Design and Evaluation of Explainable BDI Agents

Maaike Harbers[1,2], Karel van den Bosch[2] and John-Jules Meyer[1]
[1]Utrecht University, P.O.Box 80.089, 3508 TB, Utrecht, The Netherlands
[2]TNO Human Factors, P.O.Box 23, 3769 ZG, Soesterberg, The Netherlands
{maaike,jj}@cs.uu.nl, karel.vandenbosch@tno.nl

Abstract

It is widely acknowledged that providing explanations is an important capability of intelligent systems. Explanation capabilities are useful, for example, in scenario-based training systems with intelligent virtual agents. Trainees learn more from scenario-based training when they understand why the virtual agents act the way they do. In this paper, we present a model for explainable BDI agents which enables the explanation of BDI agent behavior in terms of underlying beliefs and goals. Different explanation algorithms can be specified in the model, generating different types of explanations. In a user study (n=20), we compare four explanation algorithms by asking trainees which explanations they consider most useful. Based on the results, we discuss which explanation types should be given under what conditions.

1 Introduction

Virtual training systems provide an effective means to train people for complex, dynamic tasks like negotiation, crisis management or fire-fighting. Intelligent agents can be used in virtual training to play the persons with whom a trainee interacts in the training task, e.g. opponents, colleagues or teammates. The interaction with intelligent agents prepares the trainee for interaction with real humans in the real world, and therefore the agents should display realistic human behavior. Like real human behavior, human-like agent behavior will not always be understandable. However, to learn from training, it is important that the trainee eventually understands the motives and reasoning behind the behavior of others. A solution is to provide trainees the possibility to request explanations about the agents' actions during or after a training session.

In this paper we present an account of explainable BDI (Belief Desire Intention) agents. We use a BDI-based approach for the following reasons. First, BDI agents can generate explanations that are similar human explanations of behavior. Humans explain and understand their behavior in terms of its underlying desires, goals, beliefs, intentions and the like [4, 11, 14]. BDI agents have explicit representations of goals and beliefs, and they determine their actions by a deliberation process on these mental concepts. When using BDI agents, the mental concepts underlying an action can also be used to explain that action. Second, explanations of BDI agents can clarify typical human errors. Many mistakes in critical situations involve people that make false assumptions about the knowledge and intentions of others [6]. The phenomena of attributing incorrect mental states to others is also well described in the cognitive sciences, e.g. [15, 12]. Explainable BDI agents can make trainees aware of their (false) assumptions about other agents' mental states by revealing the agents' actual ones. Finally, BDI agents have been successfully applied in games [16] and virtual training [21] before.

The explainable agent approach we present involves guidelines for agent design and a module for the generation of explanations. In this explanation module, a behavior log stores all past mental states and actions of an agent that may be needed for explanation. When there is a request for an explanation, an explanation algorithm is applied to the log selecting the beliefs and goals that become part of the explanation. Actions can be explained in different ways, e.g. someone opened a door 'because he believed someone was outside' or 'because he wanted to know who was outside'. One may explain an action by all the underlying beliefs and goals. However, not all 'explaining elements' are equally useful in an explanation [11]. Moreover, it has been shown that most of the time people provide relatively short explanations when asked to explain agent behavior [9]. Therefore, we propose four explanation algorithms which explain an action by a selection among the beliefs and goals that were responsible for it.

Psychological research provides no conclusive answers to which explanation types should be used to explain actions. Malle's framework about how people explain behavior [14] distinguishes four modes of explanation. One mode considers explanations about unintentional behavior and the

978-0-7695-4191-4/10 $26.00 © 2010 IEEE
DOI 10.1109/WI-IAT.2010.115

other three consider explanations about intentional behavior. The explanations we consider are about intentional behavior. The three explanation modes of intentional behavior are reason, causal history, and enabling factors explanations. Reason explanations are beliefs and goals, causal history explanations explain the origin of beliefs and goals, and enabling factors explanations consider the capabilities of the actor. People mostly give reason explanations for intentional behavior. Malle's framework distinguishes beliefs and goals as reason explanations, but does not (yet) describe when and how which goals and beliefs are used.

In this paper we present a user study (n=20) investigating which explanation types trainees consider most useful for different types of actions. For instance, when do trainees prefer goal-based and when belief-based explanations. The subjects in the experiment receive agent actions from a training scenario with different possible explanations for each action. The explanations are generated by different explanation algorithms and thus of a different type. The subjects are asked to indicate for each action which explanation they prefer. Our hypothesis is that different types of actions require different explanation types.

This paper is organized as follows. We start with a discussion on related work in section 2. Then we introduce our account of explainable BDI agents in section 3. Subsequently, we present the user experiment in section 4. We end the paper with a conclusion and plans for future research in section 5.

2 Related work

A lot of research on explanation has been done in the field of expert systems (e.g. [20]). In this section we will focus in particular on the explanation of agent behavior. Haynes et al provide general designs for explaining intelligent agents [10]. They distinguish four types of explanations: ontological explanations, mechanistic explanations, operational explanations and design rationale. Explanations of these types provide, respectively, answers to the following questions: what are its properties?, how does it work?, how do I use it?, and why has it been designed the way it is? This work is different from ours as it focuses on explanations about *agents*, whereas we study the explanation of agent *behavior*. Oh et al have worked on different explanatory styles [17], where explanatory style concerns how people explain good or bad consequences to themselves. Their work differs from ours as explanations in our intended application are given by agents to someone else, the trainee.

The two first systems explaining actual agent behavior are Debrief [13] and XAI [22]. Debrief is implemented as part of a fighter pilot simulation and allows trainees to ask explanations about any of the artificial fighter pilot's actions. To generate an answer, Debrief modifies the recalled situation repeatedly and systematically, and observes the effects on the agent's decisions. Based on the observations, Debrief explains what must have been the factors responsible for the agent's decisions. The XAI component forms part of a simulation-based training for commanding a light infantry company. After a training session, trainees can select a time and an agent, and ask questions about the agent's state, like its location or health. Thus, in short, the XAI system provides information about an agent's physical state, and Debrief provides explanations in terms of an agent's beliefs. Neither of the two systems provides explanations involving the goals and intentions behind an agent's actions.

A more recent version of the XAI system is claimed to overcome the shortcomings of the first. XAI version two supports domain independence, modularity and the ability to explain the motivations behind an agent's actions [7, 2]. To demonstrate its domain independence, the system has been applied to a tactical military simulator, and a virtual trainer for soft skills such as leadership, teamwork, negotiation and cultural awareness. For the generation of explanations, the system depends on information that is made available by the simulation. The developers notice that simulations differ in their 'explanation-friendliness' [1]. At best, agent behavior is represented by goals and the preconditions and effects of actions, and the XAI system can automatically import behaviors. In the worst case, behavior is represented by procedural rules, and a hand-build XAI representation of the behaviors has to be made. A disadvantage of the last approach is that any change in the simulation must be duplicated in the explanation component by the programmer.

We advocate an approach that, like the second XAI system, is independent of the simulation that is used, but avoids double bookkeeping for the programmer. We argue to integrate the development of agents and explanation facilities by specifying agent behavior in a BDI-based agent programming language instead of in the simulation. A BDI representation ensures that agent behavior can be explained by the mental concepts that were actually responsible for the agent's actions. The agents should be connected to the simulation, e.g. by a middle layer in which information from the simulation is translated to a representation fit for agents and vice versa [5]. Keeping the agent specification separate from the simulation makes it reusable in other simulations.

3 Explainable BDI agents

In this section we describe an architecture for explainable BDI agents, involving a BDI agent and an explanation module. In section 3.1, we summarize earlier work on explainable agents [8]. This work discusses guidelines for the design of explainable BDI agents, and demonstrates how an explainable agent model can be implemented in the BDI-based agent programming language 2APL [3]. Subsequently, we introduce an explanation module that can generate explanations about agent behavior when connected to a BDI agent. The module consists of a behavior log in which the history of mental states of the agent are stored, and one or more explanation algorithms that are applied to the behavior log when there is a request for an explanation. We implemented the explanation module as a 2APL environment. As 2APL environments are implemented in Java, the module can easily be connected to other BDI languages that are also built on Java. In section 3.2 we discuss the organization of the behavior log, and in section 3.3 we present four explanation algorithms.

3.1 Agent design

When connecting behavior generation and explanation, the purpose to explain an agent's behavior should be taken into account during its design. Namely, the information that is needed to explain an agent's actions must be available. Thus, to obtain useful behavior explanations in terms of beliefs, goals and intentions, agents must have meaningful, explicit representations of these mental concepts.

To obtain a BDI specification of the (human) role which an agent is supposed to play, we use a goal hierarchy as an intermediate step. In a goal hierarchy, one main goal is divided into subgoals, which are in turn divided into subgoals, until atomic actions are reached. Furthermore, the conditions under which tasks are adopted and achieved are specified. We use a hierarchical goal model because hierarchical task analysis (HTA) is a well established technique for cognitive task analysis in cognitive psychology, and has proven to be appropriate for the specification of complex human tasks [19]. Moreover, task hierarchies have similarities with BDI models [18]. Instead of tasks, we specify goals. Adoption conditions can be seen as the beliefs of an agent. Thus, goal hierarchies are translatable to BDI agents, e.g. to 2APL agents as shown in [8].

Figure 1 shows an example of a goal hierarchy with an agent's goals, beliefs and actions. The main goal A is divided into subgoals B, C and D, and subgoal C is divided into actions E, F and G. The goal hierarchy is not complete because goal B and D are not divided into subgoals or actions here. The circles in Figure 1 denote the possible beliefs of the agent. Only when the agent has the belief above a subgoal or action, it can adopt that subgoal or action. For example, action E can only be executed if the agent believes e. Furthermore, the adoption of subgoals and actions depends on the relation of a goal to its subgoals or actions. A relation of the type *all* means that all subgoals or actions must be adopted. A relation of the type *if* means that all *applicable* subgoals or actions must be adopted. Dependent on the environment, this may be all or none of the subgoals or action, or something in between. A relation of the type *one* implies that exactly one of the subtasks has to be adopted. The term *seq* refers to sequential and it means that all subtasks must be adopted, but one by one and in a specific order.

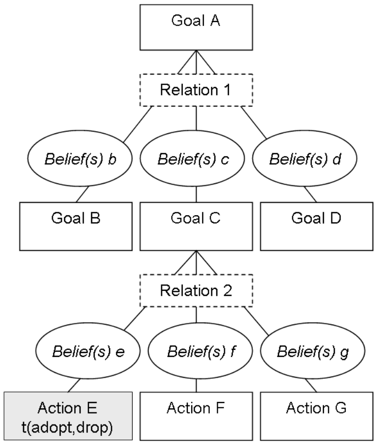

Figure 1. Goal hierarchy of an explainable BDI agent.

The agent can perceive its environment, and changes in the environment lead to changes in its beliefs. Thus, when an explainable BDI agent is executed, circumstances in the agent's environment determine, via the agent's beliefs, how it ”walks through” the hierarchy. To provide explanations about the actions that are executed, this path with reasoning steps and actions needs to be remembered. Therefore, this information is updated from the agent to the behavior log in the explanation module.

3.2 Behavior log

The explanation module can generate explanations about the behavior of agents with different goal hierarchies, implemented in different BDI-based programming languages. But though the updates to the behavior log may differ per agent, the structure of the updates has to be the same. A fixed update structure ensures that the agents' updates are 'understood' by the explanation module, or in other words, that they fit in the organization of the behavior log. For each goal or action that an agent adopts, the behavior log is updated with the following information.

<goal/action, parent, relation, [belief 1, ..., belief n], [child 1, ..., child n], time(adoption, dropping)>

The update is a tuple containing, in this order, the identity of the goal or action (e.g. action E), the identity of its parent goal (goal C), the relation between the goal/action and its parent (relation 2), its adoption conditions (belief(s) e), the identities of the child goals/actions of the parent goal (action E, F and G), and the time at which is was adopted and dropped (t(adopt,drop)). Note that the list with children also includes the goal/action itself. The order of execution of the children is maintained in the behavior log.

An agent's goal hierarchy contains all goals and actions that the agent can *possibly* adopt. Consequently, in the BDI implementation, all goals that may be adopted are present in the plan library of the agent. A behavior log, in contrast, does not contain a specification of the agent's goal hierarchy. Before execution of the agent, the behavior log is empty. After execution of the agent, the behavior log contains a set tuples containing the goals/actions that the agent *actually* adopted and performed. The goals and actions in the behavior log are equal to, or a subset of the goals and actions in an agent's goal hierarchy. With the information in the behavior log, the 'used' part of the goal hierarchy can be reconstructed.

3.3 Explanation algorithms

The behavior log is built up during the execution of an agent and may differ for each agent and each session. The explanation algorithms are fixed and can be applied to different behavior logs. When a user sends a request for an explanation about a certain agent action to the explanation module, one of the explanation algorithms is applied to select information from the behavior log. The result of this process, an explanation for the given agent action, is presented to the user.

An action can be explained by the beliefs and goals underlying that action. For example, in the goal hierarchy in Figure 1, action e can be explained by goal A, goal C, belief(s) c and belief(s) e. However, providing the whole trace of beliefs and goals responsible for an outcome, in particular with big task hierarchies, often does not create a useful explanation. Therefore, explanation algorithms not only select which goals and beliefs are responsible for an action, but also make a selection among those beliefs and goals.

In an earlier experiment, we asked experts to explain actions of agents in a virtual training scenario [9]. From their answers, we identified the following five explanation types: goals that are achieved by an action, goals or actions that becomes achievable, the inducement of an action, background information, and beliefs about other agents' goals. The first two explanation types can be modeled as the goals and the last three can be modeled as the beliefs of a BDI agent. In a second experimentation session, a new group of experts was asked to indicate their preferred explanation for the same set of actions. They could choose between a low-level goal that was achieved by the action, a high-level goal that was achieved by the action, a low-level belief that enabled the adoption of the action, a high-level belief that enabled the adoption of the action, and an action or goal that became achievable by execution of the action. Except for high-level beliefs enabling the adoption of an action, all other explanation types were sometimes preferred. However, from the results we could not infer which explanation type is preferred for which action type because we did not make a categorization of different actions.

In section 4 we present a new study in which the actions to be explained are categorized into four different types. The results can show whether which explanation types are preferred for which action types. To generate the explanations for the experiment, we used the same algorithms as in the previous study, except for high-level beliefs. In the following definitions, the A in action/goal$_A$ is a variable denoting the identity of the tuple in the log. Child$_A$ i refers to the i'th child of the parent of the action/goal in tuple A.

Algorithm A. According to the first algorithm, an action is explained by the goal directly above the action, i.e. the parent goal of an action.

```
if (Explain(action/goal$_A$)) then
    (explanation = parent$_A$ = action/goal$_B$)
```

Following this explanation algorithm, action E in Figure 1 is explained by goal C.

Algorithm B. Explanation algorithm B explains actions by the goal two levels up in the goal tree, that is, the parent goal of the action's parent goal.

```
if (Explain(action/goal$_A$) and
    parent$_A$ = action/goal$_B$) then
        (explanation = parent$_B$ = action/goal$_C$)
```

In this case, action E is explained by goal A.

Algorithm C. Algorithm C explains actions by the belief(s) that enabled the execution of that action.

```
if (Explain(action/goal$_A$)) then
    (explanation = belief$_A$ 1...n)
```

According to explanation algorithm C, action E is explained by belief(s) e.

Algorithm D. Algorithm D checks if the relation of an action to its parent is of type *seq* and whether the action is not the last one of the sequence. If so, the action is explained by the next action or goal in the sequence. If not, the algorithm tries to explain the parent of the action in a similar way. Each time no relations of type *seq* are found or the action/goal is the last in the sequence, the algorithm searches one level higher in the tree for an explanation. When the top goal is reached without finding an explanation the top goal is provided as an explanation. Thus, an action is explained

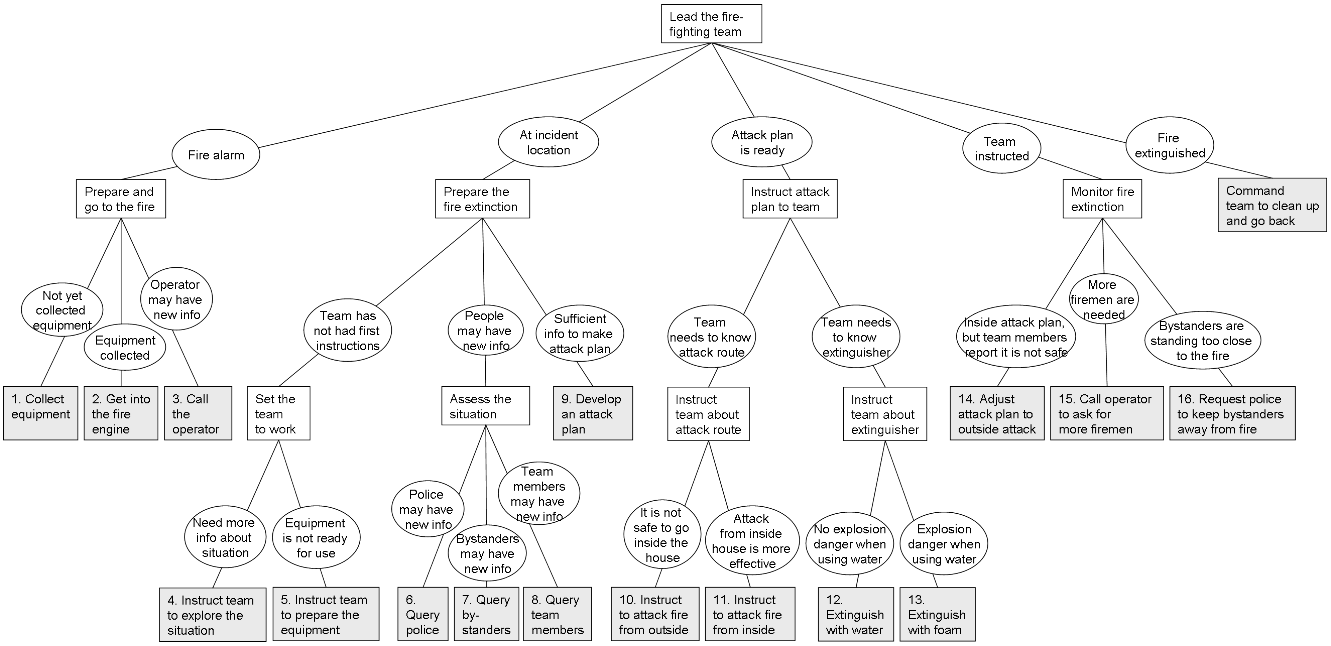
Lead the fire-fighting team
Fire alarm
At incident location
Attack plan is ready
Team instructed
Fire extinguished
Prepare and go to the fire
Prepare the fire extinction
Instruct attack plan to team
Monitor fire extinction
Command team to clean up and go back
Not yet collected equipment
Equipment collected
Operator may have new info
Team has not had first instructions
People may have new info
Sufficient info to make attack plan
Team needs to know attack route
Team needs to know extinguisher
Inside attack plan, but team members report it is not safe
More firemen are needed
Bystanders are standing too close to the fire
1. Collect equipment
2. Get into the fire engine
3. Call the operator
Set the team to work
Assess the situation
9. Develop an attack plan
Instruct team about attack route
Instruct team about extinguisher
14. Adjust attack plan to outside attack
15. Call operator to ask for more firemen
16. Request police to keep bystanders away from fire
Need more info about situation
Equipment is not ready for use
Police may have new info
Bystanders may have new info
Team members may have new info
It is not safe to go inside the house
Attack from inside house is more effective
No explosion danger when using water
Explosion danger when using water
4. Instruct team to explore the situation
5. Instruct team to prepare the equipment
6. Query police
7. Query by-standers
8. Query team members
10. Instruct to attack fire from outside
11. Instruct to attack fire from inside
12. Extinguish with water
13. Extinguish with foam

Figure 2. Goal hierarchy of a leading firefighter agent.

by the first action or goal that must follow the action (independent of the conditions in the environment), and that becomes achievable or executable because of the action.

```
if $(Explain(action/goal_A)$ and $action/goal_A = child_A\ i$
    and $relation_A = seq)$ then
        $(explanation = child_A\ i{+}1)$
else if $(Explain(action/goal_A)$ and $(action/goal_A =$
    $child_A\ n$ or not $relation_A = seq))$ then
        $(Explain(parent_A))$
else $(explanation = action/goal_A)$
```

Following algorithm D, if relation 2 is of type *seq*, action E is explained by action F. If relation 2 is not of type *seq*, but relation 1 is, action E is explained by goal D. If both relation 1 and 2 are not of type *seq*, action E is explained by goal A.

4 User evaluation of explanation algorithms

In this section we present a user study in which explanations generated by the four explanation algorithms are compared on their usefulness for learning. Our hypothesis is that preferred explanation depends on the relation between the action to explained and its parent.

4.1 Method

The domain of the study is fire-fighting training because fire-fighting is a complex, dynamic task requiring a lot of interaction. We expect that explainable agents are particularly useful in such domains.

We analyzed the task of a leading fire-fighter by reading protocols and consulting experts. Based on that, we constructed the goal hierarchy shown in Figure 2, and implemented the agent in 2APL. For the experiment, we used the actions in the numbered grey boxes in Figure 2. Action 4, 5, 6, 7 and 8 have a relation of type *all* to their parent goal, action 10, 11, 12, 13 have a relation of type *one*, action 14, 15 and 16 have a relation of type *if*, and action 1, 2, 3 and 9 have a relation of type *seq*. We applied the four explanation algorithms, A, B, C, and D, to the 16 actions to generate four explanations for each action. We composed a questionnaire in English containing the 16 actions with each four explanations. For each subject, the questions were placed in a (different) random order to correct for order effects. The order of the four explanations per action was also randomized. The question concerning action 1 was for example:

Why did the leading firefighter collect his equipment?
- To prepare and go to the fire (A)
- To lead the fire-fighting team (B)
- Because he had not yet collected his equipment (C)
- To get into the fire engine after that (D)

The study was performed with 20 subjects (12 male, 8 female). All had higher education, the average age was 31

		Explanation types			
Action	**Relation**	**A**	**B**	**C**	**D**
1.	seq	13	1	3	3
2.	seq	7	6	5	2
3.	seq	4	1	14	1
4.	all	1	2	13	4
5.	all	1	12	6	1
6.	all	10	0	9	1
7.	all	7	0	12	1
8.	all	6	2	10	2
9.	seq	10	3	5	2
10.	one	1	1	18	0
11.	one	1	1	18	0
12.	one	2	3	15	0
13.	one	2	1	17	0
14.	if	0	1	19	0
15.	if	0	0	20	0
16.	if	1	0	19	0
Total		**66**	**34**	**203**	**17**

Table 1. Frequencies of preferred explanation types per action.

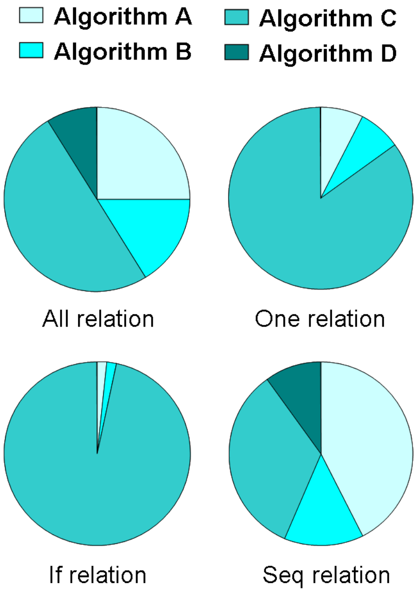

Figure 3. Percentages of preferred explanation type per action relation.

(range 24-58) and 17 were native Dutch speakers. The subjects were unfamiliar with the domain of fire-fighting and the task of a leading fire-fighter. This was done because the explanations are intended to be useful for learning a task. By choosing a non-familiar domain, the subjects could judge for themselves which explanation was most useful for this purpose.

The procedure of the experiment was as follows. Before the experiment, the subjects received instructions about the goal of the experiment and they were given a brief overview of the tasks of a leading firefighter. After the briefing, the subjects completed the paper questionnaire with 16 multiple choice questions. The subjects had to indicate for 16 actions which one of four explanations they considered most useful in learning the task of a leading fire-fighter. Usefulness was further explained as: 'which explanation helps you best in understanding the leading firefighter's motives for performing those actions?'.

4.2 Results

Table 1 shows for each of the fire-fighter's actions (1-16): the action's relation to its parent task (all, seq, if, one), and per explanation type (A, B, C and D), the number of subjects who preferred that explanation for the action. To remind, algorithm A explains an action by the parent goal of the action, algorithm B explains an action by the parent goal of the parent goal of the action, algorithm C explains an action by the enabling belief(s) of that action, and algorithm D explains an action by the first action or goal that must follow the action.

As the frequencies in Table 1 are dependent observations (the subjects make more than one observation), the statistical analysis of the results is done per action. We performed a Chi-square goodness-of-fit test to the results of every action, with an expected frequency of 5 per explanation algorithm. The tests on action 2 (χ^2 = 2.8, p = 0.42) and action 9 (χ^2 = 7.6, p = 0.055) indicate that the results are not significant, the test on action 8 (χ^2 = 8.8, p = 0.032) shows significance at a level of 5%, and the tests on all other actions are significant at a level of 1%.

We calculated kappa to express the agreement among subjects and found K = 0.23 (p = 0.15), indicating fair agreement. The low agreeability can partly be explained by the fact that algorithm C was overall often chosen, which gives a high P(E) in the calculation of kappa. Some statisticians suggest using free-marginal kappa when raters are not forced to assign a certain number of cases to each category. A free-marginal kappa calculation gives K = 0.44 and when action 2 and 9 (not significant) are omitted even K = 0.50, indicating moderate agreement.

The pie charts in Figure 3 summarize the data in Table 1. Each pie chart shows for one action relation (all, one, if, seq) the percentages of preferred explanation types. The charts clearly show that for actions with a *one* or an *if* relation to their parent, explanations generated by algorithm C are preferred, i.e. explanations containing an enabling belief. For actions with relations *and* and *seq*, also other explanation types are preferred. For all action relations holds that explanations generated by algorithm D were rarely chosen.

4.3 Discussion

First of all, our hypothesis that different actions yield different preferred explanation types is confirmed by the results. Table 1 clearly shows that different explanation types are preferred for different actions, and the observations are supported by statistical analysis on the results.

More specifically, we had the hypothesis that the preferred explanation type of an action depends on the relation between the action and its parent goal. The pie charts in Figure 3 indeed show different patterns of preferred explanation types for the different action relations (all, one, if, seq). For action relations *one* and *if*, subjects clearly preferred explanations generated by algorithm C, and the action relations *all* and *seq* yield different preferred explanation types. However, for actions with *all* or *seq* relations the results do not show one single preferred explanation type. Instead, the answers of the subjects are spreaded over different explanation types.

There are different explanations for the variation among preferred explanations within the actions with *seq* and *all* relations. For some of the actions there is no general agreement on preferred explanation type (explanations generated by algorithm A, B, C and D). This holds for action 2 and 9, which results are not significant, and to a lesser extent for action 6 and 8, where not more than 50% of subjects preferred the same explanation. Possible explanations for this disagreement among subjects are that none of the offered explanations are considered useful and the results do not give much information (this would require a whole other type of explanation), or that a combination of the provided explanations is considered useful. The low agreeability could also be due to methodological errors. For instance, for action 2, 'Get into the fire engine', the word 'fire engine' may have caused problems for mostly native Dutch speaking subjects, with English as their second language. And action 9, 'Develop an attack plan', could be misplaced in the goal hierarchy. Instead of being a subaction of the goal 'Prepare the fire extinction', it could be a subaction of the main goal, 'Lead the firefighting team'. In that case, other explanations would have been generated.

For other actions, it seems that preferred explanation type does not (only) depend on the action relation. The frequencies in Table 1 show that for most actions with relation *seq* and *all*, one explanation type is preferred by 60 to 70% of the subjects (action 1, 3, 4, 5 and 7). However, for other actions, the majority of the subjects prefers an explanation generated by algorithm A (action 1), algorithm B (action 5), and by algorithm C (3, 4, 7). Consequently, in addition to action relation, other factors are needed to account for preferred explanation type.

Let us consider the different explanation algorithms. Explanations generated by algorithm D are rarely preferred and not discussed further here. Of the other three algorithms, A and B generate explanations containing a goal to be achieved, and C generates explanations containing an enabling belief. The result of application of algorithm A and B strongly depends on the design of an agent's goal hierarchy. Actions can sometimes be placed at different levels in the goal hierarchy, as seen with action 9, and it is up to the designer how many subgoals are included in the hierarchy. Thus, the explanations generated by algorithm A and B are more similar to each other than explanations generated by algorithm C. When we collapse A and B into one category, of the four actions with a *seq* relation, C explanations are preferred for action 3, but for the others at least 65% of the subjects prefers an explanation with a goal to be achieved (either one or two levels above the action). This gives a stronger indication that, in general, goal-based explanations (A or B) are preferred for actions with a *seq* relation.

Though there are no clearly preferred explanation types for actions with relation *all* and *seq*, there is a clear difference between actions with relation *if* and *one* on the one hand, and *seq* and *all* on the other hand. A difference between the two groups is that *all* and *seq* actions have to be executed in order to achieve their parent goal, whereas the execution of *if* and *one* actions is not always necessary for the achievement of their parent goal. For *if* and *one* actions holds that their execution strongly depends on the conditions in the environment. For example, dependent on the risk of explosion, the fire is either extinguished with water (action 12) or with foam (action 13). And only if more firemen are needed, the leading firefighter has to call the operator (action 15). Conditions in the environment are reflected in the agent's beliefs and it is therefore not surprising that explanations containing a belief are preferred. Actions like getting into the fire engine (action 2) and instructing the team to prepare the equipment (action 5), on the other hand, always need to be performed when there is a fire. Actions with a relation *all* and *seq* are often part of a procedure or embody a rule. Thus, when the performance of an action strongly depends on the state of the environment, a belief-based explanation is preferred, and when that is not the case, other explanation types are also preferred.

Finally, we make some remarks on the scope of the experiment. As already mentioned, we did not consider explanations containing more than one mental concepts, e.g. a combination of algorithm A and C. The reason for this is that in an earlier experiment when we asked subjects to explain agent actions, they provided explanations which mostly contained one element. Furthermore, in this experiment we asked subjects to indicate the explanation they considered most useful, but we did not examine *how* useful the explanations are. It is possible that the preferred explanations were only considered least bad, but this is not probable as there is a lot of evidence that people prefer explanations in terms of beliefs and goals.

5 Conclusion and future work

In this paper we have presented a model for explainable agents involving BDI agent design and an explanation module with explanation algorithms. We have proposed four explanation algorithms generating different explanation types. In an empirical study, we examined user preferences about the explanation types generated by the different algorithms. We found that, in general, actions that are executed to follow a rule or procedure are sometimes preferred to be explained by goals and sometimes by beliefs, and actions whose execution depends on conditions in the environment are preferred to be explained by beliefs.

In future work, first some research is required to finetune when to use which (combination of) explanation algorithms, possibly taking user characteristics into account. Different users may prefer different explanation types, e.g. novices versus experts, or users that already received certain explanations versus users that did not yet receive them. Then, we plan to perform an experiment in which we compare student performance in two conditions: one group receiving virtual training with explanations about agent behavior, and the other group receiving training without explanations. By examining the effects of the explanations on learning, the explainable agent model is actually validated.

Acknowledgments

This research has been supported by the GATE project, funded by the Netherlands Organization for Scientific Research (NWO) and the Netherlands ICT Research and Innovation Authority (ICT Regie).

References

[1] M. Core, H. Lane, M. Van Lent, D. Gomboc, S. Solomon, and M. Rosenberg. Building explainable artificial intelligence systems. In *AAAI*, 2006.

[2] M. Core, T. Traum, H. Lane, W. Swartout, J. Gratch, and M. Van Lent. Teaching negotiation skills through practice and reflection with virtual humans. *Simulation*, 82(11):685–701, 2006.

[3] M. Dastani. 2APL: a practical agent programming language. *Autonomous Agents and Multi-agent Systems*, 16(3):214–248, 2008.

[4] D. Dennett. *The Intentional Stance*. MIT Press, 1987.

[5] F. Dignum, J. Westra, W. Van Doesburg, and M. Harbers. Games and agents: Designing intelligent gameplay. *International Journal of Computer Games Technology*, 2009.

[6] R. Flin and K. Arbuthnot, editors. *Incident command: Tales from the hot seat*. Ashgate Publising, 2002.

[7] D. Gomboc, S. Solomon, M. G. Core, H. C. Lane, and M. van Lent. Design recommendations to support automated explanation and tutoring. In *Proc. of BRIMS 2005*, Universal City, CA., 2005.

[8] M. Harbers, K. Van den Bosch, and J.-J. Meyer. A methodology for developing self-explaining agents for virtual training. In M. Dastani, A. Fallah Seghrouchni, J. Leite, and P. Torroni, editors, *Proc. of Lads 2009*, Turin, Italy, 2009.

[9] M. Harbers, K. Van den Bosch, and J.-J. Meyer. A study into preferred explanations of virtual agent behavior. In Z. Ruttkay, M. Kipp, A. Nijholt, and H. Vilhjlmsson, editors, *Proc. of IVA 2009*, pages 132–145, Amsterdam, Netherlands, 2009. Springer Berlin/Heidelberg.

[10] S. Haynes, M. Cohen, and F. Ritter. Designs for explaining intelligent agents. *International Journal of Human-Computer Studies*, 67:90–110, 2009.

[11] F. Keil. Explanation and understanding. *Annual Reviews Psychology*, 57:227–254, 2006.

[12] B. Keysar, S. Lin, and D. Barr. Limits on theory of mind use in adults. *Cognition*, 89:25–41, 2003.

[13] W. Lewis Johnson. Agents that learn to explain themselves. In *Proc. of the 12th Nat. Conf. on Artificial Intelligence*, pages 1257–1263, 1994.

[14] B. Malle. How people explain behavior: A new theoretical framework. *Personality and Social Psychology Review*, 3(1):23–48, 1999.

[15] S. Nickerson. How we know -and sometimes misjudge-what others know: Imputing one's own knowledge to others. *Psychological Bulletin*, 125(6):737–759, 1999.

[16] E. Norling. Capturing the quake player: using a bdi agent to model human behaviour. In J. Rosenschein, T. Sandholm, M. Wooldridge, and M. Yokoo, editors, *Proceedings of AAMAS 2003*, pages 1080–1081, 2003.

[17] S. Oh, J. Gratch, and W. Woo. Explanatory style for socially interactive agents. In *Proceedings of the International Conference on Affective Computing and Intelligent Interaction*, 2007.

[18] S. Sardina, L. De Silva, and L. Padgham. Hierarchical planning in BDI agent programming languages: A formal approach. In *Proceedings of AAMAS 2006*. ACM Press, 2006.

[19] J. Schraagen, S. Chipman, and V. Shalin, editors. *Cognitive Task Analysis*. Lawrence Erlbaum Associates, Mahway, New Jersey, 2000.

[20] W. Swartout and J. Moore. *Second-Generation Expert Systems*, chapter Explanation in Second-Generation Expert Systems, pages 543–585. Springer-Verlag, New York, 1993.

[21] W. A. Van Doesburg, A. Heuvelink, and E. L. Van den Broek. Tacop: A cognitive agent for a naval training simulation environment. In S. T. M. Pechoucek, D. Steiner, editor, *Proceedings of the Industry Track of AAMAS 2005*, pages 34–41, 2005.

[22] M. Van Lent, W. Fisher, and M. Mancuso. An explainable artificial intelligence system for small-unit tactical behavior. In *Proc. of IAAA 2004*, Menlo Park, CA, 2004. AAAI Press.