Beyond obscurantism and illegitimate curiosity: how to be transparent only with a restricted set of trusted agents

Nicolas Cointe and Amineh Ghorbani and Caspar Chorus

Delft University of Technology, Faculty of Technology, Policy and Management, Delft, The Netherlands
{Nicolas.Cointe, A.Ghorbani, C.G.Chorus}@tudelft.nl

Abstract. Plans and goals recognition algorithms provide hypotheses about an agent's motivations from an observed behavior. As these techniques are useful to detect maleficent agents, they might also be used to threaten privacy and jeopardize their strategy. But having in mind the existence of an observer, an agent may willingly modify her behavior to clarify her intentions or *a contrario* to obfuscate her underlying goals. This research agenda paper explores the problem of goals inference, transparency and obfuscation through a framework based on the introduction of this dimension in the agent's decision making process into the classic BDI architecture. Then, we highlight the need of explainability to make the resulting behavior as understandable as possible to a set of trusted artificial agents or human users and raise several questions. A proof of concept illustrates the implementation of this model in a simple scenario.

Introduction

Observing the behavior of the agents through a shared environment in a multi-agent system is often mentioned as a necessity to identify threats, misconducts or any improper goal and eventually react in an appropriate way. This task is considered as a fundamental problem in AI with many practical applications as computer network security [5], software user assistance [8] or dangerous people detection in a crowd [16]. Plan recognition algorithms have been proposed to get this information and provide a set of explanations for an observed behavior. Being able to verify the coherency of a behavior according with a set of acceptable intendable goals is considered as a plus for the supervision of such systems. In such context, agents may be interested in optimizing the conspicuity of their goals to be not considered as a threat. But in many application domains, especially where privacy and safety of the users is involved, being able to identify the goals of the others might be considered as intrusive and maleficent.

Being conscious that their behavior may reveal their goals to an observer, agents may need to control the given information through their observable behavior. Our proposition lies in a mechanism designed to provide an evaluation

of the quantity of given information and to integrate this consideration in the decision making process of an agent. To that end, the agent perceives her own behavior through the environment and anticipate the impact of a decision on the given information to an observer. This knowledge is used as an element of the context in the decision making process of an agent to obfuscate her goals (i.e. minimize the given information) or, at the opposite, increase the transparency (i.e. maximize the given information). However, obfuscation might be considered as a drawback when agents are looking for others to coordinate with, and create a trust-based group to cooperate. We highlight here the need to let agents explain their plans to the trusted collaborators. This ability seems also relevant to keep the agent's intentions as clear as possible towards a human user.

Section 1 introduces the problem of goal inference and sequential plan recognition as a set of concepts and techniques used to identify the most probable goal of an agent, based on a set of observations. Section 2 presents our model of obfuscation and transparency-based decision making and illustrates the model with a proof of concept implemented with the JaCaMo Framework to show how it might be implemented, and evaluates the efficiency of this model. And Section 3 extends the framework to a trust-based cooperation mechanism and discusses about the need of explainability towards trusted agents.

1 Background

This section provides an overview of the *Plan Recognition* branch of Artificial Intelligence, and the way to model and analyze an observed behavior in order to infer the most probable goal of an agent[10]. An example illustrates these concepts with a simple scenario used all along this article in order to instantiate the provided definitions and to illustrate the mechanisms.

To identify the goals and plans used by an agent with an observation and interpretation of its behavior, the Plan Recognition (PR) methodology proposes a way to select the most likely goal according with a shared plan library [9,13]. This approach consists in evaluating the likelihood of each plan to identify the most probable goal (MPG) intended by the acting agent. The observer compares the behavior (i.e. an ordered set of actions) of an observed agent with a plan library. A plan library is defined by Kabanaza et al. as a tuple $L = \langle A, G, I, R \rangle$ with A a set of actions, G a set of goals, $I \subseteq G$ a set of intendable goals and R a set of goal-decompostion rules defined as $g \rightarrow \tau$ meaning that the goal g can be accomplished by achieving each element of the partially ordered string τ over the alphabet $(A \cup G)$ represented as a pair $[\beta, C]$. $\beta \in (A \cup G)*$ represents a string of goal and action symbols and C is a set of ordering constraints. A constraint (i, j) means that the i^{th} symbol of β must be achieved or executed before the j^{th} symbol of β.

This definition is only a conceptual view of the observer and is used to evaluate the similarity between an observed behavior and a set of plans, even if the observed agent is not really using this sort of plans (for instance, if the observed agent is a human being) or if the actions are not entirely observable (e.g. if they

are internal actions or if agents only have a local perception). We denote $\mathcal{A}$ the set of the agents and b_a a sequence of actions executed by an agent $a \in \mathcal{A}$ (called behavior of a).

A common way to illustrate a plan library is to build a *plan forest* where plans are displayed as AND-OR trees, where leafs are actions, internal nodes are goals, AND-nodes have an arc across the lines to their children while OR-nodes don't, and constraints are represented with arrows from the first element of a constraint to the second one. An example is graphically displayed page 3 to illustrate a simple plan library.

An *explanation* of an observed behavior b_a is defined [5] as a minimal forest of plan trees with pending sets recorded for each AND-node and OR-node sufficient to allow the assignment of each observation to a specific action in the plans.

Considering these concepts, a *Plan recognition problem* PR [13] is defined by a tuple $\langle L, b \rangle$. A PR solving algorithm takes the plan library L and an observed sequence of actions b, and returns a set of explanations consistent with the observations. Many efficient algorithms have been proposed in the literature to solve PR problems, such as ELEXIR[4], PHATT[5], YAPPR[6], DOPLAR[9] and SLIM[12].

Example 1. Let us consider a simple example, where agents are workers in the second floor of an office building. They are able to use their perception function to perceive the behavior of the other agents on the same floor and they share a plan library L. We consider in this example three different intendable goals able to motivate an agent to go out of her office on the second floor to have a snack break, a coffee break or improve the state-of-the-art of their next paper with a book. The shared plan library describes the knowledge on the different ways to achieve these goals.

The shared plan library L contains the following elements :

- A , the set of actions, is defined as $A =$ { Go to ground floor, Go to second floor, Go to third floor, Buy a coffee, Drink a coffee, Buy a snack, Eat a snack, Take a book, Read the book };
- G , the set of goals, is defined as $G =$ { Have a coffee break, Find a coffee machine, Enjoy coffee anywhere, Have a snack, Find a vending machine, Enjoy snack anywhere, Improve the state-of-the-art, Read anywhere };
- $I \subseteq G$, the set of intendable goals, is defined as $I =$ { Have a coffee break, Have a snack, Improve the state-of-the-art };
- R , the set of goal-decomposition rules. The set used in this example is too big to be detailed here, but contains such rules as

$$\begin{aligned} \text{Have a coffee break} \leftarrow [&\text{Find a coffee machine,} \\ &\text{Buy a coffee,} \\ &\text{Enjoy coffee anywhere,} \\ &\{(1,2),(2,3)\}] \end{aligned}$$

meaning that the goal "Have a coffee break" can be accomplished by achieving or executing the elements of the set $\beta =$ {Find a coffee machine, Buy a coffee,

Enjoy coffee anywhere}, in any order consistent with the set of constraints $C = \{(1,2),(2,3)\}$. This plan library also include alternatives such as the following pair of goal-decomposition rules :

$$\text{Find a coffee machine} \leftarrow [\text{Go to ground floor}, \varnothing]$$
$$\text{Find a coffee machine} \leftarrow [\text{Go to third floor}, \varnothing]$$

meaning that the goal "Find a coffee machine" can be accomplished by achieving one of the two actions mentioned here.

Figure 1 illustrates the plan library mentioned in this example as a forest of three plan trees. The first plan tree describes for instance how to have a coffee break by going to a coffee machine on the ground floor or third floor, then buying it, and finally drinking it before or after coming back to the office on the second floor.

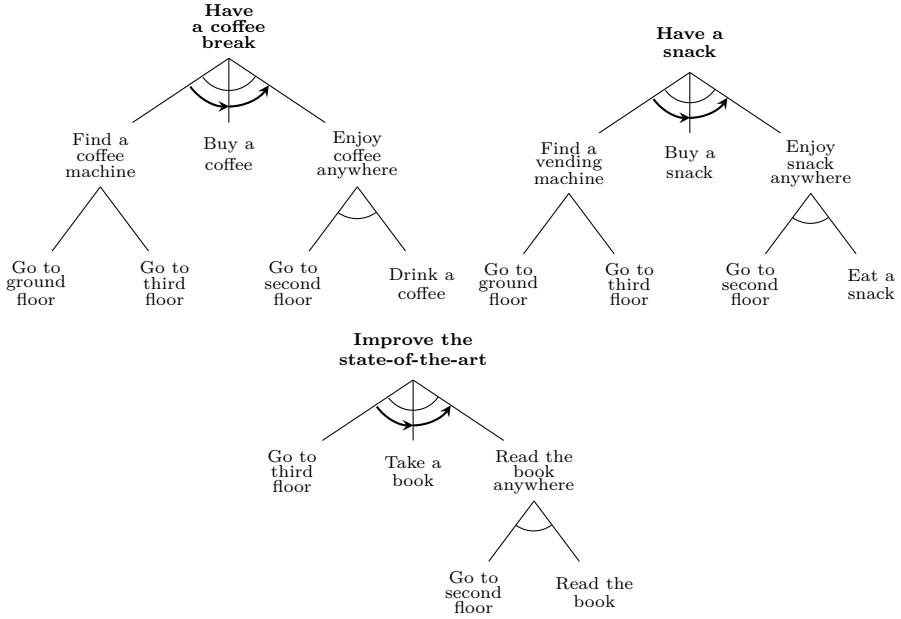

Fig. 1: Example of a plan library : plans to have a coffee break, a snack or improve the state-of-the-art

Let us now consider an agent staying on the second floor and observing the behavior of the others. We consider a first observed behavior b_{ag_i} executed by an agent $ag_i \in \mathcal{A}$ such as $b_{ag_i} = \{$ Go to ground floor, Go to second floor $\}$ and a second observed behavior b_{ag_j} executed by an other agent $ag_j \in \mathcal{A}$ such as $b_{ag_j} = \{$ Go to third floor, Go to second floor, Read the book $\}$.

According with the plan library, two of the three plan trees may produce a behavior including the actions of b_{ag_i} as they both contain these leaves and this order is permitted by the constraints. However, in the case of b_{ag_j}, the plan tree with the root goal "Improve the state-of-the-art" is the only one able to produce this behavior as the action "Read the book" $\in b_{ag_j}$ is absent of the other plan trees.

2 Obfuscation and transparency in BDI

We introduced in Section 1 the PR problem and provided a simple example to illustrate the Plan-Library approaches, a set of classic plan recognition methodologies in artificial intelligence to identify the possible plans and intended goals of an agent according to an observed behavior.

As an observer agent may use these techniques to analyze a behavior, an observed agent may take in consideration the information provided to the observer through the observations in order to obfuscate [3] her goals or at the opposite, make it as transparent as possible.

Obfuscation is neither considered here as obscurantism or deception. Agents may both do obfuscation and be able to explain their decisions to a user or a trusted agent. It is a manner to have more control and select who you want (or should) share information with. Obfuscation must also be distinguished from deception as it does not broadcast fake information or leads to delude observers, but cares about (and minimizes) the information given to unwanted observers.

The following section introduces and illustrates a mechanism to introduce the concept of obfuscation and transparency in the decision making process of a BDI autonomous agent.

The previous section presented the problem of acting in a system were agents are simultaneously acting and observing the others. This section provides the formal definition of a set of predicates to introduce plan recognition and obfuscation in the BDI framework. Section 2.1 proposes a mechanism design to infer the most probable plan from an observed behavior. Section 2.2 shows how agent may use this knowledge in their decision making process to obfuscate as much as possible their goals.

2.1 Interpreting an observed behavior

We consider an autonomous agent perceiving the environment (totally or partially) through a perception function and observing several actions executed by an agent ag_i. We denote b_{ag_i} the set of these observed actions.

In order to infer the most probable plan associated with this observed behavior, we introduce the set $\mathcal{E}_{b_{ag_i}}$ of the explanations of the behavior b_{ag_i} according with the shared plan library L and produced by an explanation function EF such as

$$EF : I \times b_{ag_i} \rightarrow \mathcal{E}_{b_{ag_i}}$$

And $e_j \in \mathcal{E}_{b_{ag_i}}$ is set of pairs of an observed action $a_{ag_i,t}$ associated with an intendable goal $g \in I$. An intuitive, but inefficient approach to evaluate the set of the possible explanations require to explore the space of all the possible combinations of pairs. The size of this space is $\mathcal{O}(|I|^{|b_{ag_i}|})$, that makes any simple implementation suffers from a combinatorial explosion. Many efficient algorithm are building incrementally the explanation set and updating it during the observation [1], and tends to eliminate the most unlikely hypothesis at each step [9].

However, any combination of an actions $a \in A$ and intendable goal $ig \in I$ cannot be considered as part of an explanation if it violates at least one of the following constraints :

- The observed action is a leaf of ig's plan tree;
- There is no action a' previously observed associated to ig with an ordering constraint such as a must be executed before a'.
- There is no action a' previously observed associated to ig supposed to be in an alternative subtree. In other words, there is no a' already associated to ig such as a and a' are in two different subtrees of an OR-node.

Example 2. Let us illustrate this with the example presented in Section 1, and imagine an agent staying in front of the elevator of the second floor to observe some actions of the others and for instance tries to find several explanations to the observed behavior $b_{ag_i} = \{$ Go to third floor, Go to second floor $\}$.

As an example, $\{\langle$ Go to third floor, Have a coffee break $\rangle$, $\langle$ Go to second floor, Have a coffee break $\rangle\ \} \in \mathcal{E}_{b_{ag_i}}$ is a valid explanations as all the actions of the observed behavior b_{ag_i} are also in the plan tree of the intendable goal "Have a coffee break" and the constraints are respected.

Another valid explanation may be $\{\langle$ Go to third floor, Have a coffee break $\rangle$, $\langle$ Go to second floor, Improve the state-of-the-art $\rangle\ \} \in \mathcal{E}_{b_{ag_i}}$ as the constraints are respected. As an agent may have several goals to achieve, her behavior may be explained with a combination of plan trees.

We define the *Most Probable Goal* (or MPG) as the intendable goal which is the root node the most associated with actions in the set $\mathcal{E}_{b_{ag_i}}$ of all the possible explanations for the behavior b_{ag_i}. As a naive measure of the *likelihood* of this plan, we may consider the proportion of action-goal pairs containing the MPG in $\mathcal{E}_{b_{ag_i}}$.

Finally, an *onlooker* agent is defined as an agent that infers and updates, for each action executed by an agent ag_i and perceived through her perception function, such belief as `isTheMPGof(`ig `,` ag_i `)`. This predicate mean that $ig \in I$ is considered at this moment as the Most Probable Goal of the agent ag_i according with the reasoning process defined in this section.

2.2 Obfuscation-based decision making process

In Section 1 we presented some strategies to recognize the plans executed by an agent according with an observed behavior. In Section 2.1 we adapted these

methods in the context of an onlooker agent. We propose here a decision making process in order to both achieve the agent's goals and obfuscate or clarify it as much as possible from an onlooker's perspective.

We consider a set $\mathcal{A}$ of agents based on a BDI architecture [17]. Each $a \in \mathcal{A}$ of the system is described with :

- $\mathcal{G}_a$ a set of active goals to achieve;
- $\mathcal{B}_a$ a belief base, containing the beliefs of the agent, obtained from both the perception function of the agent and her own reasoning;
- $\mathcal{P}_a$ a plan library containing the plans the agent knows.

A plan is defined as $p ::= te : ct \leftarrow h$ with te a triggering event (for instance a belief addition generated by a perception function, of a goal addition generated by an other plan), ct a context where the plan is feasible described through a conjunction of atomic formulas, and h is the body of the plan, made of a sequence of internal actions (belief addition, goal addition) or external actions (interaction with the environment, interaction with other agents). Through the description of plans, the designer of the agent defines how the agent will make decisions, depending on its own beliefs, in order to achieve goals. In this paper, we consider the shared plan library L (see definition in Section 1) as a shared knowledge in the system. And as L describes the preconditions of actions, an agent a may have additional conditions, and its own decision making process (for instance to decide which subtree is the most interesting in case of OR-nodes in a plan tree).

In order to integrate the impact of a decision for a potential observer, we provide here a new process to add in the context of a plan some additional information and introduce the predicates $maximizeEntropy(c, t, O, b, h_{c,t,b})$ and $minimizeEntropy(c, t, O, b, h_{c,t,b})$ with the following arguments :

- c is a choice, such as $c \in A \cup G$. In a plan tree, such choice might be any child node of an OR-node ;
- t the moment when the choice is supposed to be executed (or at least the first action, i.e. the leftmost leaf, of the corresponding subtree if they are several actions);
- O the set of options, or possible choices, such as $O \subseteq A \cup G$ and $c \in O$;
- b the past behavior of the agent until t ;
- $h_{c,t,b}$ the shannon entropy of the likelihood (see Section 2.1) of all the intendable goals if the choice c is executed from t and observed in addition with the behavior b.

The predicate $maximizeEntropy(c, t, O, b, h_{c,t,b})$ is true iff $\nexists c' \in O$ such as $h_{c',t,b} > h_{c,t,b}$. Respectively, the predicate $minimizeEntropy(c, t, O, b, h_{c,t,b})$ is true iff $\nexists c' \in O$ such as $h_{c',t,b} < h_{c,t,b}$. For a decision maker, this knowledge indicates if a choice, according with an existing behavior observation and a set of alternative options, is the most appropriate to respectively obfuscate or make transparent the intended goal.

Using this predicate in the context of a plan allows agents to be informed on the way to achieve a goal that maximize or minimize entropy. Obfuscation-based decision makers maximizes entropy for all the intendable goals of I. Transparency-based decision maker minimizes the entropy to maximize the likelihood of their intended goal. Here, obfuscation-based decision making differs from deception-based agents as it does not tend to behave such that $P(b_a|ig') > P(b_a|ig)$ with $ig \in \mathcal{I}$ the selected intendable goal the agent actually wants to achieve and $ig' \in \mathcal{I}$ another intendable goal of the library L.

2.3 Proof of concept

To show how this decision process may be implemented in a BDI framework and observe the impact of this process on the agents' behavior, this section describes a proof of concept and a set of experiments using the JaCaMo Framework [2] to describe the agents internal mechanisms and the shared environment. The source code of the experiment is available online[1].

Overview of the experiment

In JaCaMo, agents are described through the AgentSpeak language [18], a prolog-like language providing a simple syntax to describe the use and definitions of rules, beliefs, plans and goals of autonomous agents. The default agent architecture provides the internal mechanisms to perceive the environment, select intentions in the plan library to achieve active goals and execute these intentions.

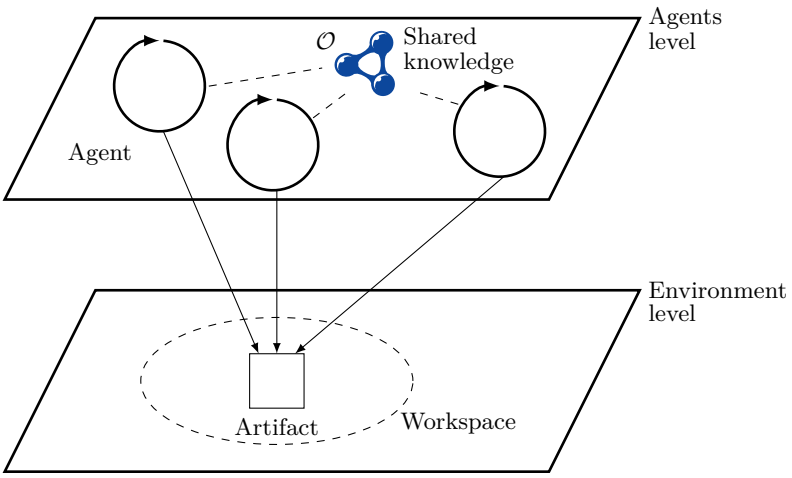

Fig. 2: Overview of the proof of concept

[1] http://www.nicolascointe.eu/projects/

The figure 2 illustrates the multiagent system designed for this experiment, with a single workspace, including a single artifact focused by every agents. Focusing the artifact allows agents to perform the set A of actions presented in Section 1 and notify all the other agents when an action is executed through the signal mechanism provided by CArtAgO. Agents also share knowledge $\mathcal{O}$, containing the shared plan library L. For more convenience, $\mathcal{O}$ is implemented in this experiment as a set of shared beliefs, rules and plans.

To symbolize artificial autonomous agents, we use cycling arrows to represent their internal process, continuously looking for relevant plans to select in order to achieve their goals.

Representing the shared plan library

To make a decision in case of OR-node or evaluate a behavior, agents need to manipulate a representation of the plan trees. To do so, $\mathcal{O}$ contains a shared set of beliefs formatted as in the following example presented in implementation 1.1 to represent the plan tree forest.

```
1  // Name of the leafs (to match with the signals)
2  leaf(buySnack,"buySnack").
3
4  //****** First plan tree ******
5  isIntendableGoal(haveSnack,"haveSnack").
6
7  isSonOf(buySnack,haveSnack,haveSnack).
8
9  // Ordered AND-nodes
10 orderConstraint(findVendingMachine,buySnack,haveSnack).
11
12 // OR-nodes
13 exclusionConstraint(goToGroundFloor,goToThirdFloor,
       haveSnack).
```

Implementation 1.1: Excerpt from planForestDescriptor.asl

Each leaf of the plan tree forest is described as illustrated in the second line of the example. The first element is the symbol, used in the reasoning of the agent, and the second one is a string, to be compared with the signals emitted by the artifact when actions are executed. With a similar belief illustrated with the fifth line, root nodes of plan trees, i.e. intendable goals, are described with both their symbol and the corresponding signal.

All the tree from the root node to the leaves is represented with a set of beliefs such as `isSonOf(A,B,T)` illustrated on line seven. This predicate means that `A` is a child node of `B` in the plan tree `T`.

Order constraints on AND-nodes are also represented, through a set of specific beliefs such as `orderConstraint(A,B,T)` as illustrated on the tenth line, meaning that the subtree represented by node `A` should be executed before `B` in the plan tree `T`. In the same way, OR-nodes are represented through an exclusion constraint between two subtrees of a specific plan tree.

As the set of beliefs previously presented depict the structure of the plan tree forest, $\mathcal{O}$ contains a second file to describe this structure through a set of plans. As the first file is used for reasoning, the second one is required to use the classic plan selection process. The excerpt of implementation 1.2 illustrates how nodes are represented with agentspeak plans.

```
1  +!haveSnack :
2    decideIn(Choice,[goToThirdFloor,goToGroundFloor])
3    <- !findVendingMachine(Choice);
4       !buySnack;
5       !enjoySnackAnywhere;
6       !finishIfIGotSnack.
7
8  +!findVendingMachine(goToGroundFloor) :
9    baseWaitingTime(T)
10   <- .my_name(Name);
11      goToGroundFloor(Name);
12      .wait(T);
13      .print("I move to the ground floor").
14
15 +!findVendingMachine(goToThirdFloor) :
16   baseWaitingTime(T)
17   <- .my_name(Name);
18      goToThirdFloor(Name);
19      .wait(T);
20      .print("I move to the third floor").
```

Implementation 1.2: Excerpt from planLib.asl

The first plan calls a generic predicate `decideIn(C,L)` unifying a chosen node `C` with an element in a list `L` representing the different subtrees available from an OR-node. This predicate is implemented in various ways, depending on the agents types (see Section 2.3 - *Decision process*). As each node is implemented here as a "Jason plan", subgoals are triggered to explore the subtrees (see for instance lines three to 6 in this example). When a plan representing a leaf node is executed, the agent executes an action on the artifact and wait a few seconds to let the others perceive and react to this action.

Evaluating behaviors

When an action is executed, the artifact broadcast a signal mentioning the name of the agent performing the action and the action itself. On this signal, an observer agent updates a belief `behavior(A,B)` with `A` the agent's name and `B` the observed behavior as a list of pairs of actions and the corresponding moment of execution.

```
1  +!reactToBehavior(Agent) : behavior(Agent,Behavior)
2      <- !detailBehaviorExplanation(Agent,Behavior).
3
4  +!detailBehaviorExplanation(Agent,Behavior):
5    listOfIntendableGoals(IGs)
6    & time(T)
7    & exploreExplanations(Behavior, [],Exp, IGs)
8    & countPlansInExp(Exp,[],PlansOccurrences)
9    & searchMostProbableGoal(PlansOccurrences,P,N)
10   & getSpecificCountPlanInExplanationSet(haveCoffeeBreak,
       PlansOccurrences,NofHaveCoffeeBreak)
11   & getSpecificCountPlanInExplanationSet(haveSnack,
       PlansOccurrences,NofHaveSnack)
12   & getSpecificCountPlanInExplanationSet(improveSOTA,
       PlansOccurrences,NofImproveSOTA)
13   <-   .length(Exp, L);
14        jia.shannonEntropy(H,NofHaveCoffeeBreak,NofHaveSnack,
       NofImproveSOTA);
15        .print("Agent=", Agent, " T=", T," MPG=", P," Entropy
       =", H, "(" , NofHaveCoffeeBreak, ",", NofHaveSnack, ","
       , NofImproveSOTA, ")");
16        .print("I can explain the behavior ", Behavior, " of
       agent ", Agent, " with the ", L ," possible
       explanations : ", Exp).
```

Implementation 1.3: Excerpt of onlooker.asl

The onlooker agent analyzes this behavior with the plan presented in Implementation 1.3. The predicate `exploreExplanations(Behavior, [],Exp, IGs)` on the seventh line unifies `Exp` with an exhaustive list representing the set $\mathcal{E}$ of all the possible explanations (see Section 2.1).

Then the predicate `countPlansInExp(Exp,[],PlansOccurrences)` unifies `PlansOccurrences` with a list containing all the plan trees associated with their occurrence in the set `Exp` of all the possible explanations. The predicate `searchMostProbableGoal(PlansOccurrences,P,N)` unifies `P` with the MPG and the values of both entropy and individual plans probabilities are printed to produce the post-experimental results analysis shown in the next section.

Decision process

As mentioned in the description of the plan library, the definition of the predicate `decideIn(C,L)` depends on the agent type.

```
1 decideIn(Choice,Options):-
2   myName(Name)
3   & myGoal(Tree)
4   & time(T)
5   & behavior(Name,Behavior)
6   & makeEntropyList(T,Options,Tree,Behavior,EntropyList)
7   & pickTheMostObfuscatingOptionIn(EntropyList,Choice,_).
```

Implementation 1.4: Excerpt of obfuscator.asl

Random agents simply pick a random choice `C` in the list of all the possible choices `L`.

Obfuscator agents are initialized with an intended goal and use the predicate presented in implementation 1.4. The predicate used on the sixth line, `makeEntropyList(T,Options,Tree,Behavior,EntropyList)` unifies the argument `EntropyList` with a list of pairs of each option in `Options` associated with the estimated entropy if the corresponding subtree is executed. Then, the last predicate unifies `Choice` with the option associated with the maximal value in this list.

Transparency-based agents use a similar process, but pick the option associated with the minimal entropy value instead of the maximal one.

Results analysis

The simulation is initialized with three random agents, three obfuscator agents and three transparency-based agents assigned to each of the three intendable goals. An onlooker agent is added in order to observe and evaluate all the behaviors simultaneously. We aggregate the results of a set of experiments to let the random agents explore all the possible combinations of choices.

The Figure 3 illustrates the behavior of the agents assigned to the intendable goal “Have a coffee break” from the onlooker perspective.

As expected, the entropy in the likelihood of intendable goals for random agents takes several values (represented by the area on the linechart). On the top of this area, the behavior of obfuscator agents is on the upper limit of the area, due to their decision process. On the opposite, transparency-based agents always follow the bottom of this area.

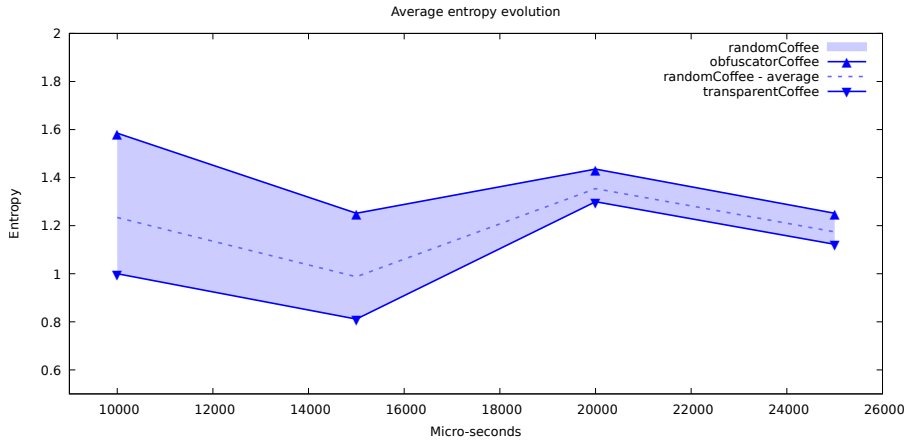

Fig. 3: Evolution of the entropy in the likelihood of plans

3 Obfuscation, cooperation and explainability

Promoting transparency in a society seems common, moral and intuitive, but Section 3.1 presents different models of the human tendency to motivate the avoidance of knowing others' goals and propose to explore the concept of obfuscation-based cooperation. Then Section 3.2 highlight the value of explainability in an organization as a strategic asset.

3.1 Information avoidance and obfuscation-based cooperation

We consider in this paper the case where agents have to collaborate in a shared environment where they are able to observe, at least partially, the behavior of the others. Having in mind these observations, agents may willingly avoid to cooperate with those who do or don't obfuscate their own goals. This restriction makes sense for instance if agents care about privacy, or want to maximize uncertainty on the goal of the agents in the system for a strategic purpose. For instance, this specificity might be relevant in a community of agents to avoid punishment and to face a tyrannical authority by obfuscating the goals of the whole community members. This model is a proposition to design artificial autonomous agents and, even if it is inspired by some strong studies in economy and psychologies, it does not pretend to be a model of any human cognitive process.

In economy, the concept of *Active Information Avoidance* (or AIA) [7] is defined in the case where an agent is aware that the information is available, and has a free access to the information but refuse to perceive it. The authors mention various possible motivations such as disappointment or regret aversion, optimism maintenance, dissonance avoidance (similar to the problem of beliefs consistency in AI) and strategically driven information avoidance. In a multiagent system,

where agents have some responsibilities (for instance due to a role in an artificial organization, or accountability toward a user) but also need to collaborate with the other agents to achieve their goals, it might be preferable and strategically relevant to ask them not to reveal their goals.

The model of *Rational ignorance* [15] proposes a way to represent how a duty-oriented (due to moral responsibility) *homo œconomicus* may prefer to ignore information if it tends to decrease their self-image (a way to evaluate how satisfying their behavior is regarding their duties).

If avoiding an information about the unethical behavior of a collaborator may be a way for a human being to avoid cognitive dissonance [15] or guilt, it is not a solution to avoid social consequences such as being considered culpable of negligence or collusion. To this end we propose to allow an obfuscator agent to collaborate with the others only if they obfuscate their goals enough, no matter how acceptable or not these goals are.

A classic mechanism used in the literature to enable such conditions for a cooperation is the introduction of trust, defined as "a belief in someone else's actions acquired by direct experiences"[14]. An *obfuscation-based trust* is then a belief in the ability and will of another agent to obfuscate her goals. Such belief may be a condition to include an agent in a community of obfuscation-based decision makers.

3.2 Explanation as a strategic asset for an organization

Let us imagine an application where agents are allowed to cooperate, able to hamper the success of the others' plans, and have a motivation to do so (for instance in a cooperative zero-sum game). If the target is engaged in a cooperation, a plan failure may have an impact on the coalition's efficiency. In this case, to hide the goals and selected plans to the opponents seems a major concern. As a counterpart, being able to explain your goals and plans to the coalition might be a requirement to both prevent betrayals from malicious agents and coordinate actions (for instance : resources sharing, avoid deadlocks and so on). To verify the likelihood of a declared explanation and build a trust-based relationship, agents may observe the other members of the coalition and verify that the given explanation is in the set of the possible explanations of the observed behavior. The design of such framework seems an interesting next step in our research.

If the agent is in interaction and collaborate with a human being, the obfuscation mechanism suggested here may lead to a disturbing, unexpected and counter-intuitive user experience. As the legibility of the decision is considered as a welfare for human-autonomous agents interactions [11], it seems necessary to explain the plan and the reasons of its selection to the user. It might be relevant also to alert the user, and eventually stop the execution of the planned actions, if the agent is no longer able to obfuscate her goals. This feature might be especially interesting and legitimate if the purpose of obfuscation in such system is to ensure the protection of the user's privacy.

4 Conclusion and Further Work

We have presented a mechanism to embed a classic plan recognition technique into a BDI agent, not only to evaluate the most probable goal of the others, but also to anticipate their evaluation of the decision maker's behavior and ensure that a decision minimizes or maximizes the given information to the others. The formal model has been defined and illustrated with a very simple example and an operational proof of concept. Results have been interpreted and confirmed the expectations. In the last section, we exposed a set of propositions and questions to explore the impact of obfuscation-based trust mechanisms or information sharing in agents communities, and discussed about the need of explanations to deal with cooperation and maintain a sufficient legibility for a human being user.

Future work will first focus on some extensions, such as a way to deal with partial observability of the environment, i.e. to associate to each action a probability to be perceived or not, depending or not on the context of the execution (for instance, to deal with local visions problems in case of spatial dimension in the environment, or sensors requirement). An interesting contribution might be the development of such slightly different strategies, as real-time entropy maximization or minimization instead of the long-term one presented in this paper and leading in more complex simulations to such counter-intuitive results. We have also willingly let apart in this paper the possibility to execute some actions of an other plan tree in order to increase the entropy. It might be interesting to explore this type of strategy.

Another important suggestion is to deal with proofs of intention. For instance if an agent is observed during the execution of an action which is a leaf in only one plan tree, an observer should directly infer that the root of this plan tree is obviously in the set of her active goals. For the agent executing this action, there is maybe no reasons to obfuscate her behavior anymore if all the observers perceived this action.

In a higher perspective, we want to explore the involvement of the incorporation of such mechanisms in the decision process of every, or at least a subset of the agents. In Section 3 we mentioned several cases where obfuscation seems relevant to model dissonance avoidance and rational ignorance, design obfuscation-based or transparency-based cooperation, or enforce trust with explainability within an organization or towards a user. Being aware of (and able to manage) the information provided to external observers trough the behavior of a set of agents requires to explore a wide set of questions, both from agents-centered and organizations-centered perspectives.

Acknowledgment

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 724431).

References

1. Dorit Avrahami-Zilberbrand and Gal A Kaminka. Fast and complete symbolic plan recognition. In *IJCAI*, pages 653–658, 2005.
2. Olivier Boissier, Rafael H Bordini, Jomi F Hübner, Alessandro Ricci, and Andrea Santi. Multi-agent oriented programming with jacamo. *Science of Computer Programming*, 78(6):747–761, 2013.
3. Caspar G Chorus. How to keep your av on the right track? an obfuscation-based model of decision-making by autonomous agents. *hEART 2018*, 2018.
4. Christopher W Geib. Delaying commitment in plan recognition using combinatory categorial grammars. In *IJCAI*, pages 1702–1707, 2009.
5. Christopher W Geib and Robert P Goldman. Partial observability and probabilistic plan/goal recognition. In *Proceedings of the International workshop on modeling other agents from observations (MOO-05)*, volume 8, pages 1–6, 2005.
6. Christopher W Geib, John Maraist, and Robert P Goldman. A new probabilistic plan recognition algorithm based on string rewriting. In *ICAPS*, pages 91–98, 2008.
7. Russell Golman, David Hagmann, and George Loewenstein. Information avoidance. *Journal of Economic Literature*, 55(1):96–135, 2017.
8. Eric Horvitz, Jack Breese, David Heckerman, David Hovel, and Koos Rommelse. The lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 256–265. Morgan Kaufmann Publishers Inc., 1998.
9. Froduald Kabanza, Julien Filion, Abder Rezak Benaskeur, and Hengameh Irandoust. Controlling the hypothesis space in probabilistic plan recognition. In *IJCAI*, pages 2306–2312, 2013.
10. Henry A Kautz and James F Allen. Generalized plan recognition. In *AAAI*, volume 86, page 5, 1986.
11. Alexandra Kirsch. Explain to whom? Putting the User in the Center of Explainable AI. In *Proceedings of the First International Workshop on Comprehensibility and Explanation in AI and ML 2017*, Bari, Italy, 2017.
12. Reuth Mirsky et al. Slim: Semi-lazy inference mechanism for plan recognition. *arXiv preprint arXiv:1703.00838*, 2017.
13. Reuth Mirsky, Roni Stern, Kobi Gal, and Meir Kalech. Sequential plan recognition: An iterative approach to disambiguating between hypotheses. *Artificial Intelligence*, 260:51–73, 2018.
14. G. Muller, L. Vercouter, and O. Boissier. Towards a general definition of trust and its application to openness in MAS. In *6th Workshop on Deception, Fraud and Trust in Agent Societies*, pages 49–56, 2003.
15. Karine Nyborg. I don't want to hear about it: Rational ignorance among duty-oriented consumers. *Journal of Economic Behavior & Organization*, 79(3):263–274, 2011.
16. Jérémy Patrix, Abdel-Illah Mouaddib, Simon Le Gloannec, Dafni Stampouli, and Marc Contat. Discrete relative states to learn and recognize goals-based behaviors of groups. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 933–940. International Foundation for Autonomous Agents and Multiagent Systems, 2013.
17. Anand S Rao and Michael P Georgeff. Modeling rational agents within a BDI-architecture. pages 473–484, 1991.
18. Renata Vieira, Álvaro F Moreira, Michael Wooldridge, and Rafael H Bordini. On the formal semantics of speech-act based communication in an agent-oriented programming language. *Journal of Artificial Intelligence Research*, 29:221–267, 2007.