Intention-based Decision Making via Intention Recognition and its Applications

The Anh Han
Centro de Inteligência Artificial (CENTRIA)
Departamento de Informática, Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
E-mail: h.anh@campus.fct.unl.pt

Luis Moniz Pereira
Centro de Inteligência Artificial (CENTRIA)
Departamento de Informática, Faculdade de Ciências e Tecnologia
Universidade Nova de Lisboa, 2829-516 Caparica, Portugal
E-mail: lmp@fct.unl.pt

ABSTRACT

In this chapter we present an intention-based decision making system. We exhibit a coherent combination of two Logic Programming based implemented systems, Evolution Prospection and Intention Recognition. The Evolution Prospection system has proven to be a powerful system for decision making, designing and implementing several kinds of preferences and useful environment-triggering constructs. It is here enhanced with an ability to recognize intentions of other agents—an important aspect not well explored so far. The usage and usefulness of the combined system are illustrated with several extended examples in different application domains, including Moral Reasoning, Ambient Intelligence, Elder Care, and Game Theory.

INTRODUCTION

Given the crucial role and ubiquity of intentions in our everyday decision making (Bratman, 1987; Meltzoff, 2007; O. Roy, 2009b; Searle, 2010; Woodward, Sommerville, Gerson, Henderson, & Buresh, 2009), one would expect intentions to occupy a substantial place in any theory of action. However, in what concern perhaps the most prominent theory of action—rational choice theory (Binmore, 2009; Russell & Norvig, 2003)— which includes the theory of decision making—the attention is mainly, if not exclusively, given to actions, strategies, information, outcomes and preferences, but not to intentions (O. Roy, 2009a; van Hees & Roy, 2008).

This is not to say that no attention has been paid to the relationship between rational choice and intentions. Quite the contrary, a rich philosophical and Artificial Intelligence (AI) literature has developed on the relation between rationality and intentions (Bratman, 1987; Cohen & Levesque, 1990; Malle, Moses, & Baldwin, 2003; Singh, 1991; van Hees & Roy, 2008). Some philosophers, for example in (Bratman, 1987; O. Roy, 2009b), have been concerned with the role that intention plays in directing rational decision making and guiding future actions. In addition, many agent researchers have recognized the importance of intentions in developing useful agent theories, architectures, and languages, such as Rao and Georgeff with their BDI model (Rao & Georgeff, 1991, 1995), which has led to the commercialization of several high-level agent languages, e.g. in (Burmeister, Arnold, Copaciu, & Rimassa, 2008; Wooldridge, 2000, 2002). However, to the best of our knowledge, there has been no real attempt to model and implement the role of intentions in decision making, within a rational choice framework. Intentions of other relevant agents are always assumed to be given as the input of a decision making process; no system that integrates a real intention recognition system into a decision making system has been implemented so far.

In this chapter, we present a coherent Logic Programming (LP) based framework for decision making—which extends our previous work on Evolution Prospection for decision making (Pereira & Han, 2009a, 2009b)—but taking into consideration now the intentions of other agents. Obviously, when being immersed in a multi-agent environment, knowing the intentions of other agents can benefit the recognizing agents in a number of ways. It enables the recognizing agents to predict what other agents will do next or might have done before—thereby being able to plan in advance to take the best advantage from the prediction, or to act so as to take remedial action. In addition, an important role of recognizing intentions is to enable coordination of your own actions and in collaborating with others (Bratman, 1987, 1999; Kaminka, Tambe, Pynadath, & Tambe, 2002; O. Roy, 2009b; Searle, 1995, 2010). We have also recently shown the role of intention recognition in promoting improved cooperative behavior in populations or societies of self-interested agents (Han, 2012; Han, Pereira, & Santos, 2011b, 2012a, 2012b). A large body of literature has exhibited experimental evidence of the ability to recognize/understand intention of others in many kinds of interactions and communications, not only in Human but also many other species (Cheney & Seyfarth, 2007; Meltzoff, 2005, 2007; Tomasello, 1999, 2008; Woodward, et al., 2009). Furthermore, the important role of intention-based decision making modeling has been recognized in a diversity of experimental studies, including behavioral economics (Falk, Fehr, & Fischbacher, 2008; Frank, Gilovich, & Regan, 1993; Radke, Guroglu, & de Bruijn, 2012) and morality (Hauser, 2007; Young & Saxe, 2011). In AI application domains wherein an ability to recognize users' intentions is crucial for the success of a technology, such as the ones of Ambient Intelligence (Friedewald, Vildjiounaite, Punie, & Wright, 2007; Sadri, 2011a) and Elder Care (Giuliani, Scopelliti, & Fornara, 2005; Pereira & Han, 2011a; Sadri, 2010, 2011b), intention-based decision making is also becoming of increasing interest.

The Evolution Prospection (EP) system is an implemented LP-based system for decision making (Pereira & Han, 2009a, 2009b). An EP agent can prospectively look ahead a number of steps into the future to choose the best course of evolution that satisfies a goal. This is achieved by designing and implementing several kinds of prior and post preferences (Pereira, Dell'Acqua, & Lopes, 2012; Pereira & Lopes, 2009) and several useful environment-triggering constructs for decision making. In order to take into account intentions of other agents in decision making processes, we employ our previously implemented, also LP-based, intention recognition system, as an external module of the EP system. For an easy integration, the Bayesian network inference of the intention recognition system is performed by P-log (Baral, Gelfond, & Rushton, 2009; Han, Carroline, & Damasio, 2008), a probabilistic logic system[1]. In general, intention recognition can be defined as the process of inferring the intention or goal of another agent (called "*individual intention recognition*") or a group of other agents (called "*collective intention recognition*") through their observable actions or their actions' observable effects on the environment (Han & Pereira, 2010a; Heinze, 2003; Sadri, 2010; Sukthankar & Sycara, 2008).

The remainder of this chapter is structured as follows. In Section 2 we describe our two LP-based previously implemented systems, the Evolution Prospection system and Intention Recognition. On top of these two systems, in Section 3 we describe our intention-based decision making system, the main contribution of this chapter. Section 4 describes how our framework can be utilized to address several issues in the Ambient Intelligence and Elder Care application domains. Next, Section 5 points out how intentionality is important in the moral reasoning, and how our intention-based decision making system can be used therein. This section also demonstrates how our system can be useful to model different

[1] The implementation of P-log systems described in (Baral et al., 2009) can be found in: http://www.cs.ttu.edu/~wezhu/

issues in Game Theory, when strategies are characterized as modifiable intentions. The chapter ends with concluding remarks and future work directions.

BACKGROUND (SUBHEAD STYLE 1- ARIAL, SIZE 12, BOLD)

Evolution Prospection

The implemented EP system[2] has proven useful for decision making (Han, 2009; Han & Pereira, 2011b; Han, Saptawijaya, & Pereira, 2012; Pereira & Han, 2009a, 2009b). It is implemented on top of ABDUAL[3], a preliminary implementation of (Alferes, Pereira, & Swift, 2004), using XSB Prolog (XSB, 2009). We next describe the constructs of EP, to the extent we use them here. A full account can be found in (Han, 2009; Pereira & Han, 2009b).

Language Let **L** be a first order language. A domain literal in **L** is a domain atom A or its default negation *not A*. The latter is used to express that the atom is false by default (Closed World Assumption). A domain rule in **L** is a rule of the form:

$$A \leftarrow L_1, \ldots,\ L_t \quad (t \geq 0),$$

where A is a domain atom and $L_1, \ldots,\ L_t$ are domain literals. An integrity constraint in **L** is a rule with an empty head. A (logic) program P over **L** is a set of domain rules and integrity constraints, standing for all their ground instances.

Here we consider solely Normal Logic Programs (*NLPs*), those whose heads of rules are positive literals, or empty (Baral, 2003). We focus furthermore on abductive logic programs (Alferes, et al., 2004; Kakas, Kowalski, & Toni, 1993), i.e. NLPs allowing for abducibles – user-specified positive literals without rules, whose truth-value is not fixed. Abducibles instances or their default negations may appear in bodies of rules, like any other literal. They stand for hypotheses, each of which may independently be assumed true, in positive literal or default negation form, as the case may be, in order to produce an abductive solution to a query.

Definition 1 (Abductive Solution): An abductive solution is a consistent collection of abducible instances or their negations that, when replaced by true everywhere in P, affords a model of P (for the specific semantics used on P) which satisfies the query and the ICs – a so-called abductive model.

Active Goals In each cycle of its evolution the agent has a set of active goals or desires. We introduce the *on_observe/1* predicate, which we consider as representing active goals or desires that, once triggered by the observations figuring in its rule bodies, cause the agent to attempt their satisfaction by launching all the queries standing for them, or using preferences to select them. The rule for an active goal AG is of the form:

$$on\_observe(AG) \leftarrow L_1, \ldots,\ L_t\, (t \geq 0),$$

where $L_1, \ldots,\ L_t$ are domain literals. During evolution, an active goal may be triggered by some events, previous commitments or some history-related information. When starting a cycle, the agent collects its active goals by finding all the *on_observe(AG)* that hold under the initial theory without performing any abduction, then finds abductive solutions for their conjunction.

---

[2] The implementation of the Evolution Prospection system can be downloaded at: http://centria.di.fct.unl.pt/~lmp/software/epa.zip

[3] The implementation of ABDUAL system can be downloaded at: http://centria.di.fct.unl.pt/~lmp/software/contrNeg.rar

Preferring Abducibles An abducible A can be assumed only if it is a considered one, i.e. if it is expected in the given situation, and, moreover, there is no expectation to the contrary

$$consider(A) \leftarrow expect(A),\ not\ expect\_not(A),\ A.$$

The rules about expectations are domain-specific knowledge contained in the theory of the program, and effectively constrain the hypotheses available in a situation. Note that for each abducible a consider-rule is added automatically into the EP program.

Handling preferences over abductive logic programs has several advantages, and allows for easier and more concise translation into NLPs than those prescribed by more general and complex rule preference frameworks. The advantages of so proceeding stem largely from avoiding combinatory explosions of abductive solutions, by filtering irrelevant as well as less preferred abducibles (Pereira, et al., 2012).

To express preference criteria among abducibles, we envisage an extended language $\mathbf{L}^*$. A preference atom in $\mathbf{L}^*$ is of the form $a <| b$, where a and b are abducibles. It means that if b can be assumed (i.e. considered), then $b <| a$ forces a to be considered too if it can. A preference rule in $\mathbf{L}^*$ is of the form:

$$a <| b \leftarrow L_1, \ldots,\ L_t\ (t \geq 0)$$

where $L_1, \ldots,\ L_t$ are domain literals over $\mathbf{L}^*$.

A priori preferences are used to produce the most interesting or relevant conjectures about possible future states. They are taken into account when generating possible scenarios (abductive solutions), which will subsequently be preferred amongst each other a posteriori.

Example (Choose tea or coffee) Consider a situation where I need to choose to drink either tea or coffee (but not both). I prefer coffee to tea when sleepy, and do not drink coffee when I have high blood pressure. This situation can be described with the following EP program, including two abducibles *coffee* and *tee*:

```
abds [tea/0 , coffee/0].
on_observe(drink).
drink ← tea.
drink ← coffee.
← tea, coffee.
expect(tea ).            expect(coffee ).
expect_not(coffee) ← blood_high_pressure.
coffee <| tea ← sleepy.
```

This program has two abductive solutions, one with *tea* and the other with *coffee*. Adding literal *sleepy* triggers the only *a priori* preference in the program, which defeats the solution where only *tea* is present (due to the impossibility of simultaneously abducing coffee). If later we add blood pressure high, *coffee* is no longer expected, and the transformed preference rule no longer defeats the abduction of *tea*, which then becomes the single abductive solution, despite the presence of *sleepy*.

A Posteriori Preferences Having computed possible scenarios, represented by abductive solutions, more favorable scenarios can be preferred a posteriori. Typically, *a posteriori* preferences are performed by evaluating consequences of abducibles in abductive solutions. An *a posteriori* preference has the form:

$$A_i \ll A_j \leftarrow holds\_given(L_i, A_i), holds\_given(L_j , A_j),$$

where A_i, A_j are abductive solutions and L_i, L_j are domain literals. This means that A_i is preferred to A_j a posteriori if L_i and L_j are true as the side effects of abductive solutions A_i and A_j, respectively, without any further abduction when testing for the side effects. Optionally, in the body of the preference rule there can be any Prolog predicate used to quantitatively compare the consequences of the two abductive solutions.

Evolution Result A Posteriori Preference While looking ahead a number of steps into the future, the agent is confronted with the problem of having several different possible courses of evolution. It needs to be able to prefer amongst them to determine the best courses from its present state (and any state in general). The a posteriori preferences are no longer appropriate, since they can be used to evaluate only one-step-far consequences of a commitment. The agent should be able to also declaratively specify preference amongst evolutions through quantitatively or qualitatively evaluating the consequences or side effects of each evolution choice.

A posteriori preference is generalized to prefer between two evolutions. An *evolution result a posteriori* preference is performed by evaluating consequences of following some evolutions. The agent must use the imagination (look-ahead capability) and present knowledge to evaluate the consequences of evolving according to a particular course of evolution. An evolution result a posteriori preference rule has the form:

$$E_i <<< E_j \leftarrow holds\_in\_evol(L_i, E_i), holds\_in\_evol(L_j , E_j),$$

where E_i , E_j are possible evolutions and L_i, L_j are domain literals. This preference implies that E_i is preferred to *Ej* if *Li* and L_j are true as evolution history side effects when evolving according to E_i or $\boldsymbol{E_j}$, respectively, without making further abductions when just checking for the side effects. Optionally, in the body of the preference rule there can be recourse to any Prolog predicate, used to quantitatively compare the consequences of the two evolutions for decision making.

Intention Recognition

We describe our previously implemented intention recognition system, which operates upon Bayesian Network (BN) inference (Pereira & Han, 2009c, 2011b). To begin with, we provide some basic definitions regarding BNs needed for further understanding of the system.

Bayesian Networks

Definition 2 (Bayesian Network): A Bayesian Network (BN) is a pair consisting of a *directed acyclic graph* (DAG) whose nodes represent variables and missing edges encode conditional independencies between the variables, and an associated probability distribution satisfying the *Markov assumption* of conditional independence, saying that variables are independent of non-descendants given their parents in the graph (Pearl, 1988, 2000).

In a BN, associated with each node of its DAG is a specification of the distribution of its variable, say A, conditioned on its parents in the graph (denoted by $pa(A)$)—i.e., $P(A|pa(A))$ is specified. If $pa(A)$ is empty (A is called root node), its unconditional probability distribution, $P(A)$, is specified. These distributions are called *Conditional Probability Distribution* (CPD) of the BN.

The joint distribution of all node values can be determined as the product of conditional probabilities of the value of each node on its parents

$$P(X1, ..., XN) = \prod_{i=1}^{N} P(Xi|pa(Xi)) \qquad (1)$$

where $V = \{X_1, ..., X_N\}$. is the set of nodes of the DAG. Suppose there is a set of evidence nodes (i.e. their values are observed) in the DAG, say $O = \{O_1, ..., O_N\} \subset V$. We can determine the conditional probability distribution of a variable X given the observed value of evidence nodes by using the conditional probability formula

$$P(X|O) = \frac{P(X,O)}{P(O)} = \frac{P(X, O1, \ldots, Om)}{P(O1, \ldots, Om)} \qquad (2)$$

where the numerator and denominator are computed by summing the joint probabilities over all absent variables with respect to V as follows

$$P(X = x, O = o) = \sum_{av \in ASG(AV1)} P(X = x, O = o, AV1 = av)$$
$$P(O = o) = \sum_{av \in ASG(AV2)} P(O = o, AV2 = av)$$

where $o = \{o_1, ..., o_m\}$ with $o_1, ..., o_m$ being the observed values of $O_1, ..., O_m$, respectively; $ASG(V_t)$ denotes the set of all assignments of vector V_t (with components are variables in V); AV_1, AV_2 are vectors components of which are corresponding absent variables, i.e. variables in $V \backslash (O \cup \{X\})$ and $V \backslash O$, respectively.

Bayesian Networks for Intention Recognition

In (Pereira & Han, 2009c, 2011b), a general BN model for intention recognition is presented and justified based on Heinze's *causal intentional model* (Heinze, 2003; Tahboub, 2006). Basically, the BN consists of three layers: *cause/reason* nodes in the first layer (called *pre-intentional*), connecting to *intention* nodes in the second one (called *intentional*), in turn connecting to *action* nodes in the third (called *activity*) (Figure 1).

Fig. 1: ***General structure of a Bayesian network for intention recognition***. The Bayesian network consists of three layers. The pre-intentional layer consists of cause/reason nodes, connecting to intention nodes in the intentional layer, which in turn connect to action nodes in the activity layer.

In general, intention recognition consists in computing the probabilities of each conceivable intention conditional on the current observations, including the observed actions in the third layer, and some of the causes/reasons in the first layer. The prediction of what is the intention of the observed agent can simply be the intention with the greatest conditional probability, possibly above some minimum threshold. Sometimes it is also useful to predict what are the N ($N \geq 2$) most likely intentions given the current observations (Armentano & Amandi, 2009; Blaylock & Allen, 2003; Han & Pereira, 2011a).

Example 2 (Fox-Crow): Consider Fox-Crow story, adapted from Aesop's fable (Aesop). There is a crow, holding a cheese. A fox, being hungry, approaches the crow and praises her, hoping that the crow will sing and the cheese will fall down near him. Unfortunately for the fox, the crow is very intelligent, having the ability of intention recognition.

Fig. 2: Bayesian Network for Fox's intention recognition

The BN for recognizing Fox's intention is depicted in the Figure 2. The initial possible intentions of Fox that Crow comes up with are: Food - *i(F)*, Please - *i(P)* and Territory - *i(T)*. The facts that might give rise to those intentions are how friendly the Fox is (*Friendly_fox*) and how hungry he is (*Hungry_fox*). These figure in the first layer of the BN as the causes/reasons of the intention nodes. Currently, there is only one observation, which is, Fox praised Crow (*Praised*).

In this work, Bayesian Network inference will be performed using P-log, a probabilistic logic system, described in the next section. This will not only allow us to effectively represent the causal relations present a BN for intention recognition, the logic-based implementation of P-log will also allow us to make an easy integration with the EP system.

P-log

The P-log system in its original form (Baral, et al., 2009) uses Answer Set Programming (ASP) as a tool for computing all stable models (Baral, 2003; Gelfond & Lifschitz, 1993) of the logical part of P-log. Although ASP has proven a useful paradigm for solving a variety of combinatorial problems, its non-relevance property (Castro, Swift, & Warren, 2007) makes the P-log system sometimes computationally redundant. A new implementation of P-log (Han, et al., 2008; Han, Carroline, & Damasio, 2009), which we deploy in this work, uses the XASP package of XSB Prolog (XSB, 2009) for interfacing with Smodels (Niemela & Simons, 1997), an answer set solver. The power of ASP allows the representation of both classical and default negation, to produce 2-valued models. Moreover, using XSB as the underlying processing platform enables collecting the relevant abducibles for a query, obtained by need with top-down search. Furthermore, XSB permits to embed arbitrary Prolog code for recursive definitions. Consequently, it allows more expressive queries not supported in the original version, such as meta-queries (probabilistic built-in predicates can be used as usual XSB predicates, thus allowing the full power of probabilistic reasoning in XSB) and queries in the form of any XSB predicate expression (Han, et al., 2008). In addition, the tabling mechanism of XSB (Swift, 1999) significantly improves the performance of the system.

In general, a P-log program Π consists of a sorted signature, declarations, a regular part, a set of random selection rules, a probabilistic information part, and a set of observations and actions.

Sorted signature and Declaration: The sorted signature Σ of Π contains a set of constant symbols and term-building function symbols, which are used to form terms in the usual way. Additionally, the signature contains a collection of special function symbols called attributes. Attribute terms are expressions of the form *a(t)*, where *a* is an attribute and *t* is a vector of terms of the sorts required by a. A literal is an atomic expression, *p*, or its explicit negation, *neg_p*.

The declaration part of a P-log program can be defined as a collection of sorts and sort declarations of attributes. A sort *c* can be defined by listing all the elements $c = \{x_1, \ldots, x_m\}$ or by specifying the range of values $c = \{L..U\}$, where *L* and *U* are the integer lower bound and upper bound of the sort *c*. Attribute *a* with domain $\boldsymbol{c_1} \times \ldots \times \boldsymbol{c_n}$ and range $\boldsymbol{c_0}$ is represented as follows:

$a{:}\boldsymbol{c_1} \times \ldots \times \boldsymbol{c_n} \text{-->} \boldsymbol{c_0}$

If attribute a has no domain parameter, we simply write $a : \boldsymbol{c_0}$. The range of attribute a is denoted by *range(a)*.

Regular part: This part of a P-log program consists of a collection of XSB Prolog rules, facts and integrity constraints (IC) formed using literals of Σ. An IC is encoded as a XSB rule with the false literal in the head.

Random Selection Rule: This is a rule for attribute a having the form:

random(RandomName,a(t),DynamicRange) :- Body.

This means that the attribute instance *a(t)* is random if the conditions in *Body* are satisfied. The *DynamicRange* allows us to restrict the default range for random attributes. The *RandomName* is a syntactic mechanism used to link random attributes to the corresponding probabilities. A constant *full* can be used in *DynamicRange* to signal that the dynamic range is equal to *range(a)*.

Probabilistic Information: Information about probabilities of random attribute instances *a(t)* taking a particular value *y* is given by probability atoms (or simply pa-atoms) which have the following form:

pa(RandomName, a(t,y), d (A,B)):- Body,

meaning that if the *Body* were true, and the value of *a(t)* were selected by a rule named *RandomName*, then *Body* would cause *a(t) = y* with probability *A/B*. Note that the probability of an atom *a(t,y)* will be directly assigned if the corresponding *pa/3* atom is the head of some *pa*-rule with a true body. To define probabilities of the remaining atoms we assume that, by default, all values of a given attribute, which are not assigned a probability, are equally likely.

Observations and Actions: These are, respectively, statements of the forms *obs(l)* and *do(l)*, where *l* is a literal. Observations *obs(a(t,y))* are used to record the outcomes *y* of random events *a(t)*, i.e. random attributes and attributes dependent on them. Statement *do(a(t,y))* indicates *a(t) = y* is enforced as the result of a deliberate action.

In an EP program, P-log code is embedded by putting it between two reserved keywords *beginPlog* and *endPlog*. In P-log, probabilistic information can be obtained using the XSB Prolog built-in predicate *pr/2* (Han, et al., 2008). Its first argument is the query, the probability of which is needed to compute. The second argument captures the result. Thus, probabilistic information can be easily embedded by using *pr/2* like a usual Prolog predicate, in any constructs of EP programs, including active goals, preferences, and integrity constraints. What is more, since P-log (Han, et al., 2008) allows us to code Prolog probabilistic meta-predicates (Prolog predicates that depend on *pr/2* predicates), we also can directly use probabilistic meta-information in EP programs. We will illustrate those features with several examples below.

Example 3 (Fox-Crow) The BN for Fox's intention recognition (Figure 2) can be coded with the following P-log program.

1. bool = {t,f}. fox_intentions = {food,please,territory}.
2. hungry_fox : bool.   friendly_fox : bool.

```
 i : fox_intentions --> bool.       praised : bool.
3. random(rh, hungry_fox, full).   random(rf, friendly_fox, full).
   random(ri, i(I), full).          random(rp, praised, full).
4. pa(rh,hungry_fox(t),d_(1,2)).
   pa(rf,friendly_fox(t),d_(1,100)).
5. pa(ri(food),i(food,t),d_(8,10)) :- friendly_fox(t),hungry_fox(t).
   pa(ri(food),i(food,t),d_(9,10)) :- friendly_fox(f),hungry_fox(t).
   pa(ri(food),i(food,t),d_(0.1,10)) :- friendly_fox(t),hungry_fox(f).
   pa(ri(food),i(food,t),d_(2,10)) :-
   friendly_fox(f),hungry_fox(f).
6. pa(ri(please),i(please,t),d_(7,10)) :- friendly_fox(t),hungry_fox(t).
   pa(ri(please),i(please,t),d_(1,100)) :- friendly_fox(f),hungry_fox(t).
   pa(ri(please),i(please,t),d_(95,100)) :- friendly_fox(t),hungry_fox(f).
   pa(ri(please),i(please,t),d_(5,100)) :- friendly_fox(f),hungry_fox(f).
7. pa(ri(territory),i(territory,t),d_(1,10)) :- friendly_fox(t).
   pa(ri(territory),i(territory,t),d_(9,10)) :- friendly_fox(f).
8. pa(rp, praised(t),d_(95,100)) :- i(food, t), i(please, t).
   pa(rp, praised(t),d_(6,10))     :- i(food, t), i(please, f).
   pa(rp, praised(t),d_(8,10))     :- i(food, f), i(please, t).
   pa(rp, praised(t),d_(1,100))   :- i(food, f), i(please,f), i(territory,t).
   pa(rp, praised(t),d_(1,1000)) :- i(food,f), i(please,f), i(territory,f).
```

Two sorts *bool* and *fox_intentions*, in order to represent Boolean values and the current set of Fox's conceivable intentions, are declared in part 1. Part 2 is the declaration of four attributes *hungry_fox*, *friendly_fox*, *praised* and *i*, which state the first three attributes have no domain parameter and get Boolean values, and the last one maps each Fox's intention to a Boolean value. The random selection rules in part 3 declare that these four attributes are randomly distributed in their ranges. The distributions of the top nodes (*hungry_fox*, *friendly_fox*) and the CPD corresponding to the BN in Figure 2 are given in part 4 and parts 5-8, respectively, using the probabilistic information *pa*-rules. For example, in part 4 the first rule says that fox is hungry with probability *1/2* and the second rule says he is friendly with probability *1/100*. The first rule in part 5 states that if Fox is friendly and hungry, the probability of him having intention Food is *8/10*.

Note that the probability of an atom *a(t,y)* will be directly assigned if the corresponding *pa/3* atom is in the head of some *pa*-rule with a true body. To define probabilities of the remaining atoms we assume that by default, all values of *a* given attribute which are not assigned a probability are equally likely. For example, the first rule in part 4 implies that fox is not hungry with probability *1/2*. And, actually, we can remove that rule without changing the probabilistic information since, in that case, the probability of fox being hungry and of not being hungry are both defined by default, thus, equal to *1/2*.

The probabilities of Fox having intention *Food*, *Territory* and *Please* given the observation that Fox praised Crow can be found in P-log with the following queries, respectively:

?− pr(i(food,t) '|' obs(praised(t)),V_1). The answer is: $V_1 = 0.9317$.
?− pr(i(territory,t) '|' obs(praised(t)),V_2). The answer is: $V_2 = 0.8836$.
?− pr(i(please,t) '|' obs(praised(t)),V_3). The answer is: $V_3 = 0.0900$.

From the result we can say that Fox is most likely to have the intention of deceiving the Crow for food, *i(food)*.

Intention-based Decision Making

There are several ways an EP agent can benefit from the ability to recognize intentions of other agents, both in friendly and hostile settings. Knowing the intention of an agent is a means to predict what he will do next or might have done before. The recognizing agent can then plan in advance to take the best advantage of the prediction, or act to take remedial action. Technically, in the EP system, this new kind of knowledge may impinge on the body of any EP constructs, such as active goals, expectation and counter-expectation rules, preference rules, integrity constraints, etc., providing a new kind of trigger.

In order to account for intentions of other agents in decision making with EP, we provide a built-in predicate, *has_intention(Ag,I)*, stating that an agent *Ag* has the intention *I*. The truth-value of this predicate is evaluated by the intention recognition system. Whenever this predicate is called in an EP program, the intention recognition system is employed to check if *Ag* has intention *I*, i.e. *I* is the most likely conceivable intention at that moment. We also provide predicate *has_intention(Ag,I,Pr)*, stating that agent *Ag* has intention *I* with probability *Pr*. Hence, one can express, for example, the situation where one needs to be more, or less, cautious.

One can also generalize to consider the *N*-best intention recognition approach, that is, to assess whether the intention of the agent is amongst the *N* most likely intentions. It has been shown that by increasing *N*, the recognition accuracy is significantly improved (Armentano & Amandi, 2009; Blaylock & Allen, 2003; Han & Pereira, 2011a).

In the sequel we draw closer attention to some EP constructs, illustrating with several examples how to take into account intentions of other agents for enhancement of decision making.

Intentions Triggering Active Goals

Recall that an active goal has the form

$$on\_observe(AG) \leftarrow L_1, \ldots,\ L_t\, (t \geq 0),$$

where $L_1, \ldots,\ L_t$ are domain literals. At the beginning of each cycle of evolution, those literals are checked with respect to the current evolving knowledge base and trigger the active goal if they all hold. For intention triggering active goals, the domain literals in the body can be in the form of has intention predicates, taking into account intentions of other agents.

This way, any intention recognition system can be used as the goal producer for decision making systems, the inputs of which are (active) goals to be solved (see for instance (Han & Pereira, 2011b; Pereira & Han, 2011a, 2011b)).

It is easily seen that intention triggering active goals are ubiquitous. New goals often appear when one recognizes some intentions in others. In a friendly setting, one might want to help others to achieve their intention, which is generally represented as follows

$$on\_observe(help\_achieve\_goal(G)) \leftarrow friend(P),\ has\_intention(P,G),$$

while in a hostile setting, we probably want to prevent the opponents from achieving their goals

$$on\_observe(prevent\_achieve\_goal(G)) \leftarrow opponent(P),\ has\_intention(P,G).$$

Or, perhaps we simply want to plan in advance to take advantage of the hypothetical future obtained when the intending agent employs the plan that achieves his intention

on_observe(take_advantage(F)) ← agent(P), has_intention(P,G), future(employ(G),F).

Let us look a little closer at each setting, providing some ideas how they can be enacted. When helping someone to achieve an intention, what we need to do is to help him/her with executing a plan achieving that intention successfully, i.e., all the actions involved in that plan can be executed. This usually occurs in multi-agent collaborative tasks (see for example (Kaminka, et al., 2002)), wherein the agents need to be able to recognize their partners' intention to secure an efficient collaboration.

In contrast, in order to prevent an intention from being achieved, we need to guarantee that any conceivable plans achieving that intention cannot be executed successfully. To that effect, at least one action in each plan must be prevented if the plan is conformant (i.e., a sequence of actions (Phan Huy Tu, Son, Gelfond, & Morales, 2011)). If the plan is conditional (see for (Pereira & Han, 2009c; P. H. Tu, Son, & Baral, 2007)), each branch is considered a conformant plan and must be prevented.

We shall exhibit a diversity of examples in the following sections.

Intention Triggering Preferences

Having recognized an intention of another agent, the recognizing agent may either favor or disfavor an abducible (*a priori* preferences), an abductive solution (*a posteriori* preferences) or an evolution (*evolution result a posteriori* preferences) with respect to another, respectively; depending on the setting they are in. If they are in a friendly setting, the one that provides more support to achieve the intention is more favored; in contrast, in a hostile setting, the one providing more support is disfavored. The recognizing agent may also favor the one that takes better advantage of the recognized intention.

To illustrate the usage of intention triggering a priori preferences, we revise here Example 1.

Example 4 (Choose tea or coffee taking into account a friend's intentions) Being thirsty, I consider making tea or coffee. I realize that my roommate, John, also wants to have a drink. To be friendly, I want to take into account his intention when making my choice. This scenario is represented with the following EP program.

1. abds([coffee/0, tea/0]).
2. expect(coffee).   expect(tea).
3. on_observed(drink) ← thirsty.
   drink ← tea.
   drink ← coffee.
   ← tea, coffee.
4. expect_not(coffee) ← blood_high_pressure.
5. tea   <| coffee   ← **has_intention**(john,tea).
   coffee <| tea   ← **has_intention**(john,coffee).

It is enacted by the preference rules in part 5. The first rule says that *tea* is preferable, a priori, to *coffee* if John intends to drink *tea*; and vice versa, the second rule says that if John intends to drink *coffee*, *coffee* is preferable. Note that the recognition of what John intends is performed by the intention recognition system—which is triggered when a reserved predicate *has_intention/2* is called.

This scenario also can be encoded using intention triggering a posteriori preferences. As a good friend of

John, I prefer an abductive solution with a side effect of John being happy to the one with a side effect of John being unhappy. This can be coded as follows.

unhappy ← coffee, **has_intention**(john, tea).
happy ← coffee, **has_intention**(john, coffee).
unhappy ← tea, **has_intention**(john, coffee).
unhappy ← tea, **has_intention**(john, tea).
$A_i << A_j$ ← holds_given(happy, A_i),
holds_given(unhappy, A_j).

Despite its simplicity, the example demonstrates how to solve a class of collaborative situations, where one would like to take into account the intentions and the need of others when deriving relevant hypothetical solutions of our current goals.

Next, to illustrate other kinds of preferences, we consider the following revised extended version of the saving city example, presented in (Pereira & Han, 2009b).

Example 5 (Saving cities by means of intention recognition) During war time, agent David, a general, needs to decide to save a city from his enemy's attack or leave it to keep the military resource, which might be important for some future purpose. David has recognized that a third party is intending to make an attack to the enemy on the next day. David will have a good chance to defeat the enemy if he has enough military resource to coordinate with the third party. The described scenario is coded with the following EP program.

1. abds([save/0, leave/0]).
2. expect(save). expect(leave).
3. on_observe(choose) ← **has_intention**(enemy,attack_my_city).
   choose ← save.
   choose ← leave.
4. save_men(5000) ← save. save_men(0) ← leave.
   lose_resource ← save. save_resource ← leave.
5. $A_i << A_j$ ← holds_given(save_men(N_i), A_i),
   holds_given(save_men(N_j), A_j), $N_i > N_j$.

6. on_observe(decide) ← decide_strategy.
   decide ← stay_still.
   decide ← counter_attack.
7. good_opportunity ← **has_intention**(third_party,attack).
   expect(counter_attack) ← good_opportunity, save_resource.
   expect(stay_still).
8. pr(win,0.9) ← counter_attack.
   pr(win,0.01) ← stay_still.
9. Ei <<< Ej ← holds_in_evol(pr(win,Pi), E_i),
   holds_in_evol(pr(win,Pj), E_j), $P_i > P_j$.

In the first cycle of evolution, there are two abducibles, *save* and *leave*, declared in part 1, to solve the active goal *choose*. The active goal is triggered when David recognizes the intention of the enemy to attack his city (part 3).

Similar to the original version in (Pereira & Han, 2009b), in the case of being a bad general who just sees the situation at hand, David would choose to save the city since it would save more people (5000 vs. 0, part 4), i.e. the *a posteriori* preference in part 5 is taken into account immediately, to rule out the case of leaving the city since it would save less people. Then, next day, he would not be able to attack since the military resource is not saved (part 7), and that leads to the outcome with very small probability of winning the whole war (part 8).

But, fortunately, being able to look ahead plus, being capable of intention recognition, David can see that on the next day, if he has enough military resources, he will have a good opportunity to make a counter-attack on his enemy (part 7), by coordinating with a third party who exhibits the intention to attack the enemy on that day as well; and a successful counter-attack would lead to a very much higher probability of winning the conflict as a whole (part 8). The *evolution result a posteriori* preference is employed in part 9 to prefer the evolution with higher probability of winning the whole conflict.

In this example we can see, in part 7, how a detected intention of another agent can be used to enhance the decision making process. It is achieved by providing an (indirect) trigger for an abducible expectation —thereby enabling a new opportunistic solution by means of coordinating with others —which affects the final outcome of the *evolution result a posteriori* preference in part 9.

Hostile Setting

In this hostile setting, having confirmed the intention (and possibly also the plans achieving that intention being carried out by the intending agent), the recognizing agent might act to prevent the intention from being achieved, that is, prevent at least one action of each intention achieving plan from being successfully executed; and, in case of impossibility to doing so, act to minimize losses as much as possible.

Example 6 (Fox-Crow, cont'd) Suppose in Example 2, the final confirmed Fox's intention is that of getting food (additional details can be found in (Pereira & Han, 2009c)). That is, the predicate *has_intention(fox,food)* holds. Having recognized Fox's intention, what should Crow do to prevent Fox from achieving it? The following EP program helps Crow with that.

1. abds([decline/0, sing/0, hide/2, eat/2, has_food/0, find_new_food/0]).
2. expect(decline).   expect(sing).
   expect(hide(_,_)).   expect(eat(_,_)).
3. on_observe(not_losing_cheese) ← **has_intention**(fox, food).
   not_losing_cheese ← decline.
   not_losing_cheese ← hide(crow,cheese), sing.
   not_losing_cheese ← eat(crow,cheese), sing.
4. expect_not(eat(A,cheese)) ← animal(A), full(A).
   animal(crow).
5. ← decline, sing.
   ← hide(crow,cheese), eat(crow,cheese).
6. eat(crow,cheese) <| hide(crow,cheese).
7. no_pleasure ← decline.
   has_pleasure ← sing.
8. A_i<< A_j ← holds_given(has_pleasure,A_i),
   holds_given(no_pleasure,A_j).

9. on_observe(feed_children) ← hungry(children).
   feed_children ← has_food.
   feed_children ← find_new_food.
   ← has_food, find_new_food.
10. expect(has_food) ← decline, not eat(crow,cheese).

expect(has_food) $\leftarrow$ hide(crow,cheese), not stolen(cheese).
expect(find_new_food).

11. $E_i <<< E_j \leftarrow$ hungry(children), holds_in_evol(has_food,E_i),
    holds_in_evol(find_new_food,E_j).
12. $E_i <<< E_j \leftarrow$ holds_in_evol(has_pleasure,E_i),
    holds_in_evol(no_pleasure,E_j).

There are two possible ways so as not to lose the *Food* to Fox, either simply *decline to sing* (but thereby missing the pleasure of singing) or *hide or eat the cheese* before *singing*.

Part 1 is the declaration of program abducibles (the last two abducibles are for the usage in the second phase, starting from part 9). All of them are always expected (part 2). The counter-expectation rule in part 4 states that an animal is not expected to eat if he is full. The integrity constraints in part 5 say that Crow cannot decline to sing and sing, hide and eat the cheese, at the same time. The *a priori* preference in part 6 states that eating the cheese is always preferred to hiding it (since it may be stolen), of course, just in case eating is a possible solution.

Suppose Crow is not full. Then, the counter-expectation in part 4 does not hold. Thus, there are two possible abductive solutions: *[decline]* and *[eat(crow,cheese), sing]* (since the *a priori* preference prevents the choice containing *hiding*).

Next, the *a posteriori* preference in part 8 is taken into account and rules out the abductive solution containing *decline* since it leads to having *no pleasure* which is less preferred to *has pleasure* —the consequence of the second solution that contains *sing* (part 7). In short, the final solution is that Crow eats the cheese then sings, without losing the cheese to Fox and having the pleasure of singing.

Now, let us consider a smarter Crow who is capable of looking further ahead into the future in order to solve longer-term goals. Suppose that Crow knows that her children will be hungry later on, in the next stage of evolution (part 9); eating the cheese right now would make her have to find new food for the hungry children. Finding new food may take long, and is always less favorable than having food ready to feed them right away (cf. the *evolution result a posteriori* preference in part 11). Crow can see three possible evolutions: *[[decline], [has_food]]; [[hide(crow, cheese), sing], [has_food]]* and *[[eat(crow, cheese), sing], [find_new_food]]*. Note that in looking ahead at least two steps into the future, *a posteriori* preferences are taken into account only after all evolution-level ones have been applied (Pereira & Han, 2009b).

Now the two *evolution result a posteriori* preferences in parts 11-12 are taken into account. The first one rules out the evolution including *finding new food* since it is less preferred than the other two, which includes *has_food*. The second one rules out the one including *decline*. In short, Crow will hide the food to keep it for her hungry children, and still take pleasure from singing.

In short, we have seen several extended examples illustrating diverse ways in which accounting for intentions of others might, in a simple manner, significantly enhance the final outcome of a decision situation. In the next sections we pay attention to concrete application domains, wherein we address issues on which intention-based decision making may enable improvement, and show how to tackle them using our described logic-based framework. Namely, in more technological based application domains, those regarding Ambient Intelligence in the home environment and regarding Elder Care will be studied in the next section. Then, in Section 5, more experimental based domains, those of moral reasoning and game theory, are given attention.

Ambient Intelligence in the Home Environment and Elder Care

Ambient Intelligence (*AmI*) is the vision of a future in which environments support people inhabiting in them. The envisaged environment is unobtrusive, interconnected, adaptable, dynamic, embedded and intelligent. It should be sensitive to the needs of inhabitants, and capable of anticipating their needs and behavior. It should be aware of their personal requirements and preferences, and interact with people in a user-friendly way (see a comprehensive survey in (Sadri, 2011a)).

One of the key issues of Ambient Intelligence, which has not been well studied yet, and reported as an ongoing challenge (Cook, Augusto, & Jakkula, 2009), is that AmI systems need to be aware of users' preferences, intentions and needs. Undoubtedly too, respecting users' preferences and needs in decision making processes would increase their degree of acceptance with respect to the systems, making these deemed more friendly and thoughtful.

From this perspective on AmI, we can see a number of issues where intention recognition techniques can step in, providing help and enabling improvement. For example, in order to provide appropriate support, the environment should be able to proactively recognize the inhabitants' intention —to glean whether they need help to accomplish what they intend to do —or to warn them (or their carers) in case they intend something inappropriate or even dangerous.

Undoubtedly, an ability to recognize intentions of assisted people, as well as other relevant concerns such as intruders or the like, would enable to deal with a combination of several issues, encompassing those of pro-activeness (either agonistic or antagonistic), security, and emergency, in a much more integrated and timely manner (Han & Pereira, 2010a, 2010b; P. Roy, Bouchard, Bouzouane, & Giroux, 2007). We discuss these very issues in the sequel.

Proactive Support

An important feature of AmI, particularly desirable in the Elder Care domain, is that the assisting system should take initiative to help the people it assists. To this end, the system must be capable of recognizing their intentions on the basis of their observable actions, then provide suggestions or help achieve the recognized intentions (Pereira & Han, 2011a, 2011b). A suggestion can be, for example, what are the appropriate kinds of drink for the elder, considering the current time, temperature, or even future scheduled events such as going to have a medical test on the next day, upon having recognized that he has an intention to drink something. Or, a suggestion can simply be telling the elder where he put his book yesterday, having recognized that he might be looking for it. This feature is especially desirable and important when the assisted people are elderly or individuals with disabilities or suffering from mental difficulties (P. Roy, et al., 2007). The need for technology in this area is obvious looking at the fact that in the last twenty years there has been a significant increase of the average age of the population in most western countries and the number of elderly people has been and will be constantly growing (Cesta & Pecora, 2004; Cook, et al., 2009; Geib, 2002; Giuliani, et al., 2005; Haigh, et al., 2004; Han & Pereira, 2010a; Pereira & Han, 2011a; P. Roy, et al., 2007; Sadri, 2008).

The EP system can be engaged to provide appropriate suggestions for the elders, taking into account the external environment, elders' preferences and already scheduled future events. Expectation rules and *a priori* preferences cater for the physical state information (health reports) of the elders, in order to guarantee that only contextually safe healthy choices are generated; subsequently, information such as the elders' pleasure and interests are then considered by *a posteriori* preferences and the like.

In the Elder Care domain, assisting systems should be able to provide contextually appropriate suggestions for the elders based on their recognized intentions. The assisting system is supposed to be better aware of the environment, the elders' physical states, mental states as well as their scheduled events, so that it can provide good and safe suggestions, or simply warnings.

Let us consider the following simple scenario in the Elder Care domain.

Example 7 (Elder Care) An elder stays alone in his apartment. The intention recognition system observes that he is looking for something in the living room. In order to assist him, the system needs to figure out what he intends to find. The possible things are: something to read (*book*); something to drink (*drink*); the TV remote control (*Rem*); and the light switch (*Switch*). The BN for recognizing the elder's intention, with CPD and top nodes distribution, is given in Figure 3.

Fig. 3: Bayesian Network for recognizing the elder's intentions

Similarly to the P-log representation and inference in Example 3, the probabilities that the elder has the intention of looking for *book*, *drink*, *remote control* and *light switch* given the observations that he is looking around and of the *states of the light* (on or off) and *TV* (on or off) can be obtained with the following queries, respectively:

? − pr(i(book, t) | (obs(tv(S1)) & obs(light(S2)) & obs(look(t))), V_1).
? − pr(i(drink, t) | (obs(tv(S1)) & obs(light(S2)) & obs(look(t))), V_2).
? − pr(i(rem, t) | (obs(tv(S1)) & obs(light(S2)) & obs(look(t))), V_3).
? − pr(i(switch, t) | (obs(tv(S1)) & obs(light(S2)) & obs(look(t))), V_4).

where S_1, S_2 are Boolean values (*t* or *f*), to be instantiated during execution, depending on the states of the light and TV. Let us consider the possible cases:

- If the light is off (S_2 =f), then $V_1 = V_2 = V_3 = 0$, V_4 =1.0, regardless of the state of the TV.
- If the light is on and TV is off (S_1 = t, S_2 = f), then V_1 =0.7521, V_2 = 0.5465, V_3 = 0.5036, V_4 = 0.0101.
- If both light and TV are on (S_1 = t, S_2 = t), then V_1 = 0, V_2 = 0.6263, V_3 = 0.9279, V_4 = 0.0102.

Thus, if one observes that the light is off, definitely the elder is looking for the light switch, given that he is looking around. Otherwise, if one observes the light is on, in both cases where the TV is either on or off, the first three intentions *book, drink, remote control* still need to be put under consideration in the next phase, generating possible plans for each of them. The intention of looking for the light switch is very unlikely to be the case comparing with other three, thus being ruled out. When there is light one goes directly to the light switch if the intention is to turn it off, without having to look for it.

Example 8 (Elder Care, cont'd) Suppose in the above Elder Care scenario, the final confirmed intention is that of looking for a drink [4]. The possibilities are: *natural pure water, tea, coffee* and *juice*. The EP system now is employed to help the elder with choosing an appropriate drink. The scenario is coded with the EP program below.

[4] In general, from the design point of view, one needs to provide an EP program for each intention, because, according to context, a user might have or be predicted to have distinct intentions.

The elder's physical states are utilized in *a priori* preferences and expectation rules to guarantee that just choices that are contextually safe for the elder are generated. Only after that other aspects, for example the elder's pleasure with respect to each kind of drink, are taken into account, with the *a posteriori* preferences.

```
1. abds([water/0, coffee/0, tea/0, juice/0, precise_result/0, imprecise_result/0]).
2. expect(coffee).    expect(tea).
   expect(water).     expect(juice).
3. on_observe(drink)   ← has_intention(elder,drink).
   drink ← tea.    drink ← coffee.
   drink ← water.  drink ← juice.
4. expect_not(coffee) ← prolog(blood_high_pressure).
   expect_not(coffee) ← prolog(sleep_difficulty).
   expect_not(coffee) ← prolog(late).
   expect_not(juice)  ← prolog(late).
5.  ← tea, coffee.          ← coffee, juice.
    ← tea, juice.           ← tea, water.
6. coffee <| tea    ← prolog(morning_time).
   coffee <| water  ← prolog(morning_time).
   coffee <| juice  ← prolog(morning_time).
7. juice <| coffee ← prolog(hot).
   juice <| tea    ← prolog(hot).
   juice <| water  ← prolog(hot).
   water <| coffee ← prolog(hot).
   water <| tea    ← prolog(hot).
8. tea <| coffee ← prolog(cold).
   tea <| juice  ← prolog(cold).
   tea <| water  ← prolog(cold).
9. pleasure_level(3) ← coffee.     pleasure_level(2) ← tea.
   pleasure_level(1) ← juice.      pleasure_level(0) ← water.
10. sugar_level(1) ← coffee.       sugar_level(1) ← tea.
   sugar_level(5) ← juice.         sugar_level(0) ← water.
11.caffein_level(5) ← coffee.      caffein_level(0) ← tea.
   caffein_level(0) ← juice.       caffein_level(0) ← water.
12.A_i << A_j ← holds_given(pleasure_level(V_1), A_i),
                holds_given(pleasure_level(V_2), A_j), V_1 > V_2.

13. on_observe(health_check) ← time_for_health_check.
   health_check ← precise_result.
   health_check ← imprecise_result.
14. expect(precise_result) ← no_hight_sugar, no_high_caffein.
   expect(imprecise_result).
   no_high_sugar    ← sugar_level(L), prolog(L < 2).
   no_high_caffein  ← caffein_level(L), prolog(L < 2).
15.E_i <<< E_j ← holds_in_evol(precise_result, E_i ),
                 holds_in_evol(imprecise_result, E_j).

 beginProlog.
    : - assert(scheduled_events(1, [has_intention(elder,drink)])),
    assert(scheduled_events(2, [time_for_health_check])).
    late :- time(T), (T > 23; T < 5).
```

```
    morning_time :- time(T), T > 7, T < 10.
    hot  :- temperature(TM), TM > 32.
    cold :- temperature(TM), TM < 10.
    blood_high_pressure :- physical_state(blood_high_pressure).
    sleep_difficulty    :- physical_state(sleep_difficulty).
endProlog.
```

The information regarding the environment (current time, current temperature) and the physical states of the elder is coded in parts 9-11. The assisting system is supposed to be aware of this information in order to provide good suggestions.

Part 1 is the declaration of the program abducibles: *water, coffee, tea*, and *juice*. All of them in this case are always expected (part 2). Part 3 exhibits an intention triggering active goal: since the intention recognition module confirms that the elder's intention is to find something to drink, the EP system is triggered to seek appropriate suggestions for achieving the elder's intention. The counter-expectation rules in part 4 state that *coffee* is not expected if the elder has high blood pressure, experiences difficulty to sleep or it is late; and juice is not expected if it is late. Note that the reserved predicate *prolog/1* is used to allow embedding Prolog code, put between two built-in keywords, *beginProlog* and *endProlog*, in an EP program. More details can be found in (Han, 2009; Pereira & Han, 2009a, 2009b). The integrity constraints in part 5 say that it is not allowed to have at the same time the following pairs of drink: *tea* and *coffee*, *tea* and *juice*, *coffee* and *juice*, and *tea* and *water*. However, it is the case that the elder can have coffee or juice together with water at the same time.

The *a priori* preferences in part 6 say in the morning coffee is preferred to tea, water and juice. And if it is hot, juice is preferred to all other kinds of drink and water is preferred to tea and coffee (part 7). In addition, the *a priori* preferences in part 8 state if the weather is cold, tea is the most favorable, i.e. preferred to all other kinds of drink.

Now let us look at the suggestions provided by the Elder Care assisting system modeled by this EP program, considering some cases:

1. time(24) (*late*); temperature(16) (*not hot, not cold*); *no high blood pressure*; *no sleep difficulty*: there are two a priori abductive solutions: *[tea], [water]*. Final solution: *[tea]* (since it has greater level of pleasure than water, which is ruled out by the *a posteriori* preference in part 12.
2. time(8) (*morning time*); temperature(16) (*not hot, not cold*); *no high blood pressure*; *no sleep difficulty*: there are two abductive solutions: *[coffee]*, *[coffee, water]*. Final: *[coffee]*, *[coffee, water]*.
3. time(18) (*not late, not morning time*); temperature(16) (*not cold, not hot*); *no high blood pressure*; *no sleep difficulty*: there are six abductive solutions: *[coffee]*, *[coffee, water]*, *[juice]*, *[juice, water], [tea]*, and *[water]*. Final: *[coffee]*, *[coffee, water]*.
4. time(18) (*not late, not morning time*); temperature(16) (*not cold, not hot*); *high blood pressure*; *no sleep difficulty*: there are four abductive solutions: *[juice], [juice, water], [tea]*, and *[water]*. Final: *[tea]*.
5. time(18) (*not late, not morning time*); temperature(16) (*not cold, not hot*); *no high blood pressure*; *sleep difficulty*: there are four abductive solutions: *[juice]*, *[juice, water]*, *[tea]*, and *[water]*. Final: *[tea]*.
6. time(18) (*not late, not morning time*); temperature(8) (*cold*); *no high blood pressure*; *no sleep difficulty*: there is only one abductive solution: *[tea]*.
7. time(18) (*not late, not morning time*); temperature(35) (*hot*); *no high blood pressure*; *no sleep difficulty*: there are two abductive solutions: *[juice], [juice, water]*. Final: *[juice], [juice, water]*.

If the *evolution result a posteriori preference* in part 15 is taken into account and the elder is scheduled to go to the hospital for health check in the second day: the first and the second cases do not change. In the third case: the suggestions are *[tea]* and *[water]* since the ones that have *coffee* or *juice* would cause high caffeine and sugar levels, respectively, which can make the checking result (health) imprecise (parts 13-15). It can be done similarly for all the other cases.

Note future events can be asserted as Prolog code using the reserved predicate *scheduled_events/2*. For more details of its use see (Pereira & Han, 2009a, 2009b).

As one can gather, the suggestions provided by this assisting system are quite contextually appropriate. We might elaborate current factors (time, temperature, physical states) and even consider more factors to provide more appropriate suggestions if ever the situation gets more complicated.

Security and Emergency

Security in AmI. Security is one of the key issues for AmI success (Friedewald, et al., 2007), and particularly important in home environments (Friedewald, Costa, Punie, Alahuhta, & Heinonen, 2005). It comprises two important categories: security in terms of Burglary Alarm systems and security in terms of health and wellbeing of the residents (prevention, monitoring) (Friedewald, et al., 2005).

So far Burglary Alarm technology has been mainly based on sensing and recognizing the very last action of an intrusion plan, such as “breaking the door” (Friedewald, et al., 2005; Wikipedia). However, it may be too late to provide an appropriate protection. Burglary Alarm systems need to be able to guess in advance the possibility of an intrusion on the basis of the very first observable actions of potential intruders. For example, it would be useful to find out how likely a stranger constantly staring at your house has an intrusion intention, taking into account the particular situation, e.g. if he has weapon or if it is night time. This information can be sent to the carer, the assistive system, or the elders themselves (if there are no carers or assistive systems available), for them to get prepared (e.g. turn on the light or sounders to scare off burglars or call relatives, police, firemen, etc.). Our intention-based decision making system proves appropriate to deal with this scenario. Given any currently observed actions, the probability of the on-going conceivable intentions are computed, and if the one of the intrusion intention is large enough or is among (some of) the most likely intentions, the EP component should be informed of a potential intrusion, so as to make a timely decision, and issue suggestions to the elders. To be more certain about the possibility of an intrusion, additional observations may need to be made, but at least for now it is about ready to handle any potentially negative forthcoming situations. Waiting until being sure to get ready can be too late to take appropriate actions. For illustration, consider the next example.

Example 9 (Solving Intrusion) Envisage a situation where the intention recognition system recognized an intention of intrusion at night. The system must either warn the elders who are sleeping, automatically call the nearest police, or activate the embedded burglary alarm. If the elders are sleeping and ill, they do not expect to be warned, but prefer other solutions. Due to potential disturbance, the elders prefer simply activating the burglary system to calling the police as long as no weapon is detected and there is a single intruder.

The situation is described by the program with three abducibles: *call_police*, *warn_persons*, *activate_alarm*, and can be coded in EP as follows

1. on observe(solve intrusion) ← at night, **has_intention**(stranger, intrusion).
2. solve_intrusion ← call police.
   solve_intrusion ← warn persons.

solve_intrusion ← activate alarm.
3. expect(call police). expect(warn persons). expect(activate alarm).
4. expect_not(warn persons) ← ill, sleeping.
5. activate_alarms <| call police ← no_weapon_detected, individual.
6. call_police <| activate_alarms ← weapon_detected.

Suppose it is night-time and an intrusion intention is recognized, then the active goal solve intrusion (part 1) is triggered, and the EP system starts reasoning to find the most appropriate solutions.

This program has three abductive solutions*: [call_police], [warn_persons]*, and *[activate_alarm]* since all the abducibles are expected and there is no expectations to their contrary. Suppose it detects that the elders are sleeping and known to be ill, i.e. literals ill and sleeping are factual. In this case, the elders do not expect to be warned (part 4), thus ruling out the second solution *[warn_persons]*. And if no weapon is detected and there is only a single intruder, the a priori preference in part 5 is triggered, which defeats the solution where only call police is present (due to the impossibility of simultaneously abducing activate alarm). Hence, the only solution is to activate the burglary alarm. However, if weapons were detected, the preference in part 6 is triggered and defeats the [*activate_alarm*] solution. The only solution left is to call the police (*call_police*).

Regarding Burglary Alarm systems, in the following example we consider a simple scenario of recognizing an elder's intentions.

Example 10 (Detecting Intrusion) An elder stays alone in his apartment. One day the Burglary Alarm is ringing, and the assisting system observes that the elder is looking for something. In order to assist him, the system needs to figure out what he intends to find. Possible things are: Alarm button (*AlarmB*); Contact Device (*ContDev*), Defensible Weapons (*Weapon*), and light switch (*Switch*). The BN representing this scenario is in Figure 4.

Fig. 4: Bayesian Network for recognizing the elder's intentions in an intrusion situation

The nodes representing the conceivable intentions are: *i(AlarmB), i(ContDev), i(Weapon),* and *i(Switch).* The Bayesian network for intention recognition has three top nodes in the pre-intentional level, representing the causes or reasons of the intentions, which are *Alarm_On*, *Defensible* and *Light_on*. The first and last nodes are evidence nodes, i.e. their values are observable. There is only one observable action, represented by the node *Looking* in the last layer. It is a direct child of the intention nodes. The conditional probability tables (CPD) of each node in the BN are given. For example, the table of the node *Defensible* says that the elder is able to defense himself (with weapons) with probability of 0.3 and not able to do so with probability 0.7. The table in the top-right corner provides the probability of the elder looking around for something conditional on the intentions. Based on this BN one can now compute the conditional probability of each intention given the observed action.

Another security issue concerns health and well-being of the residents. AmI systems need to be able to prevent hazardous situations, which usually come from dangerous ideas or intentions (e.g. take a bath when drunk, drink alcohol while not permitted, or even commit suicide) of the assisted persons, especially those with mental impairments (P. Roy, et al., 2007). To this end, guessing their intentions from the very first relevant actions is indispensable to take timely actions. In our incremental intention recognition method, a BN will be built to compute how likely there is a dangerous intention, with respect to any currently observed actions, and carers would be informed in case it is likely enough, in order to get prepared in time.

Emergency in AmI. Handling emergency situations is another important issue in AmI. There are a wide range of emergency situations, e.g. in security, when recognizing intrusion intention of a stranger or dangerous intentions of the assisted person. They also can occur when detecting fire, unconsciousness or unusualness in regular activities (e.g. sleep for too long), etc. Emergency handling in the EP system can be done by having an active goal rule for whichever emergency situation. For solving the goal, a list of possible actions, all represented by abducible enablers, are available to form solutions. Then, users' preferences are encoded using all kinds of preference of EP: *a priori* ones for preferring amongst available actions, *a posteriori* ones for comparing solutions taking into account their consequences and utility, and *a posteriori evolution result* ones for comparing more-than-one-step consequences. Moreover, the expectation and counter expectation rules are used to encode pros and cons of the users towards each available action, or towards any abducible in general.

Discussion of Other AmI Issues. We have shown how our intention-based decision making framework can enable the provision of proactive support for assisted people, and the tackling of the AmI security and emergency issues. We now briefly sketch how it can be utilized to address yet other important issues in AmI.

First of all, it is known that intention recognition plays a central role in human communication (Heinze, 2003; Pinker, Nowak, & Lee, 2008; Tomasello, 2008). In addition, an important aspect of intentions is future-directedness, i.e. if we intend something now it means we intend to execute a course of actions to achieve it in the future (Bratman, 1987; O. Roy, 2009b; Singh, 1991). Most actions may be executed only at a far distance in time. Thus, we usually need to guess others' intentions from the very first clues, such as their actions or spoken sentences, in order to secure a smooth conversation or collaboration. Perhaps we guess a wrong intention, but we need to be able to react in a timely manner; and that is also part of the conversation. We can simply attempt to confirm by asking, e.g. "is this (...) what you mean?". Our intention-based decision making framework can be used to design better and more friendly human-computer interaction devices that can react to human behavior and speech, communicate with them to confirm their intentions so as to provide appropriate help when necessary, after having guessed their likely intentions using an intention recognition system.

Yet another issue is that, in order to be highly accepted by the users, an assistive system should be able to proffer explanations for the suggestions it provides. In EP, that can be easily done by keeping all the preferences, integrity constraints, expectation and counter expectation rules that were used both to consider and to rule out abductive solutions.

Other domains: Intention-based Decision Making in Moral Reasoning and Game Theory

Intention-based decision making in moral reasoning

A key factor in legal and moral judgments is intention (Hauser, 2007; Young & Saxe, 2011). Intent differentiates, for instance, murder from manslaughter. When making a moral decision, it is crucial to recognize if an action or decision is intentional (or at least very likely to be intentional so as, for instance, to be judged beyond reasonable double) or not. Intentionality plays the central part in different moral rules, notably the double effect principle (Hauser, 2007; Mikhail, 2007), rendered as follows:

> *Harming another individual is permissible if it is the foreseen consequence of an act that will lead to a greater good; in contrast, it is impermissible to harm someone else as an intended means to a greater good.*

This principle is particularly applicable for the well-known trolley problems, having the following initial circumstance (Hauser, 2007): "*There is a trolley and its conductor has fainted. The trolley is headed toward five people walking on the track. The banks of the track are so steep that they will not be able to get off the track in time*." Given this circumstance, there exist several cases of moral dilemmas (Mikhail, 2007). Let us consider the following three typical cases (illustrated in Figure 5).

Fig. 5: Three trolley cases: 1) Bystander; 2) Footbridge; 3) Loop Track

Bystander. Hank is standing next to a switch that can turn the trolley onto a sidetrack, thereby preventing it from killing the five people. However, there is a man standing on the sidetrack. Hank can throw the switch, killing him; or he can refrain from doing so, letting the five die. Is it morally permissible for Hank to throw the switch?

Footbridge. Ian is on the bridge over the trolley track, next to a heavy man, which he can shove onto the track in the path of the trolley to stop it, preventing the killing of five people. Ian can shove the man onto the track, resulting in death; or he can refrain from doing so, letting the five die. Is it morally permissible for Ian to shove the man?

Loop Track. Ned is standing next to a switch that can temporarily turn the trolley onto a sidetrack, without stopping, only to join the main track again. There is a heavy man on the sidetrack. If the trolley hits the man, he will slow down the trolley, giving time for the five to escape. Ned can throw the switch, killing the man; or he can refrain from doing so, letting the five die. Is it morally permissible for Ned to throw the switch?

The trolley problem suite has been used in tests to assess moral judgments of subjects from demographically diverse populations (Hauser, 2007; Mikhail, 2007). Interestingly, although all three cases have the same goal, i.e. to save five albeit killing one, subjects come to different judgments on whether the action to reach the goal is permissible or impermissible, i.e. permissible for the Bystander case, but impermissible for the Footbridge and Loop Track cases. As reported by (Mikhail, 2007), the judgments appear to be widely shared among demographically diverse populations.

Fig. 6: Bayesian network for intentional killing recognition. The node intentional killing (*IK*) in the intentional (middle) layer receives Boolean values (*t* or *f*), stating whether the observed action in the third layer is an intentional killing act. The node *IK* is causally affected by *IM* (intended means), stating whether the observed action is performed as an intended means to a greater good, and *PR* (personal reason), stating whether the action is performed due to a personal reason. Both *IM* and *PR* receive Boolean values.

We show how the trolley problems can be modeled within our intention-based decision making framework, leading to outcomes complying with the moral principle of double effect. In all these three cases, as the action to be judged is given explicitly, one just has to decide whether the action is an intentional act of killing or not. The three-layered Bayesian network in Figure 6 is provided for this purpose. Here since we are deciding whether the observed action O is an intentional killing act, we can easily define the CPD of O, $P(O = t \mid IK) = 1$ for all $IK \in \{t,f\}$. Next, the CPD of IK can be defined as follows: *P(IK = t | IM = t, PR) = 1*; *P(IK = t | IM = f, PR = t) = 0.6*; *P(IK =t |IM = f, PR =f) = 0*.

We now only need to focus on prior probabilities of *IM* and *PK*. The P-log program representing this BN can be provided similarly to the one in Example 3.

In the original form of the trolley cases presented above, any personal reason is not considered, thus having prior probability of 0. The prior probability of IM is 0 for the Bystander case and 1 for the other cases. Hence, the probability of intentional killing, i.e. *P(IK = t | O = t)*, is 0 for the Bystander case and 1 for the other two cases.

Let us consider how to model the first two cases, those of the Bystander and the Footbridge. The Loop track case can be done similarly.

Example 11 (Bystander) In the following we see how the Bystander case can be coded using our intention-based decision making framework.

```
1. abds([watching/0, throwing_switch/0]).
2. on_observe(decide)  ← train_comming.
   decide  ← watching.
   decide  ← throwing_switch.
    ← throwing_switch, watching.
3. expect(watching).
   train_straight  ← watching.
   end(die(5))  ← train_straight.
4. expect(throwing_switch).
   redirect_train  ← throwing_switch.
   end(die(1))  ← human(X), side_track(X), redirect_train.
5. side_track(john).    human(john).
6. intentional_kill  ← throwing_switch, has_intention(ned, kill, Pr), prolog(Pr > 0.95).
     ← intentional_killing.
7. A_i << A_j  ← holds_given(end(die(N)), A_i ),
                      holds_given(end(die(K)), A_j), N < K.
```

Part 1 is the declaration of abducibles. Parts 6-7 model the principle of double effect. Namely, part 6 says it is impermissible to have an action (that is, throwing the switch) of intentional killing, which is judged so if intentional killing is predicted by the model with a probability greater some given threshold. This threshold depends on how certain the judgment needs to be provided, for instance, say 0.95 if it is '*guilty beyond reasonable doubt*'. Part 7 says the scenario involving the saving of more people is more favorable. When the train with the fainted conductor is coming, agent Hank has to decide either to watch the train go straight or throw the switch (part 2). There is always the possible expectation to watch the train go straight or the possible expectation to throw the switch, there being no expectations to their contrary (parts 3 and 4).

Because in this Bystander case, the probability of intentional killing is 0, *P(IK = t | O = t) = 0*, there are two prior abductive solutions*: [watching, not throwing_switch], [throwing_switch, not watching]*.

Next the *a posteriori* preferences are taken into account to rule out the less preferred abductive solutions. Considering the a posteriori preference in part 7, the abductive solution including watching is ruled out since it leads to the consequence of five people dying (part 3), which is less preferred than the one including *throwing_switch* that leads to the consequence of non-intentional killing of one person. In short, Hank's decision is to throw the switch to save five people although one will die (unintentionally killed).

Now let us modify the original Bystander case to see how the factor 'personal reason' (*PR*) in the BN model may affect any moral judgment. Supposed there is a good chance that there is some evidence showing that Hank wants to kill the person on the sidetrack: $P(PR = t) = 0.85$. Now, the probability of intentional killing is: $P(IK = t \mid O = t) = 0.51$. It is not enough to judge that Hank's action is one of intentional killing, beyond reasonable doubt, but the probability is high enough to require further investigation to clarify the case.

Example 12 (Footbridge) The footbridge case can be coded with the following program.

```
1. abds([watching/0, shove/1]).
   on_observe(decide) ← train_comming.
   decide ← watching.
   decide ← shove(X).
    ← watching, shove(X).
2. expect(watching).
   train_straight ← watching.
   end(die(5)) ← train_straight.
3. expect(shove(X)) ← stand_near(X).
   on_track(X) ← shove(X).
   stop_train(X) ← on_track(X), heavy(X).
   kill(1) ← human(X), on_track(X).
   kill(0) ← inanimate_object(X), on_track(X).
   end(die(N)) ← kill(N).
4. human(john).                 heavy(john).
   inanimate_object(rock).      heavy(rock).
5. stand_near(john).
   %stand_near(rock).
6. intentional_kill ← human(X), shove(X), has_intention(ian, kill, Pr), Prolog(Pr > 0.95).
    ← intentional_killing.
7. A_i << A_j ← holds_given(end(die(N)), A_i),
                holds_given(end(die(K)), A_j), N < K.
```

Similarly to the previous case, part 1 provides is the declaration of program abducibles. There is always expectation to watch the train go straight and no expectation to its contrary (part 2). However, the action of shoving an object is only possible if there is an object near Ian to shove (part 3). To make this case more interesting we can have an additional heavy object, e.g. rock, on the footbridge near to Ian and see whether our model of the moral rule still allows the reasoning to deliver moral decisions as expected. Similarly to the Bystander case, the double effect principle is modeled in parts 6 and 7.

If there is a person, named John, standing near to Ian (part 5), then there is a possible expectation to shove John (part 3). However, shoving a human is an intentional killing action, which does not satisfy the integrity constraint in part 6, since the probability of intentional killing predicted by the BN model is 1: $P(IK = t \mid O = t) = 1$. Therefore, there is only one abductive solution, to merely watch the train go towards the five people: *[watching, not shoving(john)]*.

Now consider the same initial situation but, instead of a person, there is a heavy inanimate object, a rock, standing near Ian (replace *stand_near(john)* in part 5 with *stand_near(rock)*). Now there is expectation to shove the rock. In addition, it is not an intentional killing. Thus, there are two abductive solutions:

[watching, not shove(rock)], [shove(rock), not watching]. Next, the *a posteriori* preferences in part 7 are taken into account. The abductive solution including watching is ruled out since it leads to the consequence of killing five people, less preferred than the one including *shove(rock)* that leads to the consequence of killing nobody.

In short, if standing near to Ian is a person he has only one choice to watch the train go straight and kill five people since shoving a person to the sidetrack is an intentional killing action. However, if standing near to him were an inanimate object, he would shove the object to stop the train, saving the five and killing no one.

Uncertainty about observed actions. Usually moral reasoning is performed upon conceptual knowledge of the actions. But it often happens that one has to pass a moral judgment on a situation without actually observing the situation, i.e. there is no full, certain information about the actions. The BN in Figure 6 is not applicable anymore. In this case, it is important to be able to reason about the actions, under uncertainty, that might have occurred, and thence provide judgment adhering to moral rules within some prescribed uncertainty level. Courts, for example, are required to proffer rulings beyond reasonable doubt. There is a vast body of research on proof beyond reasonable doubt within the legal community, e.g. (Newman, 2006). For illustration, consider this variant of the Footbridge case.

Example 13 (Moral reasoning with uncertain actions) Suppose a jury in a court is faced with the case where the action of Ian shoving the man onto the track was not observed. Instead, they are only presented with the fact that the man died on the sidetrack and Ian was seen on the bridge at the occasion. Is Ian guilty (*beyond reasonable doubt*), i.e. does he violate the double effect principle, of shoving the man onto the track intentionally?

To answer this question, one should be able to reason about the possible explanations of the observations, on the available evidence. The following code shows a model for this example. Given the active goal judge (part 2), two abducibles are available, i.e. *verdict(guilty_beyond_reasonable_doubt)* and *verdict(not_guilty)*. Depending on how probable is each possible verdict, *verdict(guilty_beyond_reasonable_doubt)* or *verdict(not_guilty)* is expected a priori (part 3 and 9). The sort intentionality in part 4 represents the possibilities of an action being performed intentionally (*int*) or non-intentionally (*not_int*). Random attributes *df_run* and *br_slip* in part 5 and 6 denote two kinds of evidence: Ian was definitely running on the bridge in a hurry (*df_run*) and the bridge was slippery at the time (*br_slip*), respectively. Each has prior probability of 4/10. The probability with which shoving is performed intentionally is captured by the random attribute shoved (part 7), which is causally influenced by both evidence. Part 9 defines when the verdicts (*guilty* and *not_guilty*) are considered highly probable using the meta-probabilistic predicate *pr_iShv/1*, defined in part 8. It denotes the probability of intentional shoving, whose value is determined by the existence of evidence that Ian was running in a hurry past the man (signaled by predicate *evd_run/1*) and that the bridge was slippery (signaled by predicate *evd_slip/1*).

```
1. abds([verdict/1]).
2. on_observe(judge).
   judge ← verdict(guilty_beyond_reasonable_doubt).
   judge ← verdict(not_guilty).
3. expect(verdict(X)) ← prolog(highly_probable(X)).

beginPlog.
4. bool = {t, f}.   intentionality = {int, not_int}.
5. df_run : bool.   random(rdr,df_run,full).
   pa(rdr,df_run(t),d_(4, 10)).
6. br_slip : bool.  random(rsb,br_slip,full).
```

```
   pa(rsb,br_slip(t),d_(4, 10)).
7. shoved : intentionality.    random(rs, shoved, full).
   pa(rs,shoved(int),d_(97,100)) :- df_run(f),br_slip(f).
   pa(rs,shoved(int),d_(45,100)) :- df_run(f),br_slip(t).
   pa(rs,shoved(int),d_(55,100)) :- df_run(t),br_slip(f).
   pa(rs,shoved(int),d_(5,100))  :- df_run(t),br_slip(t).

   :- dynamic evd_run/1, evd_slip/1.
8. pr_iShv(Pr)  :- evd_run(X), evd_slip(Y), !,
                   pr(shoved(int) '|' obs(df_run(X)) & obs(br_slip(Y)), Pr).
   pr_iShv(Pr)  :- evd_run(X), !,
                   pr(shoved(int) '|' obs(df_run(X)), Pr).
   pr_iShv(Pr)  :- evd_slip(Y), !,
                   pr(shoved(int) '|' obs(br_slip(Y)), Pr).
   pr_iShv(Pr)  :- pr(shoved(int), Pr).
9. highly_probable(guilty_beyond_reasonable_doubt) :- pr_iShv(PrG), PrG > 0.95.
   highly_probable(not_guilty) :- pr_iShv(PrG), PrG < 0.6.
endPlog.
```

Using the above model, different judgments can be delivered by our system, subject to available evidence and attending truth-value. We exemplify some cases in the sequel. If both evidence are available, where it is known that Ian was running in a hurry on the slippery bridge, then he may have bumped the man accidentally, shoving him unintentionally onto the track. This case is captured by the first *pr_iShv* rule (part 8): the probability of intentional shoving is 0.05. Thus, the atom *highly_probable(not guilty)* holds (part 10). Hence, *verdict(not_guilty)* is the preferred final abductive solution (part 3). The same abductive solution is obtained if it is observed that the bridge was slippery, but whether Ian was running in a hurry was not observable. The probability of intentional shoving, captured by *pr_iShv*, is 0.29.

On the other hand, if the evidence shows that Ian was not running in a hurry and the bridge was also not slippery, then they do not support the explanation that the man was shoved unintentionally, e.g., by accidental bumping. The action of shoving is more likely to have been performed intentionally. Using the model, the probability of 0.97 is returned and, being greater than 0.95, *verdict(guilty_beyond_reasonable_doubt)* becomes the sole abductive solution. In another case, if it is only known the bridge was not slippery and no other evidence is available, then the probability of intentional shoving becomes 0.79, and, by parts 4 and 10, no abductive solution is preferred. This translates into the need for more evidence, as the available one is not enough to issue judgment.

Intention-based decision making in Game Theory

In strategic and economic situations as typically modeled using the game theoretical framework (Hofbauer & Sigmund, 1998; Osborne, 2004), the achievement of a goal by an agent usually does not depend uniquely on its own actions, but also on the decisions and actions of others—especially when the possibility of communication is limited (Heinze, 2003; Kraus, 1997; Pinker, et al., 2008; Tomasello, 2008). The knowledge about intention of others in such situations could enable an recognizing agent to plan in advance, either to secure a successful cooperation, to deal with potential hostile behaviors, and thus take the best advantage of such knowledge (Bratman, 1987; Cohen & Levesque, 1990; Han, 2012; Han, Pereira, & Santos, 2011a; Han, Pereira, et al., 2012a; O. Roy, 2009b; van Hees & Roy, 2008). Additionally, in more realistic settings where deceit may offer additional profits, agents often attempt to hide their real intentions and make others believe in faked ones (Han, 2012; Han, Pereira, et al., 2012b; Robson, 1990; Tomasello, 2008; Trivers, 2011). Undoubtedly, in all such situations a capability of recognizing intentions of others and take them into account when making decision is crucial, providing its withholders with significant net benefit or evolutionary advantages. Indeed, the capacity for intention recognition and intention-based decision making can be found abundantly in many kinds of humans' interactions and communications, widely documented for instance in (Cheney & Seyfarth, 2007; Meltzoff, 2007; Tomasello, 1999, 2008; Woodward, et al., 2009). In addition, there is a large body of literature on experimental economics that shows the importance of intention-based decision making in diverse kinds of strategic games, for instance, the Prisoner's dilemma (Frank, et al., 1993), the Moonlighting game (Falk, et al., 2008; Radke, et al., 2012) and the Ultimatum game (Radke, et al., 2012). In addition, computational models show that the taking into account of the ongoing strategic intentions of others is crucial for agents' success in the course of different strategic games (Han, 2012; Han, et al., 2011a, 2011b; Han, Pereira, et al., 2012a, 2012b; Janssen, 2008).

Let us consider some examples of intention-based decision making in the context of the Prisoner's dilemma (PD), where in each interaction a player needs to choose a move, either to cooperate ('c') or to defect ('d'). In a one-shot PD interaction, it is always better off choosing to defect, but cooperation might be favorable if the PD is repeated (called iterated PD), that is, there is a good chance that players will play the same PD with each other again. Several successful strategies have been provided in the context of the iterated PD (see a survey in (Sigmund, 2010) (Chapter 3)), most famously amongst them are tit-for-tat (*tft*) and win-stay-lose-shift (*wsls*).

The following two strategies (each denoted by *IR*), operating upon intent-based decision making, have been shown to be better than those famous strategies of the iterated PD (Han, et al., 2011a, 2011b; Han, Pereira, et al., 2012a). In the sequel we show how to model them within our framework.

Example 14 (Intention-based decision making rule in (Han, et al., 2011a; Han, Pereira, et al., 2012b; Janssen, 2008)) Prefer to cooperate if the co-player intends to cooperate, and prefer to defect otherwise.

1. abds([move/1]).
2. on_observed(decide) ← new_interaction.
3. decide ← move(c).
   decide ← move(d).
   ← move(c), move(d).
4. expect(move(X)).
5. move(c) <| move(d) ← has intention(co_player, c).
   move(d) <| move(c) ← has intention(co_player, d).

At the start of a new interaction, an IR player needs to choose a move, either cooperate (*c*) or defect (*d*) (parts 2-3). Both options are expected, and there are no expectations to the contrary (part 4). There are two *a priori* preferences in part 5, stating that an IR player prefers to cooperate if the co-player's recognized intention is to cooperate, and prefers to defect otherwise. The built-in predicate *has_intention/2*, in the body of the preferences, triggers the intention recognition module to validate if the co-player is more likely to have the intention expressed in the second argument.

Example 15 (Intention-based decision making rule in (Han, et al., 2011b; Han, Pereira, et al., 2012a)) Defect if the co-player's recognized intention or rule of behavior is always-cooperate (*allc*) or always-defect (*alld*), cooperate if it is *tft*; and if it is *wsls*, cooperate if last game state is both cooperated (denoted by R) or both defected (denoted by P) and defect if the current game state is IR defected and the co-player cooperated (denoted by T) or vice versa (denoted by S).

This rule of behavior is learnt using a dataset collected from prior interactions with those strategies (Han, et al., 2011b).

1. abds([move/1]).
2. on observed(decide) ← new interaction.
3. decide ← move(c).
   decide ← move(d).
   ← move(c), move(d).
4. expect(move(X)).
5. move(d) <| move(c) ← has_intention(co_player, allc).
   move(d) <| move(c) ← has_intention(co_player, alld).
   move(c) <| move(d) ← has_intention(co_player, tft).
   move(c) <| move(d) ← has_intention(co_player, wsls), game_state(s), (s = 'R'; s = 'P').
   move(d) <| move(c) ← has_intention(co_player, wsls), game_state(s), (s = 'T'; s = 'S').

At the start of a new interaction, an IR needs to choose a move, either cooperate (c) or defect (d) (parts 2-3). Both options are expected, and there are no expectations to the contrary. The *a priori* preferences in part 5 stating which move IR prefers to choose, given the recognized intention of the co- player (*allc, alld, tft* or *wsls*) and the current game state ('T', 'R', 'P' or 'S'). The built-in predicate *has_intention/2* in the body of the preferences triggers the intention recognition module to validate if the co-player is most likely to follow a given intention (strategy), specified by the second argument.

In short, our framework is general and expressive, suitable for intention-based decision making in the context of game theory.

Conclusions and Future Works

We have summarized our previous work on Evolution Prospection (EP) (Pereira & Han, 2009a, 2009b) and have shown how to obtain its coherent combination with the intention recognition system, for achieving intention-based decision making. The EP system has proven useful before for the purpose of decision making (Han & Pereira, 2010a, 2010b, 2011b; Han, Saptawijaya, et al., 2012; Pereira & Han, 2009a, 2009b, 2011b), and has now been empowered to take into account the intentions of other agents—an important aspect that has not been well explored so far (O. Roy, 2009b; van Hees & Roy, 2008). The fact that both systems are Logic Programming based enabled their easy integration. We have described and exemplified several ways in which an EP agent can benefit from having an ability to recognize intentions in other agents.

Notwithstanding, the combination of intention recognitions approach we have used here is not deemed restricted to Logic Programming based systems. In general, any intention recognition system, and indeed, any decision making system, can be considered. The ideas of combined integration described here can be adopted by other decision making systems to account for intentions.

We have addressed the need for intention-based decision making in different application domains, including Ambient Intelligence (Sadri, 2011a) and Elder Care (Cesta & Pecora, 2004; Sadri, 2008), where decision making techniques as well as intention recognition abilities are becoming of increased importance (Geib, 2002; Pereira & Han, 2011a; Sadri, 2010). Furthermore, we have also described how important and ubiquitous intention-based decision making is in the moral reasoning and game theory setting application domains.

In future work, we consider applying our combined system to other application domains, including story understanding (Charniak & Goldman, 1990), human-computer and interface-agents systems (Armentano & Amandi, 2007; Hong, 2001; Lesh, 1998), traffic monitoring (Pynadath & Wellman, 1995), assistive living (Geib, 2002; Haigh, et al., 2004; Pereira & Han, 2011a; P. Roy, et al., 2007; Tahboub, 2006), military settings (Heinze, 2003; Mao & Gratch, 2004), and moral reasoning (Han, Saptawijaya, et al., 2012), where intention recognition has proven useful and of great practicality. Another area of future development is to extend our system to enable collective or group intention recognition (Sukthankar, 2007; Sukthankar & Sycara, 2008) in a decision making process. In this regard, we have made some initial attempts in the Elder Care domain (Han & Pereira, 2010a, 2010b).

Acknowledgments

We thank Ari Saptawijaya for his comments on an earlier version of this chapter. TAH acknowledges the support from FCT-Portugal (grant reference *SFRH/BD/62373/2009*).

REFERENCES

Alferes, J. J., Pereira, L. M., & Swift, T. (2004). Abduction in Well-Founded Semantics and Generalized Stable Models via Tabled Dual Programs. *Theory and Practice of Logic Programming, 4*(4), 383 - 428.

Armentano, M. G., & Amandi, A. (2007). Plan recognition for interface agents. *Artif. Intell. Rev., 28*(2), 131 - 162.

Armentano, M. G., & Amandi, A. (2009). *Goal recognition with variable-order Markov models.* Paper presented at the Proceedings of the 21st international joint conference on Artificial intelligence.

Baral, C. (2003). *Knowledge Representation, Reasoning, and Declarative Problem Solving*: Cambridge University Press.

Baral, C., Gelfond, M., & Rushton, N. (2009). Probabilistic reasoning with answer sets. *Theory and Practice of Logic Programming, 9*(1), 57-144.

Binmore, K. G. (2009). *Rational Decisions*: Princeton University Press.

Blaylock, N., & Allen, J. (2003). *Corpus-based, statistical goal recognition.* Paper presented at the Proceedings of the 18th international joint conference on Artificial intelligence (IJCAI'03).

Bratman, M. E. (1987). *Intention, Plans, and Practical Reason*: The David Hume Series, CSLI.

Bratman, M. E. (1999). *Faces of Intention: Selected Essays on Intention and Agency*: Cambridge University Press.

Burmeister, B., Arnold, M., Copaciu, F., & Rimassa, G. (2008). *BDI-agents for agile goal-oriented business processes.* Paper presented at the Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: industrial track.

Castro, L., Swift, T., & Warren, D. S. (2007). XASP: Answer set programming with xsb and smodels. Accessed at http://xsb.sourceforge.net/packages/xasp.pdf.

Cesta, A., & Pecora, F. (2004). *The Robocare Project: Intelligent Systems for Elder Care.* Paper presented at the AAAI Fall Symposium on Caring Machines: AI in Elder Care.

Charniak, E., & Goldman, R. P. (1990). *Plan Recognition in Stories and in Life.* Paper presented at the Proceedings of the Fifth Annual Conference on Uncertainty in Artificial Intelligence.

Cheney, D. L., & Seyfarth, R. M. (2007). *Baboon Metaphysics: The Evolution of a Social Mind*: University Of Chicago Press.

Cohen, P. R., & Levesque, H. J. (1990). Intention is Choice with Commitment. *Artificial Intelligence, 42*(2-3), 213 - 261.

Cook, D., Augusto, J., & Jakkula, V. (2009). Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing, 5*(4), 277-298.

Falk, A., Fehr, E., & Fischbacher, U. (2008). Testing theories of fairness---Intentions matter. *Games and Economic Behavior, 62*(1), 287 - 303.

Frank, R. H., Gilovich, T., & Regan, D. T. (1993). The evolution of one-shot cooperation: An experiment. *Ethology and Sociobiology, 14*(4), 247 - 256.

Friedewald, M., Costa, O. D., Punie, Y., Alahuhta, P., & Heinonen, S. (2005). Perspectives of ambient intelligence in the home environment. *Telemat. Inf., 22*.

Friedewald, M., Vildjiounaite, E., Punie, Y., & Wright, D. (2007). Privacy, identity and security in ambient intelligence: A scenario analysis. *Telematics and Informatics, 24*(1), 15 - 29.

Geib, C. W. (2002). *Problems with intent recognition for elder care.* Paper presented at the Proceedings of AAAI Workshop Automation as Caregiver.

Gelfond, M., & Lifschitz, V. (1993). Representing actions and change by logic programs. *Journal of Logic Programming, 17*(2,3,4), 301 - 323.

Giuliani, M. V., Scopelliti, M., & Fornara, F. (2005). *Elderly people at home: technological help in everyday activities.* Paper presented at the IEEE International Workshop on In Robot and Human Interactive Communication.

Haigh, K., Kiff, L., Myers, J., Guralnik, V., Geib, C., Phelps, J., et al. (2004). *The Independent LifeStyle Assistant (I.L.S.A.): AI Lessons Learned.* Paper presented at the Proceedings of Conference on Innovative Applications of Artificial Intelligence.

Han, T. A. (2009). *Evolution Prospection with Intention Recognition via Computational Logic.* Technical University of Dresden, Germany, Dresden.

Han, T. A. (2012). *Intention Recognition, Commitments and Their Roles in the Evolution of Cooperation* Universidade Nova de Lisboa, Lisbon.

Han, T. A., Carroline, D. P., & Damasio, C. V. (2008). *An Implementation of Extended P-Log Using XASP*. Paper presented at the Proceedings of the 24th International Conference on Logic Programming.

Han, T. A., Carroline, D. P., & Damasio, C. V. (2009). *Tabling for P-log probabilistic query evaluation.* Paper presented at the New Trends in Artificial Intelligence, Proceedings of 14th Portuguese Conference on Artificial Intelligence (EPIA 2009), Evora, Portugal.

Han, T. A., & Pereira, L. M. (2010a). *Collective Intention Recognition and Elder Care.* Paper presented at the AAAI 2010 Fall Symposium on Proactive Assistant Agents (PAA 2010).

Han, T. A., & Pereira, L. M. (2010b). *Proactive Intention Recognition for Home Ambient Intelligence.* Paper presented at the Proceedings of 5th Workshop on Artificial Intelligence Techniques for Ambient Intelligence (AITAmI'10), Ambient Intelligence and Smart Environments - vol.8.

Han, T. A., & Pereira, L. M. (2011a). *Context-dependent Incremental Intention Recognition through Bayesian Network Model Construction.* Paper presented at the Proceedings of the Eighth UAI Bayesian Modeling Applications Workshop (UAI-AW 2011).

Han, T. A., & Pereira, L. M. (2011b). *Intention-based decision making with evolution prospection*. Paper presented at the Proceedings of the 15th Portugese conference on Progress in artificial intelligence.

Han, T. A., Pereira, L. M., & Santos, F. C. (2011a). Intention Recognition Promotes the Emergence of Cooperation. *Adaptive Behavior, 19*(3), 264-279.

Han, T. A., Pereira, L. M., & Santos, F. C. (2011b). *The role of intention recognition in the evolution of cooperative behavior.* Paper presented at the Proceedings of the 22nd international joint conference on Artificial intelligence (IJCAI'2011).

Han, T. A., Pereira, L. M., & Santos, F. C. (2012a). Corpus-based intention recognition in cooperation dilemmas. *Artificial Life*.

Han, T. A., Pereira, L. M., & Santos, F. C. (2012b, June). *Intention Recognition, Commitment and The Evoution of Cooperation* Paper presented at the The 2012 IEEE World Congress on Computational Intelligence (IEEE WCCI 2012), Congress on Evolutionary Computation (IEEE CEC 2012), Brisbane, Australia.

Han, T. A., Saptawijaya, A., & Pereira, L. M. (2012). *Moral Reasoning Under Uncertainty*. Paper presented at the Procs. of The 18th Intl. Conf. on Logic for Programming, Artificial Intelligence and Reasoning (LPAR-18).

Hauser, M. D. (2007). *Moral Minds, How Nature Designed Our Universal Sense of Right and Wrong*: Little Brown.

Heinze, C. (2003). *Modeling Intention Recognition for Intelligent Agent Systems.*

Hofbauer, J., & Sigmund, K. (1998). *Evolutionary Games and Population Dynamics*: Cambridge University Press.

Hong, J. (2001). Goal Recognition through Goal Graph Analysis. *Journal of Artificial Intelligence Research, 15*, 1-30.

Janssen, M. A. (2008). Evolution of cooperation in a one-shot Prisoner's Dilemma based on recognition of trustworthy and untrustworthy agents. *Journal of Economic Behavior & Organization, 65*(3-4), 458-471.

Kakas, A. C., Kowalski, R. A., & Toni, F. (1993). Abductive Logic Programming. *Journal of Logic and Computation, 2*(6), 719-770.

Kaminka, G. A., Tambe, D. V. P. M., Pynadath, D. V., & Tambe, M. (2002). Monitoring Teams by Overhearing: A Multi-Agent Plan-Recognition Approach. *Journal of Artificial Intelligence Research, 17*.

Kraus, S. (1997). Negotiation and Cooperation in Multi-Agent Environments. *Artificial Intelligence, 94*(1-2), 79-98.

Lesh, N. (1998). *Scalable and Adaptive Goal Recognition.*

Malle, B. F., Moses, L. J., & Baldwin, D. A. (2003). *Intentions and intentionality: foundations of social cognition*: MIT Press.

Mao, W., & Gratch, J. (2004). *A Utility-Based Approach to Intention Recognition.* Paper presented at the AAMAS 2004 Workshop on Agent Tracking: Modeling Other Agents from Observations.

Meltzoff, A. N. (2005). Imitation and other minds: the ``Like Me'' hypothesi. In S. a. C. Hurley, N. (Ed.), *Perspectives on imitation: From neuroscience to social science. Imitation, human development, and culture* (pp. 55--77): Cambridge, MA: MIT Press.

Meltzoff, A. N. (2007). The framework for recognizing and becoming an intentional agent. *Acta Psychol Amst, 124*(1), 26 - 43.

Mikhail, J. (2007). Universal Moral Grammar: Theory, Evidence, and The Future. *Trends in Cognitive Sciences, 11*(4), 143-152.

Newman, J. O. (2006). Quantifying the standard of proof beyond a reasonable doubt: a comment on three comments. *Law, Probability and Risk, 5*(3-4), 267-269.

Niemela, I., & Simons, P. (1997). *Probabilistic Reasoning with Answer Sets.* Paper presented at the LPNMR4.

Osborne, M. J. (2004). *An introduction to game theory*: Oxford University Press.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*: Morgan Kaufmann.

Pearl, J. (2000). *Causality: Models, Reasoning, and Inference*: Cambridge University Press.

Pereira, L. M., Dell'Acqua, P., & Lopes, G. (2012). Inspecting and Preferring Abductive Models *Handbook on Reasoning-based Intelligent Systems*: World Scientific Publishers.

Pereira, L. M., & Han, T. A. (2009a). *Evolution Prospection.* Paper presented at the Proceedings of International Symposium on Intelligent Decision Technologies (KES-IDT'09).

Pereira, L. M., & Han, T. A. (2009b). Evolution Prospection in Decision Making. *Intelligent Decision Technologies, 3*(3), 157-171.

Pereira, L. M., & Han, T. A. (2009c). *Intention Recognition via Causal Bayes Networks plus Plan Generation.* Paper presented at the Progress in Artificial Intelligence, Proceedings of 14th Portuguese International Conference on Artificial Intelligence (EPIA'09), Evora, Portgual.

Pereira, L. M., & Han, T. A. (2011a). Elder care via intention recognition and evolution prospection *Proceedings of the 18th international conference on Applications of declarative programming* *and knowledge management (INAP), select extended paper* (Vol. LNAI 6547, pp. 170-187). Evora, Portugal: Springer.

Pereira, L. M., & Han, T. A. (2011b). Intention Recognition with Evolution Prospection and Causal Bayes Networks *Computational Intelligence for Engineering Systems 3: Emergent Applications* (pp. 1-33): Springer

Pereira, L. M., & Lopes, G. (2009). Prospective Logic Agents. *International Journal of Reasoning-based Intelligent Systems (IJRIS), 1*(3/4), Inderscience-208.

Pinker, S., Nowak, M. A., & Lee, J. J. (2008). The logic of indirect speech. *Proceedings of the National Academy of Sciences of the United States of America, 105*(3), 833-838.

Pynadath, D. V., & Wellman, M. P. (1995). *Accounting for Context in Plan Recognition, with Application to Traffic Monitoring.* Paper presented at the Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI'95).

Radke, S., Guroglu, B., & de Bruijn, E. R. A. (2012). There's Something about a Fair Split: Intentionality Moderates Context-Based Fairness Considerations in Social Decision-Making. *PLoS ONE, 7*(2).

Rao, A. S., & Georgeff, M. P. (1991). *Modeling Rational Agents within a BDI-Architecture.* Paper presented at the Proceedings of the second International Conference of Principles of Knowledge Representation and Reasoning.

Rao, A. S., & Georgeff, M. P. (1995). *BDI Agents: from theory to practice.* Paper presented at the Proceeding of First International Conference on Multiagent Systems.

Robson, A. (1990). Efficiency in Evolutionary Games: Darwin, Nash, and the Secret Handshake. *Journal of Theoretical Biology, 144*(3), 379-396.

Roy, O. (2009a). Intentions and interactive transformations of decision problems. *Synthese, 169*(2), 335-349.

Roy, O. (2009b). *Thinking before Acting: Intentions, Logic, Rational Choice.*

Roy, P., Bouchard, B., Bouzouane, A., & Giroux, S. (2007). *A hybrid plan recognition model for Alzheimer's patients: interleaved-erroneous dilemma.* Paper presented at the Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology.

Russell, S. J., & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*: Pearson Education.

Sadri, F. (2008). *Multi-agent ambient intelligence for elderly care and assistance.* Paper presented at the Proceedings of International Electronic Conference on Computer Science.

Sadri, F. (2010). Logic-based Approaches to Intention Recognition, *Handbook of Research on Ambient Intelligence: Trends and Perspectives* (pp. 375).

Sadri, F. (2011a). Ambient intelligence: A survey. *ACM Comput. Surv., 43*(4), 1-66.

Sadri, F. (2011b). *Intention Recognition with Event Calculus Graphs and Weight of Evidence.* Paper presented at the In Proceedings 4th International Workshop on Human Aspects in Ambient Intelligence.

Searle, J. R. (1995). *The Construction of Social Reality*: New York: The Free Press.

Searle, J. R. (2010). *Making the Social World: The Structure of Human Civilization*: Oxford University Press.

Sigmund, K. (2010). *The Calculus of Selfishness*: Princeton University Press.

Singh, M. P. (1991). *Intentions, Commitments and Rationality.* Paper presented at the In 13th Annual Conference of the Cognitive Science Society.

Sukthankar, G. R. (2007). *Activity Recognition for Agent Teams.*

Sukthankar, G. R., & Sycara, K. (2008). *Robust and efficient plan recognition for dynamic multi-agent teams.* Paper presented at the Proceedings of International Conference on Autonomous Agents and Multi-Agent Systems.

Swift, T. (1999). Tabling for non-monotonic programming. *Annals of Mathematics and Artificial Intelligence, 25*(3-4), 240.

Tahboub, K. A. (2006). Intelligent Human-Machine Interaction Based on Dynamic Bayesian Networks Probabilistic Intention Recognition. *J. Intell. Robotics Syst., 45*, 31-52.

Tomasello, M. (1999). *The Cultural Origins of Human Cognition*: Harvard University Press.

Tomasello, M. (2008). *Origins of Human Communication*: MIT Press.
Trivers, R. (2011). *The Folly of Fools: The Logic of Deceit and Self-Deception in Human Life*: Basic Books.
Tu, P. H., Son, T. C., & Baral, C. (2007). Reasoning and Planning with Sensing Actions, Incomplete Information, and Static Causal Laws using Answer Set Programming. *Theory and Practice of Logic Programming, 7*(4), 377 - 450.
Tu, P. H., Son, T. C., Gelfond, M., & Morales, A. R. (2011). Approximation of action theories and its application to conformant planning. *Artificial Intelligence, 175*(1), 79 - 119.
van Hees, M., & Roy, O. (2008). Intentions and plans in decision and game theory *Reasons and intentions* (pp. 207 - 226): Ashgate Publishers.
Wikipedia Burglar Alarm: http://en.wikipedia.org/wiki/Burglar_alarm. Last access: June 18, 2012.
Woodward, A. L., Sommerville, J. A., Gerson, S., Henderson, A. M. E., & Buresh, J. (2009). The Emergence of Intention Attribution in Infancy *The Psychology of Learning and Motivation* (Vol. 51, pp. 187 - 222): Academic Press.
Wooldridge, M. (2000). *Reasoning About Rational Agents*: MIT Press.
Wooldridge, M. (2002). Reasoning about Rational Agents. *The Journal of Artificial Societies and Social Simulation, 5*(1).
XSB (2009). The XSB System Version 3.2 Vol. 2: Libraries, Interfaces and Packages.
Young, L., & Saxe, R. (2011). When ignorance is no excuse: Different roles for intent across moral domains. *Cognition, 120*(2), 202 - 214.

ADDITIONAL READING SECTION

Alferes, J. J., Pereira, L. M., & Swift, T. (2004). Abduction in Well-Founded Semantics and Generalized Stable Models via Tabled Dual Programs. *Theory and Practice of Logic Programming, 4*(4), 383 - 428.

Armentano, M. G., & Amandi, A. (2007). Plan recognition for interface agents. *Artif. Intell. Rev., 28*(2), 131 - 162.

Armentano, M. G., & Amandi, A. (2009). *Goal recognition with variable-order Markov models.* Paper presented at the Proceedings of the 21st international joint conference on Artificial intelligence.

Baral, C. (2003). *Knowledge Representation, Reasoning, and Declarative Problem Solving*: Cambridge University Press.

Baral, C., Gelfond, M., & Rushton, N. (2009). Probabilistic reasoning with answer sets. *Theory and Practice of Logic Programming, 9*(1), 57-144.

Binmore, K. G. (2009). *Rational Decisions*: Princeton University Press.

Blaylock, N., & Allen, J. (2003). *Corpus-based, statistical goal recognition.* Paper presented at the Proceedings of the 18th international joint conference on Artificial intelligence (IJCAI'03).

Bratman, M. E. (1987). *Intention, Plans, and Practical Reason*: The David Hume Series, CSLI.

Bratman, M. E. (1999). *Faces of Intention: Selected Essays on Intention and Agency*: Cambridge University Press.

Burmeister, B., Arnold, M., Copaciu, F., & Rimassa, G. (2008). *BDI-agents for agile goal-oriented business processes.* Paper presented at the Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: industrial track.

Castro, L., Swift, T., & Warren, D. S. (2007). XASP: Answer set programming with XSB and smodels. Accessed at http://xsb.sourceforge.net/packages/xasp.pdf.

Cesta, A., & Pecora, F. (2004). *The Robocare Project: Intelligent Systems for Elder Care.* Paper presented at the AAAI Fall Symposium on Caring Machines: AI in Elder Care.

Charniak, E., & Goldman, R. P. (1990). *Plan Recognition in Stories and in Life.* Paper presented at the Proceedings of the Fifth Annual Conference on Uncertainty in Artificial Intelligence.

Cheney, D. L., & Seyfarth, R. M. (2007). *Baboon Metaphysics: The Evolution of a Social Mind*: University Of Chicago Press.

Cohen, P. R., & Levesque, H. J. (1990). Intention is Choice with Commitment. *Artificial Intelligence, 42*(2-3), 213 - 261.

Cook, D., Augusto, J., & Jakkula, V. (2009). Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing, 5*(4), 277-298.

Falk, A., Fehr, E., & Fischbacher, U. (2008). Testing theories of fairness---Intentions matter. *Games and Economic Behavior, 62*(1), 287 - 303.

Frank, R. H., Gilovich, T., & Regan, D. T. (1993). The evolution of one-shot cooperation: An experiment. *Ethology and Sociobiology, 14*(4), 247 - 256.

Friedewald, M., Costa, O. D., Punie, Y., Alahuhta, P., & Heinonen, S. (2005). Perspectives of ambient intelligence in the home environment. *Telemat. Inf., 22*.

Friedewald, M., Vildjiounaite, E., Punie, Y., & Wright, D. (2007). Privacy, identity and security in ambient intelligence: A scenario analysis. *Telematics and Informatics, 24*(1), 15 - 29.

Geib, C. W. (2002). *Problems with intent recognition for elder care.* Paper presented at the Proceedings of AAAI Workshop Automation as Caregiver.

Gelfond, M., & Lifschitz, V. (1993). Representing actions and change by logic programs. *Journal of Logic Programming, 17*(2,3,4), 301 - 323.

Giuliani, M. V., Scopelliti, M., & Fornara, F. (2005). *Elderly people at home: technological help in everyday activities.* Paper presented at the IEEE International Workshop on In Robot and Human Interactive Communication.

Haigh, K., Kiff, L., Myers, J., Guralnik, V., Geib, C., Phelps, J., et al. (2004). *The Independent LifeStyle Assistant (I.L.S.A.): AI Lessons Learned.* Paper presented at the Proceedings of Conference on Innovative Applications of Artificial Intelligence.

Han, T. A. (2009). *Evolution Prospection with Intention Recognition via Computational Logic.* Unpublished Master Thesis Technical University of Dresden, Germany, Dresden.

Han, T. A. (2012). *Intention Recognition, Commitments and Their Roles in the Evolution of Cooperation* Universidade Nova de Lisboa, Lisbon.

Han, T. A., Carroline, D. P., & Damasio, C. V. (2008). *An Implementation of Extended P-Log Using XASP*. Paper presented at the Proceedings of the 24th International Conference on Logic Programming.

Han, T. A., Carroline, D. P., & Damasio, C. V. (2009). *Tabling for P-log probabilistic query evaluation.* Paper presented at the New Trends in Artificial Intelligence, Proceedings of 14th Portuguese Conference on Artificial Intelligence (EPIA 2009), Evora, Portugal.

Han, T. A., & Pereira, L. M. (2010a). *Collective Intention Recognition and Elder Care.* Paper presented at the AAAI 2010 Fall Symposium on Proactive Assistant Agents (PAA 2010).

Han, T. A., & Pereira, L. M. (2010b). *Proactive Intention Recognition for Home Ambient Intelligence.* Paper presented at the Proceedings of 5th Workshop on Artificial Intelligence Techniques for Ambient Intelligence (AITAmI'10), Ambient Intelligence and Smart Environments - vol.8.

Han, T. A., & Pereira, L. M. (2011a). *Context-dependent Incremental Intention Recognition through Bayesian Network Model Construction.* Paper presented at the Proceedings of the Eighth UAI Bayesian Modeling Applications Workshop (UAI-AW 2011).

Han, T. A., & Pereira, L. M. (2011b). *Intention-based decision making with evolution prospection*. Paper presented at the Proceedings of the 15th Portugese conference on Progress in artificial intelligence.

Han, T. A., Pereira, L. M., & Santos, F. C. (2011a). Intention Recognition Promotes the Emergence of Cooperation. *Adaptive Behavior, 19*(3), 264-279.

Han, T. A., Pereira, L. M., & Santos, F. C. (2011b). *The role of intention recognition in the evolution of cooperative behavior.* Paper presented at the Proceedings of the 22nd international joint conference on Artificial intelligence (IJCAI'2011).

Han, T. A., Pereira, L. M., & Santos, F. C. (2012a). Corpus-based intention recognition in cooperation dilemmas. *Artificial Life*.

Han, T. A., Pereira, L. M., & Santos, F. C. (2012b, June). *Intention Recognition, Commitment and The Evoution of Cooperation* Paper presented at the The 2012 IEEE World Congress on Computational Intelligence (IEEE WCCI 2012), Congress on Evolutionary Computation (IEEE CEC 2012), Brisbane, Australia.

Han, T. A., Saptawijaya, A., & Pereira, L. M. (2012). *Moral Reasoning Under Uncertainty*. Paper presented at the Procs. of The 18th Intl. Conf. on Logic for Programming, Artificial Intelligence and Reasoning (LPAR-18).

Hauser, M. D. (2007). *Moral Minds, How Nature Designed Our Universal Sense of Right and Wrong*: Little Brown.

Heinze, C. (2003). *Modeling Intention Recognition for Intelligent Agent Systems.*

Hofbauer, J., & Sigmund, K. (1998). *Evolutionary Games and Population Dynamics*: Cambridge University Press.

Hong, J. (2001). Goal Recognition through Goal Graph Analysis. *Journal of Artificial Intelligence Research, 15*, 1-30.

Janssen, M. A. (2008). Evolution of cooperation in a one-shot Prisoner's Dilemma based on recognition of trustworthy and untrustworthy agents. *Journal of Economic Behavior & Organization, 65*(3-4), 458-471.

Kakas, A. C., Kowalski, R. A., & Toni, F. (1993). Abductive Logic Programming. *Journal of Logic and Computation, 2*(6), 719-770.

Kaminka, G. A., Tambe, D. V. P. M., Pynadath, D. V., & Tambe, M. (2002). Monitoring Teams by Overhearing: A Multi-Agent Plan-Recognition Approach. *Journal of Artificial Intelligence Research, 17*.

Kraus, S. (1997). Negotiation and Cooperation in Multi-Agent Environments. *Artificial Intelligence, 94*(1-2), 79-98.

Lesh, N. (1998). *Scalable and Adaptive Goal Recognition.*

Malle, B. F., Moses, L. J., & Baldwin, D. A. (2003). *Intentions and intentionality: foundations of social cognition*: MIT Press.

Mao, W., & Gratch, J. (2004). *A Utility-Based Approach to Intention Recognition.* Paper presented at the AAMAS 2004 Workshop on Agent Tracking: Modeling Other Agents from Observations.

Meltzoff, A. N. (2005). Imitation and other minds: the ``Like Me'' hypothesi. In S. a. C. Hurley, N. (Ed.), *Perspectives on imitation: From neuroscience to social science. Imitation, human development, and culture* (pp. 55--77): Cambridge, MA: MIT Press.

Meltzoff, A. N. (2007). The framework for recognizing and becoming an intentional agent. *Acta Psychol Amst, 124*(1), 26 - 43.

Mikhail, J. (2007). Universal Moral Grammar: Theory, Evidence, and The Future. *Trends in Cognitive Sciences, 11*(4), 143-152.

Newman, J. O. (2006). Quantifying the standard of proof beyond a reasonable doubt: a comment on three comments. *Law, Probability and Risk, 5*(3-4), 267-269.

Niemela, I., & Simons, P. (1997). *Probabilistic Reasoning with Answer Sets.* Paper presented at the LPNMR4.

Osborne, M. J. (2004). *An introduction to game theory*: Oxford University Press.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*: Morgan Kaufmann.

Pearl, J. (2000). *Causality: Models, Reasoning, and Inference*: Cambridge University Press.

Pereira, L. M., Dell'Acqua, P., & Lopes, G. (2012). Inspecting and Preferring Abductive Models *Handbook on Reasoning-based Intelligent Systems*: World Scientific Publishers.

Pereira, L. M., & Han, T. A. (2009a). *Evolution Prospection.* Paper presented at the Proceedings of International Symposium on Intelligent Decision Technologies (KES-IDT'09).

Pereira, L. M., & Han, T. A. (2009b). Evolution Prospection in Decision Making. *Intelligent Decision Technologies, 3*(3), 157-171.

Pereira, L. M., & Han, T. A. (2009c). *Intention Recognition via Causal Bayes Networks plus Plan Generation.* Paper presented at the Progress in Artificial Intelligence, Proceedings of 14th Portuguese International Conference on Artificial Intelligence (EPIA'09), Evora, Portgual.

Pereira, L. M., & Han, T. A. (2011a). Elder care via intention recognition and evolution prospection *Proceedings of the 18th international conference on Applications of declarative programming and knowledge management (INAP), select extended paper* (Vol. LNAI 6547, pp. 170-187). Evora, Portugal: Springer.

Pereira, L. M., & Han, T. A. (2011b). Intention Recognition with Evolution Prospection and Causal Bayes Networks *Computational Intelligence for Engineering Systems 3: Emergent Applications* (pp. 1-33): Springer

Pereira, L. M., & Lopes, G. (2009). Prospective Logic Agents. *International Journal of Reasoning-based Intelligent Systems (IJRIS), 1*(3/4), Inderscience-208.

Pinker, S., Nowak, M. A., & Lee, J. J. (2008). The logic of indirect speech. *Proceedings of the National Academy of Sciences of the United States of America, 105*(3), 833-838.

Pynadath, D. V., & Wellman, M. P. (1995). *Accounting for Context in Plan Recognition, with Application to Traffic Monitoring.* Paper presented at the Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI'95).

Radke, S., Guroglu, B., & de Bruijn, E. R. A. (2012). There's Something about a Fair Split: Intentionality Moderates Context-Based Fairness Considerations in Social Decision-Making. *PLoS ONE, 7*(2).

Rao, A. S., & Georgeff, M. P. (1991). *Modeling Rational Agents within a BDI-Architecture.* Paper presented at the Proceedings of the second International Conference of Principles of Knowledge Representation and Reasoning.

Rao, A. S., & Georgeff, M. P. (1995). *BDI Agents: from theory to practice.* Paper presented at the Proceeding of First International Conference on Multiagent Systems.

Robson, A. (1990). Efficiency in Evolutionary Games: Darwin, Nash, and the Secret Handshake. *Journal of Theoretical Biology, 144*(3), 379-396.

Roy, O. (2009a). Intentions and interactive transformations of decision problems. *Synthese, 169*(2), 335-349.

Roy, O. (2009b). *Thinking before Acting: Intentions, Logic, Rational Choice.*

Roy, P., Bouchard, B., Bouzouane, A., & Giroux, S. (2007). *A hybrid plan recognition model for Alzheimer's patients: interleaved-erroneous dilemma.* Paper presented at the Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology.

Russell, S. J., & Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*: Pearson Education.

Sadri, F. (2008). *Multi-agent ambient intelligence for elderly care and assistance.* Paper presented at the Proceedings of International Electronic Conference on Computer Science.

Sadri, F. (2010). Logic-based Approaches to Intention Recognition, *Handbook of Research on Ambient Intelligence: Trends and Perspectives* (pp. 375).

Sadri, F. (2011a). Ambient intelligence: A survey. *ACM Comput. Surv., 43*(4), 1-66.

Sadri, F. (2011b). *Intention Recognition with Event Calculus Graphs and Weight of Evidence.* Paper presented at the In Proceedings 4th International Workshop on Human Aspects in Ambient Intelligence.

Searle, J. R. (1995). *The Construction of Social Reality*: New York: The Free Press.

Searle, J. R. (2010). *Making the Social World: The Structure of Human Civilization*: Oxford University Press.

Sigmund, K. (2010). *The Calculus of Selfishness*: Princeton University Press.

Singh, M. P. (1991). *Intentions, Commitments and Rationality.* Paper presented at the In 13th Annual Conference of the Cognitive Science Society.

Sukthankar, G. R. (2007). *Activity Recognition for Agent Teams.* Unpublished PhD Thesis, Carnegie Mellon University, USA

Sukthankar, G. R., & Sycara, K. (2008). *Robust and efficient plan recognition for dynamic multi-agent teams.* Paper presented at the Proceedings of International Conference on Autonomous Agents and Multi-Agent Systems.

Swift, T. (1999). Tabling for non-monotonic programming. *Annals of Mathematics and Artificial Intelligence, 25*(3-4), 240.

Tahboub, K. A. (2006). Intelligent Human-Machine Interaction Based on Dynamic Bayesian Networks Probabilistic Intention Recognition. *J. Intell. Robotics Syst., 45*, 31-52.

Tomasello, M. (1999). *The Cultural Origins of Human Cognition*: Harvard University Press.

Tomasello, M. (2008). *Origins of Human Communication*: MIT Press.

Trivers, R. (2011). *The Folly of Fools: The Logic of Deceit and Self-Deception in Human Life*: Basic Books.

Tu, P. H., Son, T. C., & Baral, C. (2007). Reasoning and Planning with Sensing Actions, Incomplete Information, and Static Causal Laws using Answer Set Programming. *Theory and Practice of Logic Programming, 7*(4), 377 - 450.

Tu, P. H., Son, T. C., Gelfond, M., & Morales, A. R. (2011). Approximation of action theories and its application to conformant planning. *Artificial Intelligence, 175*(1), 79 - 119.

van Hees, M., & Roy, O. (2008). Intentions and plans in decision and game theory *Reasons and intentions* (pp. 207 - 226): Ashgate Publishers.

Wikipedia Burglar Alarm: http://en.wikipedia.org/wiki/Burglar_alarm. Last access: June 18, 2012.

Woodward, A. L., Sommerville, J. A., Gerson, S., Henderson, A. M. E., & Buresh, J. (2009). The Emergence of Intention Attribution in Infancy *The Psychology of Learning and Motivation* (Vol. 51, pp. 187 - 222): Academic Press.

Wooldridge, M. (2000). *Reasoning About Rational Agents*: MIT Press.

Wooldridge, M. (2002). Reasoning about Rational Agents. *The Journal of Artificial Societies and Social Simulation, 5*(1).

XSB (2009). The XSB System Version 3.2 Vol. 2: Libraries, Interfaces and Packages.

Young, L., & Saxe, R. (2011). When ignorance is no excuse: Different roles for intent across moral domains. *Cognition, 120*(2), 202 - 214.

KEY TERMS & DEFINITIONS

1. **Intention recognition**: It is to infer an agent's intentions (called "*individual intention recognition*") or intentions of a group of agents (called "*collective intention recognition*") through its/their observed actions and effects of actions on the environment
2. **Intention-based decision making**: The decision making process that takes into account intentions of other agents in the environment. Technically, intentions of others are now part of the constructs of decision making, such as goals and preferences.
3. **Evolution Prospection**: A decision making system that is designed and implemented based on the idea that, when making some decision at the current state for solving some current goals, one usually takes into account longer-terms goals and future events.
4. **Ambient intelligence**: This refers to electronic environments that are sensitive and responsive to the presence of people. In an ambient intelligence world, devices work in concert to support people in carrying out their everyday life activities, tasks and rituals in easy, natural way using information and intelligence that is hidden in the network connecting these devices.
5. **Moral reasoning**: Moral reasoning can be defined as being the process in which an individual tries to determine the difference between what is right and what is wrong in a personal situation by using logic.