], in order to support those needs and desires.

Agent technology is commonly used in AmI applications. This is not surprising, as the pervasive nature of AmI requires distributed information and problem solving, and agent architectures are particularly suited to these requirements. Agents can be used as useful abstractions in AmI systems, for example for devices and functionalities. They can also be used as paradigms for implementation, or as middleware, to co-ordinate the activities of the lower level entities. They can also be used at a higher level, to form the interface for humans.

Favela et al. [2004] and Rodriguez et al. [2004, 2005], for example, describe an architecture called SALSA, for health care, which uses agents as abstractions, to act on behalf of users, to represent services, and to provide wrapping of complex functionality to be hidden from the user. Amigone et al. [2005] describe a distributed architecture for planning in an AmI environment where devices are represented by agents that enter and leave the environment. Da Silva et al. [2007] describe an architecture for an AmI-enhanced bookshop, where there is one-to-one mapping from the elements of the bookshop (e.g. customers and audio-visual devices) to agents in a parallel world of software agents. Bosse et al. [2008] describe an agent model for

monitoring driving behaviour. They incorporate four types of interacting agents, a *sensoring* agent, to take periodic sensor readings of the steering wheel operation of the driver and his gaze, *monitoring* and *driver assessment* agents, to monitor the driving behaviour over time and to detect if driving is impaired, and a *cruise control* agent, to take charge if indeed driving is impaired, by disabling driving when it is safe to do so.

Different architectures have been used for organising agents in AmI systems. For example, Da Silva et al. [2007] describe a blackboard architecture whereby administrative agents look for messages and then contact appropriate service agents to deal with each message. LoudVoice [Busetta et al. 2004], on the other hand, is an architecture of implicit organisations emerging through role-based communication. Several other AmI proposals [e.g. Augusto 2008] use a centralised agent architecture, where a single agent receives all the information and is responsible for all the decision-making. Robocare [Cesta et al. 2005], for example, has an *event manager* agent that processes all requests and then directs each to an appropriate agent via agent-to-agent communication.

Whether the agent architecture is centralised or distributed, the agent system, in particular, if it is used as a co-ordinating middle-ware or as a higher human-interface level, should be context aware and have the capabilities of anticipating the needs and desires of the user, and of predicting their actions. This is where intention recognition can play a very important part.

Intention recognition, also called goal recognition[1], is the task of recognizing the intentions of an agent (human or otherwise) by analyzing his observed actions, the changes in the state (environment) resulting from his actions, the context and any information about (possibly learned) expected behaviours of the observed agent. Plan recognition is closely related to intention recognition, and extends it to recognizing the plan (i.e. the sequence of actions, including future actions) the observed agent is following in order to achieve his intention. Intention (and plan) recognition will enrich most AMI applications by strengthening their anticipatory capabilities.

Consider, for example a flexible AMI architecture proposed by Encanacao and Kirste [2005] and realised in an experimental system called SODAPOP (Self-Organisating Data-flow Architectures suPporting Ontology-based problem decomPosition). Similar to the LoudVoice architecture of Busetta et. al. [2004], this architecture is based of self-organisation, but it is not based entirely on roles but also on recognising the goals of the users. Instead of the more customary *function-based* interactions between the system and user, such as "turn on", "turn off", "play", etc, their system incorporates *goal-based* interactions, such as "I want to watch (the film) Chinatown now". On receiving a goal, the ensemble of devices is then expected to work out which devices and what sequence of functionalities are needed. For example, turn on TV, turn on VCR, select video, position video at start, adjust air-conditioning to a comfortable temperature, adjust sound level, adjust room lights, etc. An architecture such as SODAPOP's can be strengthened if it does not rely on explicit utterances of goals, but incorporates some form of intention recognition.

Indeed Burghardt and Kirste [2007] briefly outline a self-organization architecture of devices that incorporates a form of (probabilistic-based) intention recogniser. In their architecture, which focuses on the setting of a lecture or meeting room, the intention recogniser

[1] In this paper we use *intention* and *goal* interchangeably.

anticipates the goal of the user from sensor information. The anticipated goal is then facilitated by a plan that schedules the required devices, such as projectors and white boards.

Intention recognition has also been incorporated in the Smart Home setting of I.L.S.A. (The Independent LifeStyle Assistant) [Plocher and Kiff, 2003, Guralnik and Haigh, 2002]. I.L.S.A is a multi-agent system for monitoring the behaviour of inhabitants and alerting caregivers in cases of emergency. The agents have sensors and actuators, and they have plan libraries that they use for recognising the intention of users from their observed activities. The intention recognition uses a system called PHATT (Probabilistic Hostile Agent Task Tracker) [Geib and Goldman, 2001], which uses a plan library of simple hierarchical task network plans. One very simple one is shown below (Figure 1), indicating that doing action a, then action b and then action c achieves goal S. Observed action executions provide an execution trace, which is used to form probabilistic hypotheses about the actor's intention, and these hypothesis, in turn, generate sets of pending (i.e. future expected) actions.

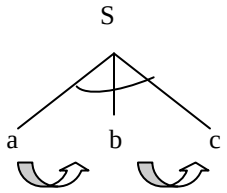


Figure1. A simple hierarchical task network

In this article we will explore the field of intention recognition. In particular we will focus on single agent cases (as opposed to multi-agents) and logic-based approaches. Logic has been a powerful tool in intention recognition since the early days [e.g. Charniak and McDermott, 1985]. For example, abduction has been a popular reasoning technique for providing hypotheses about intentions. Also, conceptually, intention recognition is directly related to planning; it is sometimes thought of as the reverse of planning. In planning, a goal is known and the reasoning is directed towards finding the actions that would achieve the goal, whereas in intention recognition the actions are known (they are observed) and the reasoning is directed towards guessing what the goal might be. Logic has been the basis of many causal theories used for planning, and it is not surprising that it finds its uses in intention recognition too.

The article is structured as follows. In Section 1 we look at the background and issues involved in intention recognition. In Section 2 we look at the relationships between logic-based causal theories and knowledge representation and reasoning for intention recognition. In Section 3 we describe and analyze several case studies. Finally, in Section 4, we conclude with a further discussion of the challenges. This article extends and updates Sadri [2011a]. In particular the Introduction and Conclusion sections have been extensively modified, seven new works have been added to the previous eight, a new section on Intention Recognition in Moral Judgments has been added, as well as more minor re-structuring, updates and extensions.

1. Background

1.1. Some Applications

Work on intention recognition has been going on for about 30 years. Examples of early work are attributed to Schmidt et al. [1978], Wilensky [1983], and Kautz and Allen [1986]. Much of

the early work has been in the context of language and story understanding and automatic response generation, for example in Unix help facilities. Other early applications include interfaces for computer-aided design [for example Goodman and Litman 1992] and collaborative problem-solving [Lesh, Rich, Sidner 1999]. However, new applications, such as assisted living and ambient intelligence, increasingly sophisticated computer games, intrusion and terrorism detection, and the military have brought new and exciting challenges to the field.

Assisted technologies, in general, and in the care of the elderly at home, in particular, are popular application areas for intention recognition. Such applications require recognizing the intentions of residents in domestic environments in order to anticipate and assist with their needs. Giroux et al. [2008] address intention recognition for the purpose of providing assistance to cognitively impaired individuals. Pereira and Anh [2009b] and Geib and Goldman [2005], focus on intention recognition in the care of the elderly. Roy et al. [2007] address complications in intention recognition when tracking the behaviour of Alzheimer patients, where it cannot always be assumed that their actions are based on an organized rational plan. So, for example, if the next action they execute is not what is expected it can be due to an interleaving of two plans for two different goals, or it can be due to error or forgetfulness.

In contrast to assisted living, applications in computer systems intrusion or terrorism detection require recognizing the intentions of the would-be-attackers in order to prevent them. Similarly, military and civil policing applications need to recognize the intentions of the enemy maneuvers and rioters' actions, respectively, in order to plan and execute counter-measures. Similar considerations may also be applicable to computer games, such as real-time strategy games. Geib and Goldman [2001] consider intention recognition in the context of computer system intrusion detection, and Jarvis et al. [2004] consider the application of terrorism detection. Mao and Gratch [2004] address intention recognition from observing military movements, and Suzić and Svenson [2006] consider the application of riot control in urban environments. In this last work, the authors hypothesise intentions from observations of movements of groups of people and contextual information, such as location (how close they are to which buildings) and resources (what weapons they have). Cheng and Thawonmas [2004] consider intention recognition in the context of strategy computer games.

Canberry and Elzer [2007] consider a rather unusual, but related, application, namely the recognition of intention behind (bar chart) graphics, in order to convey the "messages" of bar charts to sight-impaired individuals. The application involves recognizing the intention of the designer of a bar chart, by analyzing an XML representation of the chart that provides information about the heights, colours and labels of the bars.

Another application area is interactive storytelling. LOGTELL [Karlsson et al. 2007], for example, is a logic-based tool for interactive storytelling. During story creation, the user can intervene by inserting events (actions by the different characters in the story) chosen from a pre-specified list. Plan recognition is then used to find plans that incorporate these events from a plan library. The user chooses from amongst them and the story unfolds accordingly.

Another application of intention recognition, is in *intention attribution* in modeling social responsibility and morality. Mao and Gratch [2004, 2005], for example, propose a logic-based computational *attribution theory* based on intention, foreseeability and coercion, where intention and foreseeability increase responsibility and blame, and coercion decreases them. Pereira and Saptawijaya [2009] are also concerned with moral decisions based on reasoning about intentions, in particular to decide whether untoward consequences of some actions were intended by the agent that performed the actions or were merely unintended side-effects.

1.2. Classification

Cohen, Perrault, Allen [1981] classify intention recognition as either *intended* or *keyhole*. In the *intended* case the agent which is being observed wants his intentions to be identified and deliberately gives signals to be sensed by the other (observing) agent. This would apply, for example, in the case of language understanding where the speaker wants to convey his intentions. In the *keyhole* case the agent which is being observed 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 the other agent. This might be the case with help systems that provide unsolicited guidance, for example in ambient intelligence systems at home.

A third class, identified by Geib et al [2001], is *adversarial*, where the actor is hostile to his actions being observed, for example where the actions are aimed at intrusion in a network system. We can take the classification even further, to *diversionary*, where the actor is in fact attempting to conceal his intentions by performing misleading actions. Much of the work on intention recognition concentrates on the first two classes, namely *intended* and *keyhole*. Pereira and Anh [2009a] is a recent attempt to deal with diversionary intentions.

1.3. Components, Formalisms, Methodologies

Typically there are at least three components in an intention recognition system: (1) a set of intentions from which the system chooses, (2) some form of knowledge about how actions and plans achieve goals, and (3) a sequence of observed actions executed by the agent whose intention is being recognized. A possible additional component may be a set of hypotheses held about the agent's usual behaviour. For example, in a home setting, if it is late at night the recognition system may have the knowledge that the resident often prepares a hot drink and sometimes reads a book.

There is often an assumption that the actions the observed agent executes are aimed at achieving a goal, i.e. are part of the execution of a plan formed by the agent. Of course, in difficult cases, for example in the case of Alzheimer patients, or students learning a new skill, the actions may be erroneous. There is very little work that is directed at recognizing *reactive* behaviour, namely recognizing that actions may be done in reaction to external events and stimuli, and not as part of a *proactive* plan. An example is Dragoni, Giorgini and Serafini [2002] which is reviewed in Section 3.

The intention recognition problem has been cast in different formalisms and methodologies. Prominent amongst these are logic-based, case-based, and probabilistic approaches. Accordingly, component (2) of the system, i.e. the knowledge about the relationship between actions and goals, may be, for example, in the form of logic-based specifications of macro-actions [Demolombe and Fernandez 2006], or in the form of cases [for example Cox and Kerkez 2006], or in the form of plan libraries specified as Hierarchical Task Networks (HTNs) [Geib and Goldman 2005]. Geib and Steedman [2003] cast intention recognition as a problem in parsing, much as in natural language processing. Accordingly, they map Hierarchical Task Networks into context-free grammars, and the parsing is used to group together individual observations into structures that are meaningful according to the grammars. Sadri [2011b] and Hong [2001] map reasoning about intentions with logic-based theories of causality into problems of graph generation and path finding.

A common assumption is that the observer agent has (full) knowledge of the planning rules (sometimes called behaviour rules) of the acting agent. In other words, component (2) of the intention recognition system "corresponds" to the actor's knowledge about how actions and plans achieve goals. The recognition system may use the same representation of the planning

rules of the actor's, or more commonly, some transformed form of those rules which are not useful for planning, but are useful specifically for intention and plan recognition. This transformation may be, for example, in the form of HTNs augmented with probabilities and/or utilities, or the only-if direction of if-then planning rules. This will be discussed further in Section 2.

Depending on how the knowledge is cast in component (2), component (1), i.e. the set of all possible intentions at the disposal of the system to choose from, may be explicitly represented or may remain implicit. If it is explicit it will be represented as a list of possible intentions, or a set of logical facts naming each possible intention. If it remains implicit then it can be identified as, for example, the topmost operator in the HTN used in component (2), or as (ground instances of) the predicates in the conclusion of certain sets of rules used in (2).

Component (3) of an intention recognition system is the sequence of observed actions. The assumption that the acting agent's actions can be observed unambiguously is a strong one which may be justified only in virtual environments such as simulations or games. To provide the intention recognition system with this component, *activity recognition* may be used in conjunction with intention recognition.

Activity recognition, typically, uses data from cameras and RFID (Radio Frequency Identification) readers and tags to track movements of humans and to identify the objects they handle. In the home environment, for example, RFID tags may be attached to household objects. Mihailidis, Fernie, Barbenel [2001], for example, focus on recognizing very specific activities such as hand washing, similarly to Barger et al. [2002] who focus on meal preparation. PROACT [Philipose et al. 2005] employs a Dynamic Bayesian Network representing daily activities such as making tea. The user wears special gloves that can read the RFID tags of objects such as cups. Making tea is modeled as a three stage process, with high probabilities of using the kettle in the first stage and the box of tea-bags in the second stage, and medium probability of using milk, sugar or lemon in the third stage. PROACT then uses information about the objects being used and time elapsed between their usages to hypothesise possible activities.

Whatever the formalism and methodology, the result or output of intention recognition is a hypothesis, or a set of hypotheses, about the intention of the observed agent. The output of plan recognition, in addition includes hypotheses about the plan of the observed agent, including the rest of his actions yet to come. The process of generating such results may be iterative whereby hypotheses are generated, predictions are made and tested, next actions are observed, and hypotheses are refined.

1.4. Pruning the Space of Hypotheses

A substantial issue in intention recognition is the problem of narrowing down the space of possible hypotheses. Various techniques have been used to accomplish this. Early approaches impose minimality or simplicity constraints, for example via circumscription [Kautz and Allen 1986]. Here circumscription is used, in effect, to characterize the assumption that all the actions observed are being executed towards a minimum number of (top-level) intentions.

Appelt and Pollock [1992] use weighted abduction where weights are attached to conditions of the rules used for intention recognition. The cost of proving a conclusion G is the sum of the costs of proving the conditions of a rule whose conclusion matches G. The cost of proving each condition depends on whether it is assumed via abduction, or is true or is proved using other rules. The weights are intended to capture domain-specific information. For example, consider the two rules below:

$building(X, public) \wedge door\text{-}open(X)^{0.1} \rightarrow may\text{-}enter(X)$

$building(X, private) \wedge door\text{-}open(X)^{0.9} \rightarrow may\text{-}enter(X)$

They both state that one may enter a building if its door is open, but the cost of assuming that the door of the building is open is much higher (0.9) if it is a private building than if it is a public one (0.1). Put another way, intuitively, more evidence is required to believe that the door of a private building is open than to believe that the door of a public building is open.

More recently, Jarvis, Lunt and Myers [2005] augment a form of abductive reasoning with domain information regarding frequency of certain actions (for example *high* for renting a car and *low* for filing a crime report). Then in the application of terrorism intention recognition they use this information to impose a maximum on the frequency of observed actions that are to be used in the recognition system. Thus, given a set of observed actions the system focuses on a subset of these that have a maximum threshold frequency, in effect ignoring the more "common" actions and focusing on the more "rare" or "unusual" ones. Another approach is to make use of ordering constraints. For example, Avrahami-Zilberbrand and Kaminka [2005], reject hypotheses of plans which have some matching observed actions but also have actions that should have been executed earlier but have not been observed.

Associating weights with conditions of rules as in Appelt and Pollock [1992] and *a priori* frequencies to actions as in Jarvis, Lunt and Myers [2005] may be thought of as forms of probabilistic reasoning. Other more explicit forms of probabilistic reasoning to prune the search space have also been used [e.g. Geib and Goldman, 2001, 2005, Geib and Steedman, 2003]. The probabilistic approaches may be based on Bayesian reasoning or the hidden Markov model [Bui 2003]. Another approach is situation-sensitive Causal Bayes Nets [Pereira and Anh 2009b] where logic programming clauses are used to specify probabilities of intentions given information about the current state, including time of day and temperature. For example, modifying the authors' notation, the following rules

$pa\_rule(lw(T), (9,10)) \leftarrow time(T), schedule(T, football)$

$pa\_rule(lw(T), (1,10)) \leftarrow time(T), (T>23 \vee T<5)$

$pa\_rule(lw(T), (3,10)) \leftarrow temp(T, TM), TM>30$

state that the probability of (the observed agent) liking to watch TV (*lw*) is 90% when football is on, 10% when it is between 23:00 and 5:00 hours, and 30% if the temperature is higher than 30 degrees.

Mao and Gratch [2004] combine probabilities with utilities in plan recognition to choose from amongst competing hypotheses. They consider the domain of military maneuvers, and give intentions pre-specified utilities as well as probabilities. The utilities are from the point of view of the observed agent, i.e. the enemy. Thus if the observed actions lead to two equally probable hypotheses they are ranked according to their utilities, preferring the hypothesis which has higher utility, i.e. the one believed to be more profitable for the enemy. Avrahami-Zilberbrand and Kamonka [2007], on the other hand, exploit utility from the point of view of the observer, for example ordering the hypothesized intentions according to how dangerous they may be to the observing agent. In their work this is particularly useful when there is uncertainty about the observations. For example, in CCTV monitoring at an airport, someone is observed putting down their luggage and it is uncertain if they have picked it up and taken it

with them or not. The hypothesis that they have left the luggage with criminal intent is preferred as it is more dangerous from the point of view of the observers.

Once a hypothesis is chosen from amongst the possible ones, how it is used depends on the application of intention recognition. For example, two contrasting applications are identifying terrorist activity and providing assistance at home. In the first [e.g. Jarvis et al. 2004], the objective is to prevent the terrorists achieving their intentions by first identifying the intentions. In the second, for example the care of the elderly at home in an ambient intelligence setting, the objective is to help and guide the elder towards achieving his intention, by first identifying the intention. Notice that these two applications correspond to the adversarial (and possibly diversionary) and keyhole (and possibly intended) classes of problems, respectively.

To our knowledge, there is no work studying the relative effectiveness of the different approaches to intention recognition. Mayfield [2000] proposes three criteria for evaluating the effectiveness of the outcome of plan recognition systems, *applicability*, *grounding* and *completeness*. *Applicability* refers to how useful the explanation generated by the plan recognition system is to the program (or person) which is to use it in terms of its content, granularity and level of detail. *Grounding* refers to how well the plan recognition system takes account of all that is known about the actor and context (apart from actions that are observed). *Completeness* refers to how well the explanation that is produced covers all of the observations. But the three notions remain informal and anecdotal in Mayfield's work and his focus is on dialogue understanding particularly in the context of Unix help facility.

2. Logic-Based Approaches to Intention Recognition

2.1. Abductive Approaches

Abduction is a prominent methodology used in intention recognition and forms the basis of several of the papers reviewed in the case studies in Section 3. Abduction [Peirce 1958] is a form of defeasible reasoning, often used to provide explanations for observations. For example given a rule

$$\textit{room-is-hot} \leftarrow \textit{heating-is-on}$$

deduction allows deriving *room-is-hot* from the knowledge that *heating-is-on*, and abduction allows abducing *heating-is-on* to explain the observation of *room-is-hot*. In the abductive framework [e.g. Kakas, et al. for abductive logic programming], in general, given a background theory T, and an observation (or goal) Q, an abductive answer to Q is a set Δ, such that Δ consists of special pre-specified abducible atoms, $T \cup \Delta \vdash Q$ and $T \cup \Delta$ is consistent. In addition, in particular in abductive logic programming, an extra requirement may be that $T \cup \Delta$ satisfies a given set of integrity constraints. For example an integrity constraint

$$\textit{heating-is-on} \wedge \neg \textit{boiler-working} \Rightarrow \textit{false}$$

will result in disregarding *heating-is-on* as an explanation for *room-is-hot* if it is believed that the boiler is not working.

Charniak and McDermott [1985] were possibly the first to suggest that intention recognition could be framed as an abductive problem. Their focus was on intention recognition, or *motivation analysis*, as they called it, in the context of story comprehension.

They identified intention recognition as the reverse of planning. In the latter, given a task, reasoning is employed to determine what (sequence or partially ordered set of) actions would achieve it. In the former, given an action, reasoning is employed to determine what tasks it could help achieve, either directly, or in conjunction with other possible actions. This reversal of the reasoning employed for planning gives the flavour of abductive reasoning, but Charniak and McDermott's actual formalization did not strictly conform to the abductive framework, as described above. For the purpose of intention recognition plan schemas were compiled in the form *todo(G, A)* denoting that action *A* achieves goal *G*. Thus observing an instance of *A*, the same instance of *G* could be hypothesized as a possible intention.

Abduction can provide multiple hypotheses explaining an observation. Charniak and McDermott [1985] suggested a number of criteria for choosing between multiple hypotheses. One is to prefer a hypothesis that uses the most specific characteristics of the observed action. For example, if we observe that Tom picks up a newspaper, there may be two possible explanations, he intends to read it or he intends to swat a fly with it. According to the specific characteristics criteria, the first is preferred because it uses the characteristic of the newspaper as a readable object, whereas the second uses the characteristic of the newspaper just as an object. Another criterion suggested is to prefer a hypothesis which requires fewer additional assumptions (similar to the *global* criteria mentioned below). For example the explanation of swatting a fly requires an additional assumption that there is a fly, and may thus be less preferred to the explanation of reading if that requires no additional assumptions.

More recently, two broad types of criteria, *global* and *local*, are often used for ranking, and thus choosing from amongst the explanations. The global criteria may, for example, prefer explanations that are minimal in some sense, for example syntactically in terms of the number of assumed facts. The local criteria, on the other hand, may associate some form of evaluation metric with each rule in the background theory, and provide an evaluation metric for a set of hypotheses by combining the metrics of the rules from which the hypotheses originated.

In intention recognition the background theory, *T*, is a characterization of the relationships between actions and intentions, the observations, *Q*, are the actions of the observed agent, and the explanations, Δ, are hypotheses about the agent's intentions. In general, as explained in Section 1, whatever form of reasoning is employed, whether abductive, deductive, probabilistic or a mixture, intention recognition requires some knowledge of how actions achieve goals. As observed by Charniak and McDermott, such a theory is conceptually closely related to causal theories used for planning.

2.2. Causal Theories for Planning and for Intention Recognition

A common premise of intention recognition is that the observed agent is rational (even though forgetful and chaotic in some cases), and is pursuing a course of actions he believes will help him achieve a goal. This course of action must be the result of reasoning with some causal theory for planning. A further assumption is that the observer agent has some knowledge of this causal theory, although he may not use that same theory for intention recognition. The theory that he uses for intention recognition may have some direct logical relationship to the observed agent's causal theory, or may have some loose and informal relationship to it. Moreover, the theory and the representation that the observer agent uses lends itself naturally to the form of reasoning that the agent needs to employ.

Logic-based causal theories are often general purpose and can be used for different applications, such as planning and prediction. In particular there are two approaches, planning from first principles and planning from second principles or plan libraries. Both approaches

have also been used for intention recognition, but the use of plan libraries is more common. To illustrate the difference between planning from first principles and using plan libraries, let us look at the Event Calculus [Kowalski and Sergot 1986].

An abductive theory of the Event Calculus for first principle planning may include the following rules:

$holds\text{-}at(P,T) \leftarrow happens(E, T1) \wedge initiates(E,P) \wedge T1<T \wedge persists(T1,P,T)$

$persists(T1,P,T) \leftarrow not\ clipped(T1,P,T)$

$clipped(T1,P,T) \leftarrow happens(E, T2) \wedge terminates(E,P) \wedge not\ out(T2,T1,T)$

$out(T2,T1,T) \leftarrow T=T2$ $\qquad$ $out(T2,T1,T) \leftarrow T<T2$

$out(T2,T1,T) \leftarrow T2<T1$

Integrity constraint: $\qquad$ $happens(A, T) \wedge precondition(A,P) \Rightarrow holds(P, T).$

The rules state that a property *P* holds at a time *T* if an event *E* happens earlier which initiates *P* and *P* persists (at least) from the occurrence of *E* until *T*. *P* persists between two times if it is not clipped in that interval. *P* is clipped in an interval if an event *E* happens that terminates *P* and it cannot be shown that *E* occurred outside that interval. Here *not* can be thought of negation as failure. The integrity constraint states that an action can be done only if its preconditions hold.

Domain dependent information is used to specify rules defining *initiates* and *terminates*, and *preconditions* of actions, for example:

initiates(unlock-door(R), gain-entry(R))

terminates(lock-door(R), gain-entry(R))

precondition(unlock-door(R), have-key(R))

precondition(unlock-door(R), in-front-of(R)).

These state that unlocking the door to a room initiates gaining entry to that room, and has preconditions having the key to and being in front of that room, and locking the door terminates gaining entry. In this abductive framework the set of abducible predicates will consist of *happens* and the ordering relations, = and <. Thus a plan for a goal, such as *gain-entry(laboratory, T)*, will consist of a partially ordered set of actions, denoted by ground atoms in the abducible predicates.

A plan library, based on the above theory, and further information about preconditions and effects of actions, may have rules such as:

gain-entry(laboratory, T+3) ← *goto(reception, T)* ∧ *pick-up-key(laboratory, T+1)* ∧ *goto(laboratory, T+2)* ∧ *unlock-door(laboratory, T+3)*[2]

or in more simplified notation that ignores actual times and denotes sequencing with "*;*" :

gain-entry(laboratory) ← *goto(reception) ; pick-up-key(laboratory) ; goto(laboratory) ; unlock-door(laboratory).*

Thus a logic-based plan library is typically of the form:

$G \leftarrow A_1; \ldots; A_n$

where "*;*" denotes (conjunction and) sequencing. Each A_i can be a simple, atomic action or a complex action (macro-action) defined by other rules.

Much work on intention recognition assumes that the observer agent has a plan library. Very little work uses first principles causal theories (some exceptions are Quaresma and Lopes [1995] and Sadri [2011b] reviewed in Section 3). The advantage of using plan libraries for intention recognition is in restricting the search space and thus making the algorithm more scalable. On the other hand, the advantage of using first principles causal theories is that it increases the chances of recognizing the intention behind unusual and unpredicted clusters of actions, and also in recognizing short and medium term intentions as well as the ultimate intention.

The plan library used by the observer agent may consist of rules of the form:

$G \leftarrow A_1; \ldots; A_n$. R1

In such a case reasoning employed for intention recognition can be deductive, reasoning from the observed actions to the goal they establish, i.e. given instances of $A_1; \ldots; A_n$ deduce the appropriate instance of G. To make the approach more flexible this can be combined with probabilities, in the sense of increasing the probability of (an instance of) G being the intention as an increasing number of the (appropriate instances of the) actions A_i are observed. This is essentially the basis of the work of Demolombe and Fernandez [2006], described in Section 3. The reasoning may also be a combination of deductive and abductive, reasoning deductively from the occurrence of some actions $c_i; c_j; \ldots; c_k$ matching actions $A_i; A_j; \ldots; A_k$, with a most general unifier σ, to deduce a residue (resolvent)

$G\sigma \leftarrow (A_1; \ldots; A_{i-1}; A_{i+1}; \ldots\ldots; A_{j-1}; A_{j+1}; \ldots.; A_{k-1}; A_{k+1}; \ldots.; A_n)\sigma$

and then abducing (possibly a ground instance $\sigma\phi$ of) the remaining actions $(A_1, \ldots, A_{i-1}, A_{i+1}, \ldots\ldots, A_{j-1}, A_{j+1}, \ldots., A_{k-1}, A_{k+1}, \ldots., A_n)\sigma$ (and any conditions, such as the ordering), and thus hypothesizing the goal $G\sigma$ (or $G\sigma\phi$). This is essentially the approach used by Quaresma and Lopes [1995], also described in Section 3.

On the other hand, the plan library used by the observer agent may consist of rules of the form:

[2] Here we are ignoring persistence, and we are assuming that no action occurs between any two times *T* and *T+1*.

$$G \rightarrow A_1; \ldots; A_n$$

or more generally,

$$G \rightarrow (A_{11}; \ldots; A_{1n}) \vee (A_{21}; \ldots; A_{2m}) \vee \ldots\ldots \vee (A_{p1}; \ldots; A_{pq}) \qquad \text{R2}$$

which is the only-if half of a completed plan library of the earlier form R1. Then the reasoning employed is abductive, finding hypotheses G that would explain observations of actions A_{ij}. This is essentially the (logical interpretation of the) approach used by several of the studies described in Section 3, in particular Sindlar, Dastani et al. [2008], Myers [1997], Jarvis, Lunt, and Myers [2005], and Dragoni, Giorgini and Serafini [2002]. Sindlar, Dastani et al. [2008] use a meta-level representation of the form:

$$goal\ (G) \rightarrow plan(A_{11}; \ldots; A_{1n})$$

and Myers [1997], Jarvis, Lunt, and Myers [2005] employ an HTN (similar to a production rule) representation of the form:

$$G \rightarrow A_{11}; \ldots; A_{1n}.$$

In the next section we will look at a number of studies that essentially employ some form of logic-based approach for intention recognition. They cover diverse methodologies and applications. Most work described there falls in the *keyhole* or *intended* classification of intention recognition. Mulder and Voorbraak [2003] deal with enemy intention recognition, and thus conceptually fall in the *adversarial* classification.

3. Case Studies of Logic-Based Approaches

3.1. Using Simple Plan Libraries or Hierarchical Task Networks

3.1.1. Mulder and Voorbraak [2003]

This paper addresses *adversarial* intention recognition, which the authors call *tactical* intention recognition, and define as the recognition of enemy plans. In general such a task will have various identifying characteristics, for example the specialized domain of the military, the tendency of the enemy to attempt to mislead the observers, or at the very least to try and avoid detection and prevent recognition of their plans and intentions. The paper makes a contribution related to the last of these features, by way of catering for observation of actions that do not convey all the information about the actions. In other words, and in logical terms, where other work below, and in the majority of the literature, assumes fully grounded observations, such as the action *land(jet101, airbase1)*, here the observed actions may be non-ground and existentially quantified, for example $\exists X$ *land(X, airbase1)*, namely that it has been observed that something has landed at *airbase1*.

The work assumes fairly simple plan libraries, with rules of the form:

$$G \leftrightarrow A_1, \ldots., A_n,$$

where the A_i are actions, not defined by any other rules (i.e. they are simple, atomic actions and not macro-actions). The "," denotes conjunction. The reasoning for intention recognition is

abductive, with the $G \rightarrow A_1, \ldots, A_n$ direction of the rules. So given a plan library P, and set of observations O, the reasoning seeks to find abductive explanations for all observations in O, i.e. it seeks to find sets of hypotheses Δ, made up of ground atoms in the predicates that appear in the goal side (the *G*s) in the rules in P such that

$$P \cup \Delta \vdash O.$$

The following example helps illustrate the contribution:

P: *attack-plan(airbase1)* $\leftrightarrow$ *land(X, airbase1), load(X, missiles, airbase1)*

aid-plan(airbase1) $\leftrightarrow$ *land(X, airbase1), load(X, aid-supplies, airbase1), cover-with-red-cross(X, airbase1)*

The first rule in *P* specifies that an attack is planned from *airbase1* if something lands at *airbase1* and is loaded with *missiles*. The second rule specifies that an aid flight is planned if something lands at *airbase1* and is loaded with *aid-supplies*, and the carrier is covered with the Red Cross symbol.

Now an observation *load(jet1, missiles, airbase1)* leads to one hypothesis, namely *attack-plan(airbase1)*. However, an observation $\exists Z$ *load(jet1, Z, airbase1)* leads to two potential hypotheses, namely *attack-plan(airbase1)* and *aid-plan(airbase1)*. A further observation $\exists X$ *cover-with-red-cross(X, airbase1)* still maintains two hypotheses, one consisting of *aid-plan(airbase1)* explaining both observations, and the other consisting of both *attack-plan(airbase1)* and *aid-plan(airbase1)*, each explaining one of the observations.

No proof procedures are suggested for the abduction, but it is worth noting that most, if not all, proof procedures for abductive logic programming will capture this reasoning (for example, the iff-proof procedure of Fung and Kowalski [1997] and the CIFF proof procedure of Mancarella et al. [2009]). Note that with a slightly more elaborate representation of plan libraries it may well be possible to avoid reasoning with existentially quantified representations of observations. One such formalization could make use of binary representations of the known information about the observed actions, such as, for example for an event $e_1$of landing:

act(e_1, *land*) *destination*(e_1, *airbase1*),

ignoring what is not known, here the aircraft involved in e_1, and for an event e_2 of loading:

act(e_2, *load*) *base*(e_2, *airbase1*) *carrier*(e_2, *jet1*),

ignoring what is not known, here the cargo that is loaded. To accommodate such a representation the *attack-plan* rule, above, for example, can be written as:

attack-plan(airbase1) $\leftrightarrow$ *act(E1, land), destination(E1, airbase1), carrier(E1, X), act(E2, load), base(E2, airbase1), cargo(E2, missiles), carrier(E2, X),*

ignoring any temporal constraints and persistence requirements between events *E1* and *E2*.

3.1.2. K. Myers [1997]

Myers' [1997] work is motivated by making planning technology more accessible and easier to use. Here the two parties, the observer and the actor are, respectively, a planning system and its user. The objective is to allow planning to be done co-operatively, where, as well as (optionally) inputting a goal, the user can participate in the planning process by providing the planning system with a partial plan which may consist of a (partial) list of tasks (actions or subgoals). The system then attempts to identify from this partial plan any higher level goals the user may have, and then to complete the partial plan to achieve these higher level goals.

For example let us consider the domain of travel planning. A traveler may provide a partial list of tasks, for example visiting the Swiss embassy and visiting a ski shop. A co-operative planner, in principle, may fill in the gaps, by deducing or guessing (abducing) that the user wishes to take a ski holiday trip to Switzerland, and can then complete the plan by including the actions of booking the flight and booking accommodation in Switzerland.

The set of possible top level goals is pre-specified. The planner uses plan libraries, and in fact the same libraries, both for planning and for intention recognition. The plan libraries are based on the Hierarchical Task Network (HTN) model of planning [Erol, Hendler, Nau, 1994]. HTN defines operators for reducing goals to subgoals. For example:

O1: $B \Rightarrow C, D$

O2: $C \Rightarrow K, L, M$

O3: $C \Rightarrow P, Z$

O4: $D \Rightarrow W$

We comment later, at the end of this section, on the logic of HTNs and the above representation. In general HTN planning is based on hierarchical decomposition of tasks into subtasks. Tasks can be primitive, in which case they are directly executable and have no decomposition, such as *W*, above, or they are non-primitive. Non-primitive tasks can be decomposed into subtasks in one or more ways. For example *O2* and *O3*, above, show two possible decompositions of task *C*. *O2*, for example, is an operator that reduces goal *C* to subgoals *K*, *L*, and *M*. Such decompositions, together with variable bindings and ordering information, are called task networks. Planning starts with an initial task which is repeatedly decomposed until the resulting tasks are all primitive.

In Myers' work, given the HTN above, for example if the user provides a goal *C* two plans are possible, one consisting of subgoals *K, L, M*, and the other of *P, Z*. On the other hand, if instead of providing a goal the user provides a partial plan consisting of *P*, the intention recognition system guesses that the intention is *C*, and ultimately B, and provides a plan *P, Z, D*, further refined to *P, Z, W*, compatibly with the user-given partial plan.

The planning is done conventionally, using the operators to reduce goals to subgoals. On the other hand, the determination of the user goals from the actions or subgoals they input is based on a form of abduction. For example, in the case above, *B* is an abductive explanation, according to Myers, for the user input subgoal *P*. This is determined using the notion of an *abductive chain* which in this case is:

$P \Rightarrow^{O3} C \Rightarrow^{O1} B.$

It is assumed that the operator definitions are non-recursive. If there are alternative top level goals possible via such abductive chains then a subset is chosen. The choice is not addressed in the paper. Once a set of goals is identified then the planning proceeds as in HTN with the constraint that the final plan should accommodate the user-given tasks. A brief analysis of the approach provides an exponential upper-bound for the process of constructing the abductive chains, but argues that this is dependent on the length of the chain and in practice this length is small, while the user-given partial plans can reduce the search space of the planning phase.

Note that the use of abduction here calls for a logical interpretation of the HTN representation. The operator decompositions are similar to goal-reduction rules in production systems. Kowalski and Sadri [2009] argue the difficulties of giving such rules model-theoretic semantics and provide an alternative framework that does provide such semantics. However, it seems relatively straightforward to put Myers' work in the context of the discussion of the different logic-based approaches in Section 2, and to relate the abduction done here to the formal notion of abduction defined there. This can be done by representing the operators in a logically meaningful way. For example, if we interpret operator definitions O1-O4, above as goal-reduction rules for goals *B*, *C* and *D*, we have:

$B \leftrightarrow C \wedge D$

$C \leftrightarrow (K \wedge L \wedge M) \vee (P \wedge Z)$

$D \leftrightarrow W$.

Now the abductive reasoning required in the construction of the *abductive chains* is classical abduction using the rules above in the only-if direction ($\rightarrow$), as in Mulder and Voorbraak [2003], above and Sindlar et al. [2008], described below. The planning can also be seen as classical logic-based planning using the *if* direction ($\leftarrow$) of the rules.

3.1.3. Jarvis, Lunt, and Myers [2004, 2005]

This work essentially uses the approach of Myers [1997] in the application of terrorist intention recognition. This more recent work uses an architecture called CAPRe (Computer-Aided Plan Recognition), where the plan libraries are in the form of templates, and they contain more information than the earlier work [Myers 1997]. They are non-ground (i.e. contain non-ground parameters) and have additional information, such as ordering and preconditions of tasks, and *frequency* and *accuracy* of observations, described later.

An example, in a slightly simplified notation, is:

template Physical_Attack(?group, ?target)

purpose destroy(?group, ?target)

tasks 1. reconnaissance(?group, ?target)

2. prepare_attack(?group, ?target)

3. attack(?group, ?target);

Ordering 1-->3, 2-->3;

The template specifies a macro-action *Physical_Attack* of a target by a group. The purpose (effect) of the action is the destruction of the target by the group. The action is decomposed into three partially ordered actions, reconnaissance of the target and preparation of the attack, followed by the attack.

Ignoring the ordering and the variables this template corresponds to the HTN representation:

$destroy \Rightarrow Physical\_Attack$

$Physical\_Attack \Rightarrow reconnaissance, prepare\_attack, attack$

in the earlier notation (Subsection 3.1.2). A more elaborate abstract template example, ignoring variables, may be:

template TaskName

purpose P

tasks $T_1, \ldots, T_n$

Ordering set of T_i *-->* T_j

Effects

E_1 *at* T_1*, frequency* F_1

....

E_n *at* T_n*, frequency* F_n*;*

The task *TaskName*, itself may be a subtask of another task. Ignoring the ordering and the *frequency* information the template above corresponds to the HTN representation:

$P \Rightarrow TaskName$

$TaskName \Rightarrow E_1, \ldots, E_n$

$E_1 \Rightarrow T_1$

....

$E_n \Rightarrow T_n$.

The *frequency* information represents the frequency of a given action occurring in *normal* circumstances. It is given values *high*, *medium* or *low*. For example car renting has a *high* frequency, whereas filing reports of missing persons has a *low* frequency. The templates may also include information about *accuracy* of normal observation, again given as *high*, *medium* or *low*. For example observing a missing person report has higher accuracy than recollecting a license plate. In effect *Frequency* and *accuracy* are used as a means of introducing stochastic measures in reasoning about choices.

The intention recognition is done in two phases. The first is mostly identical to the approach of Myers [1997], with so-called *task seedlings* generated from the templates in a way that is the similar to the generation of *abductive chains* from Hierarchical Task Networks. The variables are treated by unification, along the chain. The templates allow constructing chains starting from observation of actions T_i or effects E_i. Any additional *frequency* information is used to specify maximum frequencies of observations for which abductive explanations are sought, thus, for example allowing the system to ignore commonly occurring (supposedly mundane) actions. Similarly the *accuracy* information can also be used to impose a threshold on the observations that are taken into account.

The result of the first phase of intention recognition consists of explanations for each chosen observation. The second phase attempts to combine these to provide compatible explanations for clusters of observations. It does so by considering sets of increasing size of the explanations generated by the first phase, and removing from consideration incompatible sets. A set is incompatible if it has incompatible variable bindings, or incompatible orderings or other constraints within the templates.

As in the case of Myers [1997] the performance of the system is dependent on the length and number of the *task seedlings* (*abductive chains*). It is also dependent on the number of sets of explanations that have to be considered in the second phase. As might be expected, experimental results report degradation in performance with increasing number of observations to be explained and increasing noise, i.e. activities that are unrelated to any attack plan.

It can be noted that the templates can be formalized in logic along the lines sketched in Subsection 3.1.2. For example the *Physical_Attack* template can be formalized as:

$$destroy(Group, Target, Time) \leftrightarrow physical\text{-}attack(Group, Target, Time)$$

$$physical\text{-}attack(Group, Target, Time) \leftrightarrow reconnaissance(Group, Target, Time1), prepare\text{-}attack(Group, Target, Time2), attack(Group, Target, Time), Time1<Time, Time2<Time$$

It would be interesting to explore if such a logic-based formalization, together with more recent abductive proof procedures with constraint handling, such as Mancarella et al. [2009] would have any impact on the performance of the system. Such an abductive proof procedure allows merging into one process the two phases of generation of explanations for each action and then finding compatible clusters. This merging may prevent generation of hypotheses and clusters that would in the end prove incompatible.

3.2. BDI-Based Approaches

Sindlar, Dastani et al. [2008]

In this work the acting agent is assumed to be a BDI-type agent [Rao and Georgeff 1995], with the particular feature of interest being the planning rules that govern its behaviour. These behaviour rules are assumed to be of the form

$$G \leftarrow B \mid \pi$$

to be interpreted in logical terms as stating that goal G holds if B is believed and plan π is executed.

A plan may consist of sequences of actions and non-deterministic choice. Actions include test actions, that do not bring about any changes but simply check a condition. For example the planning rule below:

$g \leftarrow b \mid a1; a2; (\varnothing 1?; ((\varnothing 2?; (a3; a4) + (\neg \varnothing 2?; a5))) + (\neg \varnothing 1?)); a6; a7$

specifies that *g* is achieved if *b* is believed and actions *a1* and *a2* are executed and, if $\varnothing 1$ holds then if $\varnothing 2$ holds *a3* and *a4* are executed, otherwise *a5* is executed, and then *a6* and *a7* are executed.

The observer agent may also be a BDI-type agent with a belief base, goals and behaviour rules. Those details are not relevant for the purposes of intention recognition. What is relevant and important here about the observer agent is that, in addition to any rules that govern its own (planning and action execution) behaviour, it has knowledge of the planning rules of the observed agent. This knowledge is in the form of rules of the form

$goal(G) \wedge belief(B) \rightarrow plan(\pi)$ RD

The agent is assumed to have one such rule for every rule of the form

$G \leftarrow B \mid \pi$

in the observed agent's knowledge base. Rules of the form RD are not used for planning purposes, and are not particularly appropriate for such purposes, but they are used specifically for intention recognition. They are used abductively to provide explanations for observed actions in terms of the goals and beliefs of the observed agent. The abducible predicates are *goal* and *belief*. Thus although the observer agent is a BDI-type agent, the rules it uses for intention recognition have logical semantics, and they lend themselves to abductive reasoning.

The work is based on several assumptions: One is that the observed agents execute one plan at a time, so there are no interleaved plan executions. Another is that the observer is able to determine ahead of time whether they have *complete* observability or whether their observations will be *late* or *partial*. *Complete* observation means that all the non-test actions of the actor will be observed. *Late* observation means that the observer may miss some of the initial actions of the plan executed but from the first observation it will observe all the non-test actions executed henceforth. *Partial* observation means that the observer may miss some of the actions, anywhere in the plan. In all three cases all the actions that are observed are assumed to be observed in the correct sequence.

In each of the three cases a match is sought between the sequence, S, of actions that have been observed and the sequence in π in the conclusion of rules of the form RD. The test actions in π are ignored in this matching process. In the case of *complete* observability the match sought is between S and any initial segments of π (for example *a1;a2* or *a1;a2;a3;a4* or any other initial segment of *a1;a2; a3;a4;a6;a7* or *a1;a2; a5;a6;a7* or *a1;a2; a6;a7* in the above example). In the case of *late* observation the match sought is between S and any sub-segments of π (for example a1;a2 or a2;a3;a4 or a2;a5 or …). In the case of *partial* observation the match sought is between S and any *dilution* of any sub-segment of π, i.e. any sub-segment with any elements removed (for example a2;a4 or a1;a4;a7 or …).

When such a match is found the conditions $goal(G) \wedge belief(B)$ of the RD rule provide the necessary abductions and thus the hypothesis regarding the intention of the acting agent.[3] We can interpret what is done here in the general abductive framework outlined in Section 2 by considering the theory used for abduction to consist of the rules of the form

$$goal(G) \wedge belief(B) \rightarrow (a_1;a_2;\ldots;a_n) \vee \ldots \vee (a'_1; a'_2; \ldots; a'_m)$$

where the a_i and a'_i are non-test actions, and ";" can be thought of, as is common, as representing conjunctions of actions and their sequential ordering. The three cases of observability can be thought of as providing additional integrity constraints. *Complete* and *late* observability both impose the constraint that there is no action in between any two actions observed where one immediately follows the other. *Complete* observability additionally imposes the constraint that there is no action before the first one observed. *Partial* observability imposes neither constraint.

Two quite intuitive propositions are proved. One states that the number of possible explanations (abductive hypotheses) generated in case of *complete* observability is less or the same as that generated by *late* observation which in turn is less or the same as *partial* observation. The other is that in each case the number of possible explanations decreases or at most stays the same as the number of observations increases.

Sindlar, Dastani, et al, [2010]

The authors extend their earlier work for application to role-playing games. The game is viewed as an agent society with (artificial, BDI-based) agents playing pre-specified roles. A role is characterized by a name, a set of goals, with a partial ordering, denoting priorities, and a set of behavior rules. For example the role *thief* may have a behavior rule for possessing an object by stealing it. It may have a goal of possessing an object and a higher priority goal of staying undetected.

The work assumes that every agent in the game behaves according to its role, and does not interleave plans, even if it has multiple goals. As with the earlier work, observing agents use rules such as RD, above, corresponding to the behaviour rules of the agents to recognize their goals and predict their next actions. The contribution of this more recent work is in providing various ranking criteria for the hypotheses that the goal recognition process generates.

One such ranking, for example is based on *strict role conformance*. Thus a hypothesis is ranked higher than another if the two differ only in the order of goals, and the first conforms to the priority ordering specified in the role of the observed agent. Another ranking is based on *norm conformance*. The game, viewed as an agent society, may have specified norms, in terms of prohibited and obliged actions. The hypotheses can be ranked with respect to the norm compliance of the actions they predict. For example a hypothesis that includes some obliged actions and no forbidden actions is ranked higher than one that includes some forbidden actions and no obliged ones.

[3] In the paper both the planning rules of the observed agent and rules of the form RD of the observer agent are assumed to be ground.

3.3. Using the Situation Calculus

3.3.1. Demolombe and Fernandez [2006]

This paper proposes a framework that combines probabilities with a Situation Calculus-like formalization of actions. The Situation Calculus [Reiter, 2001] and its further extension GOLOG [Levesque et al, 1997] are logical formalisms for specifying actions and their effects. They allow specifying macro-actions or, in the authors' terminology, procedures. For example, the procedure for dealing with a fire on board a plane may be represented in GOLOG as:

tackle-fire-on-board $=^{def}$ *turn-fuel-off; turn-full-throttle; turn-mixture-off*

to express the sequence of actions *turn fuel off*, *turn full throttle* and *turn mixture off*. Note the similarity between this notation and the HTN concept and representation.

Each single atomic action maps one state to its successor, and correspondingly macro-actions map one state to another. The assumption of the paper is that the actor and the observer, essentially, have the same "knowledge" about such procedures. However, a central aim of the paper is to allow intention recognition of human actors who may interleave procedures. To this end for each procedure that is in the knowledge base of the actor a modified one is included in the knowledge base of the observer. The modification is done by explicitly adding arbitrary actions, and any constraints, to the procedure definition. For example the definition above is modified to the following:

tackle-fire-on-board $=^{def}$ *turn-fuel-off;* $(\sigma$*/turn-fuel-on);* *turn-full-throttle;* $(\sigma$*/turn-fuel-on); turn-mixture-off.* OD

This modified definition represents the same procedure for tackling fire on board, but explicitly allows any sequence of actions σ, except *turn-fuel-on* in between turning the fuel off and turning full throttle, and between turning full throttle and turning the mixture off. Here, in the *tackle-fire-on-board* procedure, the three actions *turn-fuel-off, turn-full-throttle, turn-mixture-off* are *explicit* actions, and the action *turn-fuel-on* is said to be *prohibited* in between any two of the *explicit* actions. The other actions in σ are said to be *tolerated*. The set of all defined procedures provides the set of all possible intentions.

Those familiar with formalisms such as the Situation Calculus or the Event Calculus [Kowalski and Sergot, 1986] may relate the observer definition OD to a formalization that makes explicit the required persistence of properties between actions or states. An example of such formalization, letting the S_i to represent states or time points, is:

happen(tackle-fire-on-board, S_3*)* $\leftarrow$ *happen(turn-fuel-off,* S_1*),*
happen(turn-full-throttle, S_2*), happen(turn-mixture-off,* S_3*),* $S_1 < S_2$, $S_2 < S_3$,
persists(S_1*, fuel-off, ,* S_3*)*

persists(S_1*, fuel-off, ,* S_3*)*$\leftarrow \neg$*(happens(turn-fuel-on, S),* $S_1 < S$, $S < S_3$*),*[4]

[4] Alternatively to conform to the notation we introduced in Section 2 we could write:
persists(T1,P,T) $\leftarrow$ *not clipped(T1,P,T)*

where action *turn-fuel-off* is specified to have the effect *fuel-off*.

The paper assumes that there is full visibility of actions performed, or rather that the intention recognition is performed purely on the basis of the observed actions. In the initial state before any action is observed all possible intentions are given pre-specified probabilities. Then as actions are observed they are matched with the procedure definitions. The matching includes *explicit* actions in procedure definitions, such as *turn-fuel-off* in the example above, as well as any *prohibited* or *tolerated* actions.

With each match probabilities are updated for the possible intentions. The probability of an intention is increased, if the next expected *explicit* action is observed, it is decreased if a *tolerated* action is observed, and decreased to a greater extent if a *prohibited* action is observed. All the probability increments and decrements are by pre-specified amounts. The computation cost of evaluating the probabilities is said to be linear with respect to the number of observations for a given procedure.

The following is a modified version of an example in the paper. Consider three procedures

P1 = a; (σ/g); b; c

P2 = d; σ; e

P3 = a; σ; f

where, σ denotes a sequence of arbitrary actions. If in the initial state *s0* action *f* is observed then the probabilities of the agent having the intention *P1*, *P2* or *P3* in state *s0* are equal and low. So *P(Int(P1, s0))=P(Int(P2, s0))= P(Int(P3, s0))*, where *P(Int(Q,S))* denotes the probability of intention *Q* in state *S*. Then in the resulting state, *s1=do([f],s0)* in modified Situation Calculus notation, if action *a* is observed, the probabilities of *P1* and *P3* are increased equally in state *s1*. Now if an action *m* is observed in state *s2=do([f,a],s0)* there is still a match with *P1* and *P3*, but action *m* lowers the probability of both *P1* and P3, thus, for example *P(Int(P1, s0))< P(Int(P1, s1))* and *P(Int(P1, s2))< P(Int(P1, s1))*. If now an action *g* is observed in state *s3*, where *s3= do([f,a, m],s0)*, it reduces the probability of *P1*, because *g* is a prohibited action for *P1* in sate s3. But the observation of *g* does not affect the probability of *P3*, thus *P(Int(P1, s3))< P(Int(P3, s3))*.

3.3.2. Baier [2002]

Earlier work by Baier [2002] proposed using CONGOLOG for procedure recognition. CONGOLOG [Giacomo et al. 2000] is a successor of GOLOG, and extends GOLOG by allowing a form of concurrency, that is, in effect, partial ordering between actions. Like GOLOG, CONGOLOG allows complex actions such as sequences ($\sigma1;\sigma2$), non-deterministic choice ($\sigma1|\sigma2$, to mean do either $\sigma1$ or $\sigma2$), non-deterministic iteration (σ*, to mean do σ an arbitrary number of times), and conditional (*if* ϕ *then* $\sigma1$ *otherwise* $\sigma2$), and while loops (*while* ϕ *do* σ). CONGOLOG also allows concurrency, or more precisely, partial ordering, whereby

---

clipped(T1,P,T) ←happens(E, T2) ∧ terminates(E,P) ∧ not out(T2,T1,T)
terminates(turn-fuel-on, ,fuel-off).

$\sigma 1||\sigma 2$ means do $\sigma 1$ and $\sigma 2$ in whatever order. In CONGOLOG a program, commonly called a procedure, is a complex action.

The objectives of Baier are similar to Demolombe and Fernandez [2006], in recognizing what procedure an actor is performing, assuming full observability of actions, and given the possibility that the actor interleaves execution of different procedures. Baier's work differs from Demolombe and Fernandez in two main respects. He does not use probabilistic reasoning, and his procedures can be more complex, because he uses the more complex syntax of CONGOLOG. For example the following is a procedure for the goal of landing an aircraft, by extending the flaps 3 times, then reducing the thrust produced in the engines to below 100, and finally extending the landing gear:

extend-Flaps; *extend-Flaps*; *extend-Flaps*;

while thrust-Level(n) $\wedge$ $n \geq 100$ *do decrease -Thrust endWhile; extend-Gear.*

Baire's procedure recognition algorithm, allows recognition of a procedure even though extra actions have been interleaved with the procedure's actions, but the algorithm is exponential. To determine if in state *s* a procedure *p* is being executed it requires:

1. a search through past actions to determine if there is a state *s'* earlier than *s* when the execution of *p* started, and

2. determination of whether between *s'* and *s* any actions have rendered it impossible for the remainder of the actions of *p* to achieve the goal of *p*. This is somewhat similar to detecting *prohibited* actions in the terminology of Demolombe and Fernandez.

Baire provides some suggestions for a more efficient implementation.

3.3.3. Goultiaeva, A. and Lesperance, Y. [2007]

This work is similar to Baier [2002] in using CONGOLOG and no probabilities, and it is similar to Demolombe and Fernandez [2006] in using analogous concepts to their *prohibited* and *tolerated* actions. As with both approaches, the work relies on plan libraries, here specified in the CONGOLOG language, including nested procedure definitions. The plans in the plan library are specified for use in intention recognition and not planning, and they are specified, as in Demolombe and Fernandez, in such a way as to allow intention recognition in the presence of interleaved plan executions and arbitrary and unnecessary actions.

Recall how *prohibited* and *tolerated* actions are used in the work of Demolombe and Fernandez, as exemplified by the procedure:

$$tackle\text{-}fire\text{-}on\text{-}board =^{def} turn\text{-}fuel\text{-}off;\ (\sigma/turn\text{-}fuel\text{-}on);\ turn\text{-}full\text{-}throttle;\ (\sigma/turn\text{-}fuel\text{-}on);\ turn\text{-}mixture\text{-}off.$$

where σ is a set of *tolerated* actions and *turn-fuel-on* is a *prohibited* action.

Goultiaeva, and Lesperance take advantage of the more complex syntax of CONGOLOG to define *prohibited* and *tolerated* actions as (extended) CONGOLOG complex actions. For this purpose they define action operators *anybut* and *any*, respectively. The operator *anybut* is defined as follows:

$$anyBut([a1, ..., an]) =^{def} \pi a.(if(a \neq a1 \wedge ... \wedge a \neq an)\ then\ a\ else\ false?\ endIf)$$

where $\pi a.\delta$ is a CONGOLOG complex action that indicates non-deterministic choice of argument. It non-deterministically chooses an instantiation of argument *a* and then performs that instantiation of program δ. Furthermore, *false?* is a test action that fails. The operator *any* is defined as follows:

any $=^{def}$ *anyBut([]).*

They also provide an extended notion of *prohibited* actions, namely *prohibited* procedures, via a third operator, *minus*(δ,δ'), where both δ and δ' are complex actions. The complex action *minus*(δ,δ') matches any execution that would match δ, as long as it does not match the procedure δ' .

Thus, for example, to specify a procedure of doing action *a* followed by *b*, but intermingled by any actions in between the two, they provide the notation: *a; any*; b*[5]. If the intermingled actions can be anything but *c* and *d*, they have: *a; anybut[c,d]*; b.* Finally if a procedure *p* is defined as, say, *minus([a; any*; b], p')*, then *p* will be a recognised intention if the complex action *a; any*; b* is executed, provided no part of this execution corresponds to the procedure *p'* (defined elsewhere).

Using this notation the *tackle-fire-on-board* procedure is specified as follows:

tackle-fire-on-board $=^{def}$
minus([turn-fuel-off; any; turn-full-throttle; any*; turn-mixture-off],*
[(anyBut([turn-fuel-on])); turn-fuel-on]).*

Now the reasoning is much as in Demolombe and Fernandez, but without probabilities and ranking. Observed actions are matched with procedures, under the assumption that the actions are fully observed and the initial state is fully known. Intention recognition is done incrementally, and as more actions are observed the set of possible hypothesized intentions is pruned by observations of *prohibited* actions and procedures.

3.4. Using the Event Calculus

3.4.1. Quaresma and Lopes [1995]

This work uses an abductive version of the Event Calculus [Kowalski and Sergot 1986] and a modified version of the action language of Gelfond and Lifschitz [1992] for recognizing the intention behind speech acts. As with Dragoni et al., described later, it focuses on the *request* and *inform* speech acts. This work differs from the majority of the papers on intention recognition in that it uses a theory of planning from first principles as the background theory for intention recognition. This background theory would presumably be used by the speaker, *s*, to plan for its goals by reducing goals to subgoals, and it is used by the hearer, *h*, with a mixture of deductive and abductive reasoning, for intention recognition.

The background theory is a modification of the Event Calculus which is a causal theory formalized in Horn clause logic augmented with negation as failure. The modified formalization is complex and requires a large number of axioms. Here we give only a flavor. The core Event Calculus rules used by the authors are:

holds-at(P,T) $\leftarrow$ *happens(E), initiates(E,P), succeeds(E), E<T, persists(E,P,T)*

[5] Recall * denotes non-deterministic iteration in CONGOLOG.

persists(E,P,T) ← *not clipped(E,P,T)*

clipped(E,P,T) ←*happens(C), terminates(C,P), succeeds(C), not out(C,E,T)*

out(C,E,T) ← *T=C* *out(C,E,T)* ← *T<C*

out(C,E,T) ← *C<E.*

These have minor differences with those we presented in Section 2. They state that a property *P* holds at time *T* if an earlier event *E* happens which initiates *P* and *E* succeeds and *P* persists (at least) from the occurrence of *E* until *T*. *P* persists between two times, (the time of) *E* and *T*, if it is not clipped in that interval. It is clipped in that interval if an event *C* happens and succeeds, *C* terminates *P*, and it cannot be shown that *C* occurred outside that interval[6]. Here *not* is negation as failure.

An event *E* succeeds if its preconditions hold at the time of its occurrence, i.e.:

succeeds(E) ← *act(E, A), hold-at(P_1, E), ….., holds-at(P_n, E)*

where $P_1, \ldots, P_n$ are the preconditions of the action operator *A* of event *E*. An event *E* initiates a property *P* depending on its action operator and any qualifying conditions, i.e.:

initiates(E, P) ← *act(E, A), hold-at(P_1, E), ….., holds-at(P_n, E).*[7]

These last two rules are the authors' translation into the Event Calculus of the formalisation of actions

A causes P if $P_1, \ldots, P_n$

in action language of Gelfond and Lifschitz.

These core rules are augmented by the following rules that enable abductions:

F ← *not* ¬*F* R1

¬*F* ← *not F* R2

where ¬ is classical negation, and *F* is an abducible predicate. These rules model abductive reasoning and state that if it is not possible to prove ¬*F* then *F* should hold, and vice versa. The formalization adapts the Well Founded Semantics of Extended Logic Programs [Pereira et al. 1992]. In the intention recognition framework the abducible predicates are *happens/1*, *act/2* and *</2*.

[6] Events, *E*, are made up of an operator and a time of occurrence, and *E<T* is shorthand for saying the time of occurrence of *E* is before *T, and C<E* is shorthand for saying event *C* occurs before event *E*.

[7] The authors do not distinguish between precondition and qualifying conditions.

Epistemic operators are used to describe the agents' mental state, for example:

int(a, actn)	to specify that agent *a* wants action *actn* to be done
bel(a, p)	to specify that agent *a* believes that *p* is true
ach(a, p)	to specify that agent *a* believes *p* will be achieved as a consequence of the actions of some agent (itself or others).

Event-Calculus-type rules describe the relationships between the components of the mental state (as ramifications). An example is the rule:

$$holds\_at(int(A,to(\alpha,P)),\ T) \leftarrow holds\_at(bel(A,\ to(\alpha,P),\ T),\ holds\_at(int(A,\ \alpha),\ T),\ holds\_at(ach(A,P),\ T) \quad \text{R3}$$

where $to(\alpha,P)$ is a term representing the plan of performing α to make P true. The rule specifies that if an agent A believes that by doing α P will become true and A intends to do α, and A believes P will be achieved, then A intends to do α in order to make P true. Furthermore, such rules are augmented by an integrity constraint:

$$holds\_at(bel(A,P),\ T),\ holds\_at(ach(A,P),\ T) \Rightarrow false$$

stating that at no time does the agent believe both that P is true and P will be achieved.

Speech acts are specified as actions within the Event Calculus framework, for example:

$$succeeds(E) \leftarrow act(E,\ inform(S,H,P)),\ hold\text{-}at(bel(S,P),E),\ holds\text{-}at(bel(S,\ int(S,\ inform(S,H,P))),E)$$

$$initiates(E,\ bel(H,\ bel(S,P))) \leftarrow act(E,\ inform(S,H,P))$$

state that a speech act event E of S informing H some information P succeeds if S believes P and believes that he intends to inform H about it. Furthermore the speech act initiates H believing that S believes P. Similarly for the *request* speech act:

$$succeeds(E) \leftarrow act(E,\ request(S,H,A)),\ hold\text{-}at(bel(S,cando(H,A)),E),\ holds\text{-}at(bel(S,\ int(S,\ request(S,H,A))),E) \quad \text{R4}$$

$$initiates(E,\ bel(H,\ bel(S,int(S,A)))) \leftarrow act(E,\ request(S,H,A)) \quad \text{R5}$$

stating that the preconditions of S requesting H to do an action A are that S believes H can do the action and S intends to make the request, and the action has the effect that H believes S believes it intends action A done.

Further rules can provide a relationship between the speaker and hearer, for example trust:

$$holds\text{-}at(bel(H,P),\ T) \leftarrow holds\text{-}at(bel(H,\ bel(S,P)),\ T).$$

In addition to these there are domain-specific rules about action perconditions and agents' knowledge. Now, suppose *h* receives a *request* speech act to do an action *A*. Firstly, through R5 and deduction, this initiates a belief in *h* that that *s* intends *A* to be done, and through R4 and abductions the preconditions of the successful request action are assumed. These, in turn, through deductions and abductions of *</2*, *happens*, and *act* atoms (using rules including the core rules, R1, R2, R3, R4), leads to hypotheses about what *s* intends to achieve by *h*'s execution of *A*.

3.4.2. Sadri [2011b]

Another approach based on the Event Calculus is WIREC (Weighted Intention Recognition based on Event Calculus), proposed by Sadri. The primary motivation behind WIREC is to design an approach that works both with plan libraries and, in the absence of plan libraries, with a basic action theory. Thus it is aimed at increasing the chances of recognizing an intention even if the actor's plans are unknown or the actor is behaving in new ways. On the other hand, if a plan library is known for the actor, it is exploited as much as possible.

Compared to Quaresma and Lopes, WIREC requires a fairly simple set of Event Calculus axioms, specifying only what properties actions initiate, what properties actions terminate, action preconditions, and ramifications, as follows:

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

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

Precondition: $precondition(A,P)$

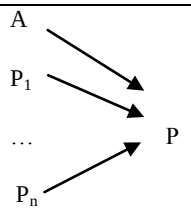
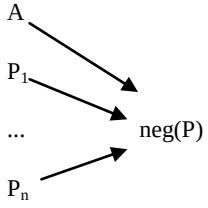
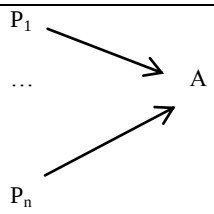
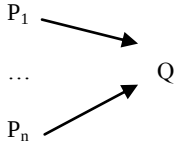
Ramification: $holds(Q, T) \leftarrow holds(P_1,T), \ldots, holds(P_n,T).$

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*. In these rules the conditions $holds(P_1,T), \ldots, holds(P_n,T)$ are called *qualifying conditions*. The third rule schema states that for action *A* to be executable fluent *P* must hold. An action may have any number of preconditions. Preconditions specify properties that need to hold for the action to be executable. Qualifying conditions specify properties that need to hold for the action to have a specific effect after it is executed. Ramifications hold as a result of other fluents (primitive or ramification) holding: For example:

$holds(draught, T) \leftarrow holds(window\text{-}open,T), \ldots, holds(windy,T).$

WIREC adopts 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 the intention recognition algorithm to be interpreted both in terms of (deductive) reasoning and in terms of graph matching or traversal. Plan libraries in WIREC consist of "joined-up" graph fragments such as the one in Figure 2, and basic action theories consist of graph fragments such as instances of those in Table 1. Figure 2 shows a plan for achieving intention *r* by doing actions *a1*, *a2*, *a3* in any order, and doing *a4* after *a1* and *a2*.

Table 1. WIREC Graph Fragments

Event Calculus Axiom Name	Event Calculus Axiom Schema	Graph Representation
Initiation	initiates(A,P,T)← holds(P_1,T) ∧ … ∧ holds(P_n,T)	
Termination	terminates(A,P,T)← holds(P_1,T) ∧ … ∧ holds(P_n,T)	
Precondition	precondition(A,P_1), precondition(A,P_2), …… precondition(A,P_n) being all the precondition axioms for action A	
Ramification	holds(Q,T)← holds(P_1,T) ∧ … ∧ holds(P_n,T)	

The search for hypothesis about intentions focuses on the executed actions and any further information known about the state, propagating them through graph matching and traversal (which can also be thought of as forward reasoning). If a plan library is used the traversal is done with the existing plans. If a basic action theory is used, instead, the graph matching will dynamically form (partial) plans from graph fragments according to the actions

of the actor. The algorithm can switch from the first mode to the second, if the known plan libraries do not prove useful in a particular instance.

As an example consider the graph fragments in Table 2, where *a1, a2, a3, a4* are actions, and *p, p1, p2, p3, q, q1, q2, q3, r, r1*are fluents. If action *a1* is observed, the graph traversal suggests a possible immediate intention *q* (via 2i), possibly to enable action *a2* (via 2ii), in turn with possible longer term intentions *q1, q2* and *r*. The other graph fragments are ignored as they provide no match. Now if next action *a3* is observed it strengthens the hypotheses that *q1* is the intention (via 2iii and 2v). The strengthening is done via propagation of the "weight of evidence", which is a number between 0 and 1, and takes into account several factors, amongst them how many actions the actor has executed so far towards a particular intention. The same weight of evidence also provides a ranking for the hypotheses that are generated.

Various features are used to control the search. For example a weight of evidence heuristic threshold cuts the propagation, and (simple) user profiles are formalized in logic (e.g. *when it is cold at night he usually makes a hot drink*) and are used to select plans from WIREC libraries where possible. Similarly user-specific constraints are formalized (e.g. *he can't climb a ladder*) and used to prune the search.

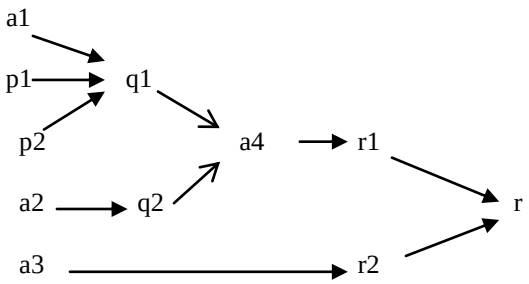


Figure 2. An Event Calculus WIREC plan for achieving an intention *r*

Table 2. An Example of some graph fragments

2i	2ii	2iii	2iv
a1 → *q*	*q* → *a2*	*a3* → *p1*	*p* → *a4*
2v	2vi	2vii	2viii
a2 → *q1* *p1* ↗	*a2* → *q2* *p2* ↗	*q2* → *r* *q3* ↗	*d* → *r1* *a4* ↗

3.5. Other Approaches

3.5.1. Hong [2001]

Hong's intention recognition method has the flavour of GraphPlan [Blum and Furst, 1997] and is indeed inspired by it. Similarly to WIREC it is not dependent on plan libraries, but uses a causality theory. The causality theory is represented in first order logic, describing action

preconditions and effect, and goals that can be complex, and are similar to ramifications in the Event Calculus. The reasoning involved can be loosely considered as forward reasoning with the logical representation, essentially inferring the new state after each action observation, and in each such state checking if any of the set of pre-specified goals is fully or partially achieved. However, the algorithm, implemented in Prolog, is presented as one that generates directed graphs and then analyses paths within the graph.

The graph has nodes representing actions, propositions and goals, and edges representing causal links. An example, simplified from one given in Hong [2001], is given below. In the example *m(b, o, h)* and *m(b, h, o)*, respectively represent the action of moving the bag from office to home, and moving the bag from home to office, and *putIn(d, b, h)* represents the action of putting the dictionary in the bag which is at home. All other nodes represent propositions (fluents); *at(Obj, Loc)* represents *Obj* is at location *Loc* (the constant *c* stands for computer), and *in(d,b)* represents the dictionary is in the bag. The heavy edges represent effects of actions, the dashed edges represent persistence of propositions for one state to another, and the other edges represent preconditions of actions or, in the case of *in(d,b)*, a condition for one of the effects (*at(d,o)*) of the action. For example the proposition *at(c, h)*, computer is at home, persists through all the observed actions in the graph. The proposition *at(b,o)*, bag is at office, terminates with action *m(b, o, h)*, moving the bag from office to home, which has an effect *at(b,h)*, bag is at home.

For simplicity we have ignored goal nodes in the example below. Goal nodes are added to each state and they represent fully or partially achieved goals. To do so the algorithm considers every instantiation of every goal definition. If some or all of the conditions of a goal definition hold in a state the goal is added as a goal node to that state.

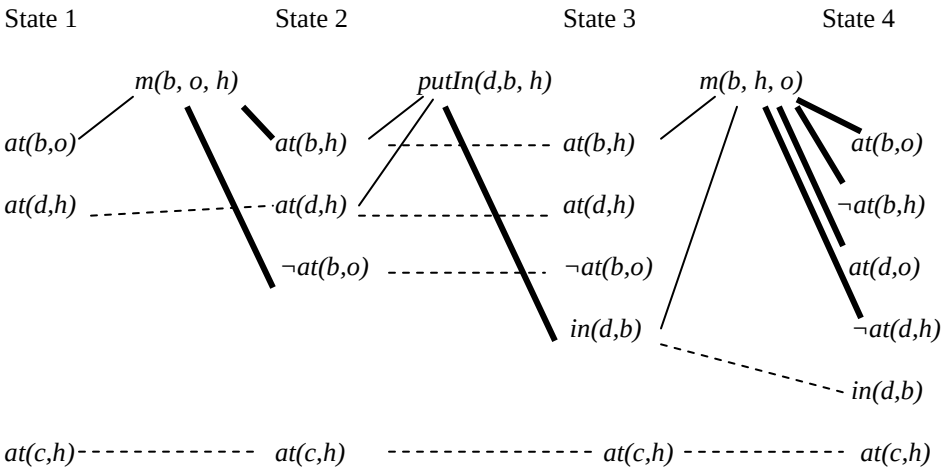


Figure 3. A goal graph

The objective of the algorithm is as follows: given a fully specified initial state and a set of observed actions and the order in which they are observed, to identify the goals that have been fully or partially achieved by a strict majority of the observed actions[8]. To determine which goals in a given state *S* fall into that category a causal pathway is traversed from goal nodes to the proposition nodes that are part of their definitions, to action nodes whose effect the propositions are, to propositions that are the precondition of the actions, and so on.

For example, suppose *office-supplied* is a goal with the following definition:

office-supplied $\leftarrow$ *at(d,o)* $\wedge$ *at (b,o)* $\wedge$ *at(c,o).*

Then in the graph above, in state 4 *office-supplied is* a partially achieved goal. The goal has a causality pathway that goes through all the observed actions.

There is no probability reasoning, so if multiple goals are identified, they are assumed to be equally likely. The algorithm is proved polynomial in the number of observed actions, the ground terms in the language, the number of propositions in the initial state, the number of goal definitions, and the largest number of effects amongst the actions.

3.5.2. Dragoni, Giorgini and Serafini [2002]

As with Quaresma and Lopes [1995], this paper focuses on recognizing the intentions behind the *request* and *inform* speech acts. Here, as in Quaresma and Lopes, the two parties are the hearer (*h*) and the speaker (*s*) of a speech act. The hearer attempts to recognise the beliefs and intentions of the speaker that led to the speaker producing the communication. The paper also uses abduction for the purpose, but it differs from Quaresma and Lopes in its formalization. It uses a *multi-context* system of beliefs and intentions where the contexts are connected via *bridge rules*.

The contexts correspond to agents' beliefs and intentions, and nested beliefs and intentions about other agents' beliefs and intentions. For example B_iI_j is a context, representing the beliefs of agent *i* about the intentions of agent j. The formula $\varnothing$ in this context, denoted $B_iI_j{:}\varnothing$, states that agent *i* believes that agent *j* intends $\varnothing$. $B_iB_jI_i$ is another context, representing the beliefs of agent *i* about the beliefs of agent *j* about the intentions of agent *i*. The formula $\varnothing$ in this context, denoted $B_iB_jI_i{:}\varnothing$, states that agent *i* believes that agent *j* believes that *i* intends $\varnothing$.

There are three kinds of bridge rules: *reflection down*, *reflection up*, and *belief-to-intention* rules. *Reflection down* allows an agent *i* to reason with its image of the mental state of another agent *j* (for example eliminating B_j from $B_j\varnothing$). *Reflection up* allows agent *i* to lift up the result of such reasoning to ascribe it to its beliefs about agent *j*. For example:

If $B_i{:}\ B_jP$ and $B_i{:}\ B_j(P\rightarrow Q)$ then by *reflection down* $B_iB_j{:}\ P$ and $B_iB_j{:}\ (P\rightarrow Q)$.

Then within the context B_iB_j, by modus ponens we can derive Q. Thus $B_iB_j{:}\ Q$, and by *reflection up* we obtain $B_i{:}\ B_jQ$.

[8] There is a common-sense assumption that most of the actions the agent performs are directed towards his goal.

The *belief-to-intention* rules connect an agent's intention to its belief, for example:

B_i*:raining* — *bring-umbrella* I_i

or

B_i*:temp-higher-20°* $\wedge$ *conditioning-on* — *stop-working* I_i

The first rule, above, states that if *i* believes it is raining then *i* will have the intention of bringing an umbrella. The second rule states that if *i* believes that the temperature is higher than 20° and the conditioning is on then *i* will have the intention to stop working. The *belief-to-intention* rules have the flavour of simple reactive production rules. In particular they capture the relationship between the beliefs and intentions of the hearer. On the other hand, what the hearer believes about the reactive behaviour rules of the speaker are formalized as ordinary rules. For example in the context B_h the rule

$$B_s(\textit{temp-higher-20°} \wedge \textit{conditioning-on}) \rightarrow I_s(\textit{stop-working})$$

represents what *h* believes about *s*'s intention when *s* believes the temperature is higher than 20° and the conditioning is on. The underlying language of the multi-context system is propositional logic, and the inference within contexts is based on natural deduction.

The work assumes a plan-based model of speech acts, in the sense that speech acts, as any other actions, have pre-conditions and post-conditions. Thus, an agent might utter certain speech acts as part of a plan to achieve an intention. The core assumption in the paper is thus that there is a causal relationship between an agent's mental state and his utterances.

Two speech acts are considered, *inform(s,h,∅)* and *request(s,h,∅)*, where *s* represents the speaker and *h*, the hearer. *inform(s,h,∅)* represents *s* telling *h* that ∅ holds, and *request(s,h,∅)* represents *s* asking *h* whether or not ∅ holds [9]. A pre-condition and a post-condition of *inform(s,h,∅)* are, respectively, that *s* believes ∅, and *h* believes that *s* believes ∅. Similarly, a pre-condition and a post-condition of *request(s,h,∅)* are, respectively, that *s* believes neither ∅ nor ¬∅, and *s* believes that *h* believes that *s* intends to believe ∅.

The hearing agent does not necessarily have any representation of the planning rules of the speaker, in contrast, say, with the work of Sindlar, et al [2008]. The hearing agent may have some snippets of the behaviour rules of the speaker, such as the rule:

$$B_s(\textit{ temp-higher-20°} \wedge \textit{conditioning-on}) \rightarrow I_s(\textit{stop-working}),$$

but he does not have any explicit representation of any planning rules of the speaker that link the speaker's intentions to his utterances of speech acts. The absence from the knowledge of the hearer of any explicit representation of any link between the intentions of the speaker and the actions that the hearer can observe (i.e. utterances of speech acts) is one crucial difference between this work and all others in this study, and, in fact in the majority of work in the literature on intention recognition.

[9] Note that here, slightly differently from Quaresma and Lopes [1995], above, the *request* is for information, rather than for an action to be done. In practice, however, Quaresma and Lopes can get the same effect by a request that an *inform* action be done.

In the absence of any such explicit information, the hearing agent works with the assumption that the intention behind a speech act *inform(s,h,∅)* is to bring about in *h* a mental state in which *h* believes or intends some formula ψ, and the intention behind a speech act *request(s,h,∅)* is to bring about in *s* a mental state in which *s* believes or intends some formula ψ. Also in both cases $\varnothing$ is expected to be useful for the derivation of the belief or intention ψ. So the intention recognition task with respect to both utterances is to determine what the ψ is, and whether the intention is to believe ψ or to intend ψ.

So, to attempt to put this in the frameworks discussed in Section 2, conceptually, it is as if the hearer has rules of the form below with which it performs abduction:

If $I_s I_h \psi$ *and "*$\varnothing$ *is relevant to* ψ*" then inform(s,h,*$\varnothing$*)*

If $I_s B_h \psi$ *and "*$\varnothing$ *is relevant to* ψ*" then inform(s,h,*$\varnothing$*)*

If $I_s I_s \psi$ *and "*$\varnothing$ *is relevant to* ψ*" then request(s,h,*$\varnothing$*)*

If $I_s B_s \psi$ *and "*$\varnothing$ *is relevant to* ψ*" then request(s,h,*$\varnothing$*).*

When *h* receives a speech act *inform(s,h,∅)*, to find out what such a ψ may be, and to determine if the intention is for him to believe or intend ψ, he reasons about the changes that $\varnothing$ brings to its beliefs and intentions. It can reason about the consequences of $\varnothing$ within any context available to it. For example, an appropriate context would be what *h* believes *s* believes about *h*'s beliefs, i.e. the context $B_h B_s B_h$. The new consequences of $\varnothing$, i.e. the consequences that were not derivable before the addition of $\varnothing$, but are derivable afterwards, are candidate hypotheses for the intention of the speaker.

A similar process is adapted with the speech act *request(s,h,∅)*. As an example suppose *h* receives a communication *request(s, h, conditioning-on)*, to be interpreted as *s* asking *h* if the conditioning is on. Then if *h* can prove within its image of the mental state of *s* that *temp-higher-20°* and the image also includes or allows the derivation of the rule

temp-higher-20° $\wedge$ *conditioning-on* $\rightarrow$ *stop-working*,

then *h* can hypothesise that *s*'s intention behind the speech act is to stop working.

3.5.3. Pereira and Anh [2009b]

This paper describes an implemented logic programming framework incorporating *situation-sensitive Causal Bayes Nets* (CBNs) for intention recognition in the care of the elderly. The CBNs provide a graphical representation and are translated into a declarative language called P-log which represents the same information in logical terms. P-log combines logical and probabilistic reasoning, the logical part based on Answer Set Programming (ASP) [Hu et al. 2007] and the probabilistic part based on CBNs.

The CBNs consist of nodes representing causes, intentions, actions and effects of actions. Causes give rise to intentions, somewhat like reactive production rules (e.g. *if you are thirsty then you (intend to) drink*). Intentions give rise to actions, somewhat like goal reduction production rules (e.g. *if you intend to drink then you look for a drink*), as in Figure 4, where we have ignored effects of actions.

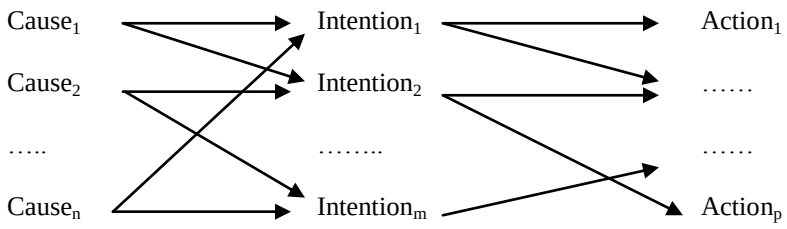


Figure 4. An abstract CBN

In the running example in the paper there is one action, which the elder is observed to be doing, namely looking for something in the sitting room. The possible intentions from which the system can choose from are looking for a book or for a drink or for the TV remote or the light switch. The causes include that he is thirsty, likes to read, or likes to watch TV.

The paper makes no assumptions about the planning methodology or representation of the observed agent or about its relationship to the knowledge and representation used by the observer. The approach is entirely based on the knowledge representation the observer uses specifically for intention recognition. The observer's statistical knowledge is compiled as CBNs. The causes in the resulting CBNs are either attributes that can be observed, for example *the light is on* or *the TV is on*, or are attributes that can be surmised, for example (*the elder is) thirsty*. The causes have pre-specified probabilities, either unconditional probabilities represented as facts (in P-log), for example:

the probability of being thirsty is 50/100,

or situation-sensitive probabilities represented as rules, for example:

the probability of being thirsty is 70/100 ← *temperature is higher than 30.*

Similarly the probability distributions of intentions conditional on causes are given as rules, for example:

the probability of the intention to look for a drink is 9/10 ← *the light is on* ∧ *thirsty*,

as are the probability distributions of actions conditional on intentions, for example:

the probability of looking for something is 99/100 ← *the intention to look for a book is true* ∧ *the intention to look for a drink is true* ∧ *the intention to look for the remote is true*

the probability of looking for something is 30/100 ← *the intention to look for a book is false* ∧ *the intention to look for a drink is true* ∧ *the intention to look for the remote is false.*

(Note the conjunction of possible intentions in the conditions of these rules, especially in the first rule, rather than the disjunction). All the probability distributions, conditional and unconditional, as well as the list of all possible intentions, causes and actions are pre-specified by the designer of the system.

Given this formalization, intended specifically for a P-log implementation, the probabilities of each of the possible intentions can be obtained conditional on observations regarding the status of the light and TV (on/off), temperature (related to the causes), and the observation that the elder is looking for something.

It is difficult to frame this approach in the methodologies seen in Section 2. Whilst the reasoning from actions to possible intentions has the flavour of abduction, here the explanation is a conjunction of possible intentions rather than a disjunction. This is because of the use of Bayesian networks. Each possible intention is exclusive of the others, and thus the choice of one intention excludes the others.

Notice that there is some similarity between this work and that of Mulder and Voorbraak, described in Subsection 3.1.1. In both the observer does not have all the information about the action it observes (here he observes that the elder is looking for *something*, for example). Here, however, the intentions are designed to provide hypotheses as to what the missing information may be (the *something* is a book, or the T.V. remote, for example).

3.6. Using Logic for Intention Recognition in Moral Judgments

Mao, W. and Gratch [2004, 2005, 2009]

Another perspective on intention recognition is its use in moral judgements, as exemplified by the work of Mao and Gratch. Their work focuses on determination of social judgements, as opposed to formal, legal judgments. To this end, intention is used as an epistemic concept as part of a computational model of social causality and attribution of responsibility and blame. The computational model combines the concept of intention with foreseeability and coercion. Intention and foreseeability (foreknowledge of the outcomes of actions that are executed) increase responsibility and blame of the actor, but coercion by another agent limits the available choices of the actor and reduces his responsibility and blame.

A simple model may always assign responsibility and blame to the actor whose action directly produces the outcome. More sophisticated models may delve deeper into how the actor came to perform the action, and may, for example, assign responsibility and blame to the highest authority, which ordered or coerced the action. Moreover, an agent may intentionally perform an action, but may not intend all its effects. Thus there are two notions of intentionality, *outcome intent* (intending the effect of an action), and *act intentionality* (intending the action, itself). It is often the former that is used in the judgment of responsibility.

Similar to other approaches to intention recognition, Mao and Gratch take into account the observer's knowledge of the actor's actions, including communication events in which he has participated, as speaker or hearer, and a causal theory of actions and their effects, in the form of action theories and plan libraries. The communication events are used to determine if the actor was coerced by another, and if he was aware of other alternative actions that would achieve the same goal, but would have other (side) effects.

Inferences about intentions are made depending on actions, dialogue exchanges and the power relationship between the speaker and the hearer. The communication acts considered are *order* and *request* on the speaker's side, and *accept*, *reject* and *counter-propose* on the hearer's side.

The interplay between communication events and intention recognition and attribution can be exemplified as follows. The communication acts *order* and *request* point to the speaker's intention (immediate intention, rather than longer term). The first creates an obligation on the hearer if he is a subordinate of the speaker. Acceptance by the hearer can point to the hearer's intention (again immediate, rather than longer term). The following is an axiom formalizing part of this:

$\forall S, H, P, T1, T2, T3, T4$
$intend(S, P, T1) \wedge \neg obligation(H, P, S, T2) \wedge accept(H, P, T3) \wedge T1<T3 \wedge T2<T3<T4 \rightarrow intend(H, P, T4)$

stating that if the speaker, *S*, intends a proposition *P* at time *T1*, and the hearer, *H* does not have an obligation *P* towards *S* at time *T2*, but *H* accepts *P* at a later time *T3* (presumably while *H* still has no obligation *P* towards *S*) , then *H* intends *P*. Also if *H* is coerced to do an action *A* by *S* then *H* intends *A*:

$\forall H, A\ (\exists S(coerced(H, A, S)) \rightarrow intend(H, A))$.

Other axioms suggested are:

$intend(X, A) \wedge \neg\exists Y\ (coerced(X, A, Y)) \rightarrow \exists P\ (P \in consequence(A) \wedge intend(X, P))$

$intend(X, P) \rightarrow \exists A\ (P \in consequence(A) \wedge intend(X, A))$

$intend(X\ A) \wedge P \in consequence(A) \wedge intend(X, P) \rightarrow know(X, bring\text{-}about(A, P))$

In each axiom all variables are assumed universally quantified over the whole axiom, unless denoted otherwise. The first states that if an agent intends an action *A*, and the agent is not coerced to do A (i.e., *A* is a voluntary act), then the agent intends at least one consequence of *A*. The second axiom states that if an agent intends an outcome *P*, then the agent intends at least one action that has consequence *P*. The third axiom states that if an agent intends an action *A* and a consequence *P* of *A*, then the agent knows that *A* brings about *P*.

Further inferences are suggested, but the axioms are not provided. For example, if the speaker orders an action *A1*, and the hearer makes a counter-proposal *A2*, then, firstly, both know the alternative *A2*, and, secondly, the hearer does not intend the original action *A1*. Now if the speaker rejects the counter-proposal and re-orders the original action, then the speaker intends the original action and not the alternative. It may also be possible to infer that the speaker intends at least one outcome which is a consequent of the original action but not of the alternative. To relate this to blame attribution, if the hearer is a subordinate of the speaker and accepts to do *A1*, then the speaker is responsible and is blamed for any negative consequences of *A1* that are not consequences of *A2*.

Tomai and Forbus [2008] provide a similar theory of intentions and responsibility, but extended to represent increasing qualitative degrees of responsibility along the criteria of an action causing an effect, without foreknowledge, with foreknowledge but without intent, with intent but coerced, with intent and not coerced. However, they provide no further contribution to intention recognition.

3. Conclusion and Challenges

In this paper we discussed logic-based approaches to intention recognition. We looked at different knowledge representation and reasoning mechanisms and we looked at the relationship between the two. We then described and analysed a number of concrete contributions in this field. Table 3 provides a brief overview of the features of these contributions.

Intention recognition has been a long-standing research area. Not surprisingly, its application areas have changed during the intervening years, moving from Unix help facilities and language understanding to broader areas of ambient intelligence and intrusion and

terrorism detection. Advances in activity recognition, sensor technology and RFID tags and readers allow further developments towards realistic and difficult applications.

Table 3. A summary of some of the features of the reviewed works

	Formalism	Reasoning	Application	Requires full observability?
Mulder & Voorbraak	Simple plan libraries	Abductive	Tactical/Military	No, allows existentially quantified variables in observations
Myers	HTN plan libraries	Abductive	Co-operative planning	Yes
Jarvis et al.	Extended HTN plan libraries	Abductive	Terrorist intention recognition	No, allows varying degrees of accuracy in observations
Sindlar et al.	Transformed BDI-type plan libraries	Abductive	Generic	No, allows partial observability
Demolombem & Fernandez	Situation Calculus and GOLOG	Probabilistic Deductive	Generic, but with focus on recognizing intentions of airplane pilots	Yes
Baier	CONGOLOG	Deductive	Generic	Yes
Goultiaeva, &. Lesperance	CONGOLOG	Deductive	Generic	Yes
Quaresma & Lopes	Event Calculus	Abductive	Recognising intentions behind speech act utterances	Yes
Sadri	Event Calculus	Deductive	Generic	No, allows partial observability
Hong	Predicate logic	Deductive	Generic	Yes
Dragoni et al.	Multi-context logical theories with	Abductive	Recognising intentions behind speech	Yes

	bridge rules		act utterances	
Pereira & Anh	Causal Bayes nets	Baysian	Generic, but with focus on elder care	No, allows partial observability
Mao & Gratch	Predicate logic	Deductive	Social Moral Judgments	No, , allows partial observability

3.1. Challenges

There are many challenges requiring further developments of methodologies. Below we discuss some of these challenges.

3.1.1. Reliance on Plan Libraries

Much of the current work on intention recognition assumes the existence of plan libraries. This requires much human effort in predicting and formalizing plans, and may be unrealistic in many cases. It may also be unrealistic to assume that the observer agent has knowledge of the plan libraries of the observed agent. Furthermore, the intention recognition system is hampered in cases where an agent attempts novel ways of achieving a goal.

Another serious issue regarding plan libraries has been discussed in Goldman, Kabanza, Bellefeuille [2010]. The issue concerns the sensitivity of intention recognition algorithms to the representation used in plan libraries. Two plan representations may be *equivalent* in the sense that they identify the same atomic actions with the same ordering in solving a goal, but they can give probabilistically different results when used for goal recognition. For example consider the two *equivalent* plan representations below for achieving goal *G*. Both plans representations denote that *G* is achieved by doing action *a*, and then actions *b*, *c*, *d* in any order. The first plan shows this directly, and the second shows this via auxiliary subgoals S1, S2 and S3. When arcs are joined by a line across them it denotes "and", and when there is no such line it denotes "or". Given a sequence of observed action executions, for example *a;b;c*, the PHATT goal recognition system [Geib and Goldman, 2001] will compute two different probabilities for goal *G*, with the two plan representations. The WIREC algorithm [Sadri 2011b] too will compute two different weights for goal *G*.

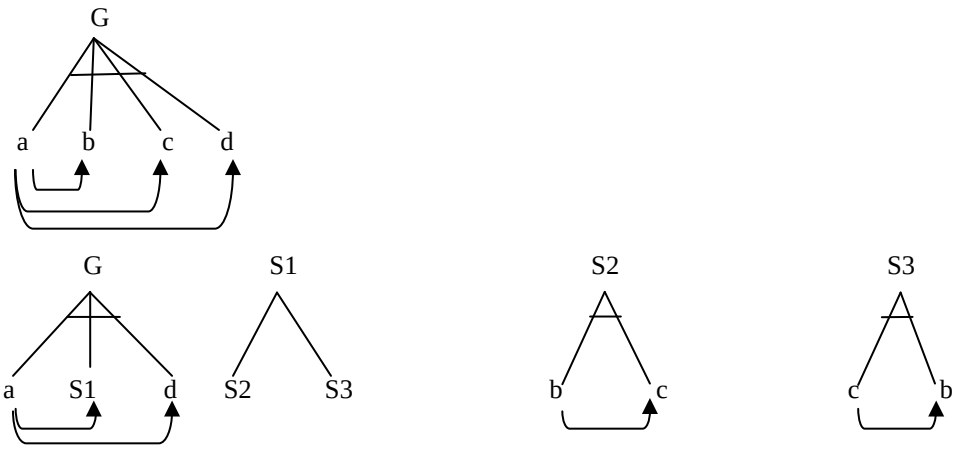


Figure 4. Two alternative plans for goal G

3.1.2. Partial Observability of Actions

Dealing with partial observability of actions remains an issue. Of course this can be one of the major issues in the adversarial and diversionary cases of intention recognition. But even, in the relatively less complex cases of intended or keyhole cases there is no guarantee of observing every action that is executed. Indeed in the assisted living Smart Home scenario clients may not be in favour of having their every move observed.

Current attempts to deal with partial observability include detecting that an action has been executed from changes in the environment [Geib and Goldman 2001], even though the action itself has not been observed, and attaching probabilities to non-observation of actions [Geib and Goldman 2005]. However, a general, efficient and scalable solution remains a challenge. This may be an area where abduction can make a further contribution in intention recognition. Abductive reasoning can be used to generate hypotheses about what actions have been executed, unobserved, compatibly with actions that have been observed, so that together they can account for the changes in the environment.

3.1.3. Multiple Intentions, Interleaved Execution of Plans for Different Intentions, Concurrent Execution of Alternative Plans for the Same Intention, Abandoned Intentions

Much work on intention and plan recognition assumes one single observed agent, with one single intention and plan. Consequently it attempts to find hypotheses consisting of one intention or one plan that explains all, or the majority of, the observed actions of that agent, with the possible exception of what it considers as "insignificant" actions.

However, there are common scenarios where the acting agent has multiple intentions and may interleave the execution of his plans of actions for achieving them. For example consider the scenario of someone preparing a meal. They may well be interleaving several plans for different goals, preparing a sauce, cooking vegetables, putting something in the oven, preparing the dessert, pouring a drink, etc. Agents may also abandon their intentions partway. Geib [2002] discusses complications that arise because of these with intention recognition in the I.L.S.A. Smart Home system. Another, similar difficult case is where the actor is concurrently trying out alternative plans for achieving the same intention.

3.1.4. Actions of Less Competent Individuals

In addition to assuming one single intention and plan, many intention recognition systems assume the actor is rational and competent and is following a correct plan. Intention recognition becomes more difficult when we attempt to interpret the actions of less competent individuals. For example in tutoring systems the student may make mistakes, someone new to a task may be learning it by trial and error, and cognitively impaired individuals may execute actions in error and confusion. Roy et. al. [2007], for example, discuss the challenges of intention recognition in the case of Alzheimer patients.

3.1.5. Actions Performed As Reactions Rather Than in Pursuit of Goals

Most, if not all, intention recognition systems assume the actions of the actor are directed towards a goal. They do not address actions that are performed in reaction to some social or environmental perception (trigger), without any specific (explicit) goal. It would be interesting to investigate to what extent the intention recognition systems can be used, not to recognize goals, but, to recognize the trigger of such reactive actions.

3.1.6. Multi-Agent Intention Recognition

Terrorism detection and other applications increase the demand for multi-agent intention recognition, where several agents co-operate on the same or on several related intentions. Even in the Smart Home scenario multiple actors may co-operatively pursue one intention. The multi-agency exacerbates all the issues discussed above, and brings with it additional challenges, for example identifying related clusters of agents, and identifying which agents are contributing to which intentions. Sukthankar and Sycara [2008] and Anh and Pereira [2010] propose probabilistic algorithms for recognizing the (joint) intention of multiple (co-operating) agents.

Of course, all the above issues become more challenging if the acting agents deliberately hide their actions or act to mislead the observers. Very little work currently addresses these types of cases.

Acknowledgement

I am grateful to Bob Kowalski for discussions about the topic of intention recognition and for reading and commenting on earlier drafts of this paper. I am also grateful to the anonymous reviewers for their helpful comments.

References

[1] Aarts, E. 2004. Ambient intelligence: a multimedia perspective. *IEEE Intelligent Systems 19(1)*, 12 19.

[2] Amigone, F. , Gatti, N., Pinciroli, C., Roveri, M. 2005. What planner for ambient intelligence applications? *IEEE Transactions on Systems, man and cybernetics – part A*, IEEE Press, 35(1), January 2005, 7-21.

[3] Anh, H. T., Pereira, L. M., 2010. Collective intention recognition and elder Care, In Proceedings of the Proactive Assistant Agents (PAA 2010) , AAAI 2010 Fall Symposium, 11-13 November 2010, Arlington, Virginia, USA, 26-31.

[4] Appelt, D.E., Pollack, M.E. (1992). Weighted Abduction for Plan Ascription. *Journal of User Modeling and User-Adapted Interaction*, Vol. 2, Nos 1-2, March 1992, 1-25.

[5] Augusto, J.C., Liu, J., McCullagh, P., Wang, H., Yang, J-B. 2008. Management of uncertainty and spatio-temporal aspects for monitoring and diagnosis in a smart home. *International Journal of Computational Intelligence Systems*, Volume 1, Number 4, December, 2008, 361-378.

[6] Avrahami-Zilberbrand, D., Kaminka, G.A.(2005). Fast and complete symbolic plan recognition. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) 2005, 653-658.

[7] Avrahami-Zilberbrand, D., Kaminka, G.A. (2007). Incorporating observer biases in keyhole plan recognition (efficiently!). In Proceedings of the Twenty-Second National Conference on Artificial Intelligence (AAAI-07), 2007.

[8] Barger, T., Alwan, M., Kell, S., Turner, B., Wood, S., Naidu, A. (2002) Objective remote assessment of activities of daily living: analysis of meal preparation patterns. Poster presentation, http://marc.med.virginia.edu/pdfs/library/ADL.pdf, 2002.

[9] Baier, J. A. (2002). On procedure recognition in the Situation Calculus. In Proceedings of the Twelfth International Conference of the Chilean Computer Science Society (SCCC-02), 33-42.

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

[11] Bosse, t., Hoogendoorn, m., Klein, m.c.a., Treur, J. 2008. A component-based ambient agent model for assessment of driving behaviour. In *Proceedings of the 5th International Conference on Ubiquitous Intelligence and Computing (UIC-08)*, Oslo, Norway, June 23-25, 2008, F.E. SANDNES, Y. Zhang, C. Rong, L.T. Yang, J. Ma, Eds., Lecture Notes in Computer Science LNCS 5061, Springer-Verlag Berlin Heidelberg, 229-243.

[12] Bui, H. H. (2003). A general model for online probabilistic plan recognition. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) 2003.

[13] Burghardt, C., Kirste, T. (2007). Inferring intentions in generic context-aware systems. In Proceedings of the International Conference on Mobile and Ubiquitous Multimedia, MUM'07, 50-54.

[14] Busetta, P., Kuflik, T., Merzi, M., Rossi, S. 2004. Service delivery in smart environments by implicit organisations. In *Proceedings of 1st International conference on Mobile and Ubiquitous Systems (MobiQuitous '04)*, Boston, Masachusetts,USA, IEEE Comp. Society.

[15] Carberry, S., Elzer, S. (2007). Exploiting evidence analysis in plan recognition. UM 2007, LNAI 4511 C. Conati, K. McCoy, G. Paliouras (Eds.), Springer-Verlag Berlin Heidelberg 2007, 7-16.

[16] Cesta, A., Cortellessa, G., Pecora, F., Rasconi, R. 2005. Exploiting Scheduling techniques to monitor the execution of domestic activities. *Intelligenza Artificiale, 2*(4).

[17] Charniak, E., McDermott, D. (1985). *Introduction to artificial intelligence*. Addison Wesley, Reading, MA, 1985.

[18] 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, Nov. 2004, 36-40.

[19] 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, 1981, 245-274.

[20] Cox, M.T., Kerkez, B. (2006).Case-Based Plan Recognition with Novel Input. *International Journal of Control and Intelligent Systems* 34(2), 2006, 96-104.

[21] Da Silva, F. S. C., Vasconcelos, W. W. 2007. Managing responsive environments with software agents. *Journal of Applied Artificial intelligence*, 21:4, 469-488.

[22] Demolombe, R., Fernandez, A.M.O. (2006). Intention recognition in the Situation Calculus and probability theory frameworks. In Proceedings of Computational Logic in Multi-agent Systems (CLIMA) 2006.

[23] Dragoni, A.F., Giorgini, P., Serafini, L. (2002) Mental States Recognition from Communication. *Journal of Logic Computation*, Vol 12, No. 1, 2002, 119-136.

[24] Encarnacao, J. L., Kirste, T. 2005. Ambient intelligence: towards smart appliance ensembles, *E.J. Neuhold Festschrift*, M. Hemmje et al. Eds., Springer-Verlag, Berlin Heidelberg, LNCS 3379, 261-270.

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

[26] Favela, J., Rodriguez, M., Preciado, A., Gonzalez, V.M. 2004. Integrating context-aware public displays into a mobile hospital information system. *IEEE Transactions on Information Technology in Biomedicine*, Volume 8, Number 3, September 2004, 279-286.

[27] Fung, T.H., Kowalski, R. (1997). The IFF Proof Procedure for Abductive Logic Programming. *Journal of Logic Programming*, 1997.

[28] Gaggioli, A. 2005. Optimal experience in ambient intelligence, Ambient Intelligence. In *Ambient Intelligence*, G. RIVA, F. VATALARO, F. DAVIDE, M. ALCANIZ. Eds., IOS Press Amsterdam, 35-43.

[29] 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, 2001.

[30] Geib, C. W., Steedman, M. (2003). On natural language processing and plan recognition. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) 2003, 1612- 1617.

[31] 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 2005.

[32] Gelfond, M, Lifschitz, V. (1992). Representing actions in extended logic programs. In Proceedings of the International Symposium on Logic Programming, 1992.
[33] Giacomo, G. D., Lesperance, Y., Levesque H. (2000). Con-Golog, a concurrent programming language based on the Situation Calculus: foundations. *Artificial Intelligence*, 121(1-2):109–169.
[34] Giroux S., Bauchet J., Pigot, H., Lusser-Desrochers, D., Lachappelle, Y. (2008). Pervasive behavior tracking for cognitive assistance. The 3rd International Conference on Pervasive Technologies Related to Assistive Environments, Petra'08, Greece, July 15-19, 2008.
[35] Goldman, R.P, Kabanza, F., Bellefeuille, P. (2010). Plan libraries for plan recognition: do we really know what they model? (Position Paper) In Proceedings of the AAAI Workshop on Plan, Activity and Intent Recognition (PAIR), 2010.
[36] Goodman, B. A., Litman, D. J. (1992). On the interaction between plan recognition and intelligent interfaces. User Modeling and User-Adapted Interactions 2(1-2), 1992, 83-115.
[37] Goultiaeva, A., Lesperance, Y. (2007). Incremental plan recognition in an agent programming framework. In Proceedings of Plan, Activity, and Intent Recognition (PAIR) 2007.
[38] Guralnik, V., Haigh, K.Z. 2002. Learning models of human behaviour with sequential patterns. In Proceedings of the *AAAI Workshop on Automation as Caregiver*, AAAI Press.
[39] Hong, Jun, 2001 Goal recognition through goal graph analysis, *Journal of Artificial Intelligence Research* 15, 2001, 1-30.
[40] Hu, P.H., Son, T.C., Baral, C. (2007). Reasoning and planning with sensing actions, incomplete information, and static laws using Answer Set Programming. *Theory and Practice of Logic Programming* 7(4), July 2007, 377-450.
[41] 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, 2004.
[42] Jarvis, P.A., Lunt, T.F., Myers, K.L. (2005). Identifying terrorist activity with AI plan-recognition technology, *AI Magazine*, Vol 26, No. 3, 2005, 73-81.
[43] Kakas, A., Kowalski, R., Toni, F. (1998). The role of logic programming in abduction. In: Gabbay, D., Hogger, C.J., Robinson, J.A. (Eds.): Handbook of Logic in Artificial Intelligence and Programming 5, Oxford University Press, 1998, 235-324.
[44] Karlsson, B., Ciarlini A.E.M., Feijo B., Furtado A.L. (2006). Applying a plan-recognition/plan-generation paradigm to interactive storytelling, The LOGTELL Case Study. Monografias em Ciência da Computação Series (MCC 24/07), ISSN 0103-9741, Informatics Department/PUC-Rio, Rio de Janeiro, Brazil, September 2007. Also in the Proceedings of the ICAPS06 Workshop on AI Planning for Computer Games and Synthetic Characters, Lake District, UK, June, 2006, 31-40.
[45] Kautz, H., Allen, J.F. (1986). Generalized plan recognition. In Proceedings of the Fifth National Conference on Artificial Intelligence (AAAI-86), 1986, 32-38.
[46] Konolige, K., Pollack, M.E. (1989). Ascribing plans to agents: preliminary report. In 11th International Joint Conference on Artificial Intelligence, Detroit, MI, 1989, 924-930.
[47] Kowalski, R. A., Sadri, F. (2009). Integrating logic programming and production systems in abductive logic programming agents. Invited papers, the 3rd International Conference on Web Reasoning and Rule Systems, October 25-26, Chantilly, Virginia, USA, 2009.
[48] Kowalski, R., Sergot, M. (1986). A logic-based calculus of events. In *New Generation Computing*, Vol. 4, No.1, February 1986, 67-95.
[49] Lesh, N., Rich, C., Sidner, C. L. (1999). Using plan recognition in human-computer collaboration. In Proceedings of the Seventh International Conference on User Modelling, Canada, July 1999, 23-32.
[50] Levesque, H., Reiter, R., Lesperance, Y., Lin, F., Scherl, R. (1997). GOLOG: A Logic Programming Language for Dynamic Domains. *Journal of Logic Programming* 31, 1997, 59-84.
[51] Mancarella, P., Terreni, G., Sadri, F., Toni, F., Endriss, U. (2009). The CIFF proof procedure for abductive logic programming with constraints: theory, implementation and experiments. *Theory and Practice of Logic Programming* 9, 2009.
[52] Mayfiled, J. (2000). Evaluating plan recognition systems: three properties of a good explanation. *Artificial Intelligence Review* 14, 2000, 351-376.
[53] Mao, W., Gratch, J. (2004). A utility-based approach to intention recognition. AAMAS 2004 Workshop on Agent Tracking: Modelling Other Agents from Observations.
[54] Mao, W., Gratch, J. (2009). Modeling social inference in virtual agents, *AI & Society*, Volume 24, Number 1, 5–11.
[55] Mao, W., Gratch, J. (2005). Social causality and responsibility: modeling and evaluation, in Proceedings of the 5th International Conference on Interactive Virtual Agents, Kos, Greece, 2005.
[56] Mao, W., Gratch, J. (2004). Social judgment in multiagent interactions, in Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, New York, 2004.

[57] Mihailidis, A., Fernie, G.R., Barbenel, J. (2001). The use of artificial intelligence in the design of an intelligent cognitive orthosis for people with dementia. *Assistive Technology* 13:23-39, 2001.

[58] Modayil, J., Bai, T., Kautz, H. (2008). Improving the recognition of interleaved activities. In the Proceedings of the 10th International Conference on Ubiquitous Computing (UbiComp), 2008, 40-43.

[59] Mulder, F., Voorbraak, F. (2003). A formal description of tactical plan recognition, *Information Fusion* 4, 2003, 47-61.

[60] Myers, K.L. (1997). Abductive completion of plan sketches. In Proceedings of the 14th National Conference on Artificial Intelligence, American Association of Artificial Intelligence (AAAI-97), Menlo Park, CA: AAAI Press, 1997.

[61] Myers, K. L., Jarvis, P. A., Tyson, W. M., Wolverton, M. J. (2003). A mixed-initiative framework for robust plan sketching. In Proceedings of the 13thInternational Conferences on AI Planning and Scheduling, Trento, Italy, June, 2003.

[62] Park, S., Kautz, H. (2008). Hierarchical recognition of activities of daily living using multi-scale, multi-perspective vision and RFID. The 4th IET International Conference on Intelligent Environments, Seattle, WA, July 2008a.

[63] Park, S., Kautz, H. (2008). Privacy-preserving recognition of activities in daily living from multi-view silhouettes and RFID-based training, AAAI Fall 2008 Symposium on AI in Eldercare: New Solutions to Old Problems, Washington, DC, November 7 - 9, 2008b.

[64] Peirce, C. S. (1958). *The collected papers of Charles Sanders Peirce*. Vols. VII-VIII, Edited by AW Burks. Cambridge, MA: Harvard, 1958.

[65] Pereira L.M., Alferes, J.J., Aparicio, J.N. (1992). Contradiction removal semantics with explicit negation. In Proceedings of the Applied Logic Conference, 1992.

[66] Pereira, L.M., Anh, H.T. (2009a). Intention Recognition via Causal Bayes Networks plus Plan Generation, in: L. Rocha et al. (eds.), Progress in Artificial Intelligence, Procs. 14th Portuguese International Conference on Artificial Intelligence (EPIA'09), Springer LNAI, October 2009.

[67] Pereira, L.M., Anh, H.T. (2009b). 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 2009.

[68] Pereira, L.M., Saptawijaya, A. (2009). Modelling Morality with Prospective Logic, in International Journal of Reasoning-based Intelligent Systems (IJRIS), 1(3/4): 209-221.

[69] 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*, 2005.

[70] Plocher, T., Kiff, L.M. 2003. Mobility monitoring with the Independent LifeStyle Assistant (I.L.S.A.) . In *Proceedings of the International Conference on Aging, Disability and Independence (ICADI)*, Washington DC, USA, 170-171.

[71] Quaresma, P., Lopes, J.G. (1995). Unified Logic Programming Approach to the Abduction of Plans and Intentions in Information-Seeking Dialogues. *Journal of Logic Programming*, Elsevier Science Inc., 1995, 103-119.

[72] Rao, M., Georgeff, P. (1995). BDI-agents: from theory to practice. In the Proceedings of the First International Conference on Multi-agent Systems (ICMAS'95), 1995.

[73] Reiter, R. (2001). *Knowledge in action: logical foundations for specifying and implementing dynamical systems*. MIT Press, 2001.

[74] Rodriguez, M.D., Favela, J., Martinez, E.A., Munoz, M.A. 2004. Location-aware access to hospital information and services. *IEEE Transactions on Information Technology in Biomedicine*, Volume 8, Issue 4, December 2004, 448 – 455.

[75] Rodriguez, M., Favela, J., Preciado, A., Vizcaino, A. 2005. Agent-based ambient intelligence for healthcare. *AI Communications* Volume 18, Issue 3, August 2005, 201-216.

[76] 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, 2007, 131- 137.

[77] Sadri, F. (2011a) Logic-based Approaches to Intention Recognition, In Handbook of Research on Ambient Intelligence: Trends and Perspectives, Nak-Young Chong and Fulvio Mastrogiovanni Eds., IGI Global, May 2011, 346-375.

[78] Sadri, F. (2011b) Intention Recognition with Event Calculus Graphs and Weight of Evidence, Proc. of the 3rd International Conference on Agents and Artificial Intelligence, January 2011, ed. J. Filipe, Springer, 2011.

[79] Sanchez, D., Tentori, M., Favela, J. (2008). Activity recognition for the smart hospital. *IEEE Intelligent Systems*, March/April 2008, 50-57.

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

[81] Sindlar, M.P., Dastani, M.M., Dignum, F., Meyer, J.-J.Ch. (2008). Mental state abduction of BDI-based agents. Declarative Agent Languages and Technologies (DALT), Estoril, Portugal 2008, 161-178.

[82] Sindlar, M.P., Dastani, M.M., Dignum, F.P.M., Meyer, J-J.Ch. (2010). Explaining and predicting the behavior of BDI-Based Agents in role-playing games. In M. Baldoni, J. Bentahar, M.B. van Riemsdijk & J. Lloyd (Eds.), Declarative Agent Languages and Technologies VII, 7th International Workshop Vol. 5948. Lecture Notes in Computer Science, 174-191.

[83] Sukthankar, G., Sycara, K. (2008). Robust and efficient plan recognition for dynamic multi-agent teams (Short Paper). In Proceedings of 7th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2008), International Foundation for Autonomous Agents and Multi-agent Systems, May 2008.

[84] Sukthankar, G., Sycara, K. (2008). Hypothesis Pruning and Ranking for Large Plan Recognition Problems. In Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (AAAI-08), June 2008.

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

[86] Tomai, E., Forbus, K. (2008). Using qualitative reasoning for the attribution of moral responsibility, In the Proceedings of the 30th Annual Conference of the Cognitive Science Society (CogSci), Washington, D.C.

[87] Wilensky, R. (1983). *Planning and understanding*. Addison Wesley, Reading, MA, 1983.