Do you get it? User-evaluated explainable BDI agents

Joost Broekens[1], Maaike Harbers[2], Koen Hindriks[1],
Karel van den Bosch[3], Catholijn Jonker[1] and John-Jules Meyer[2]

Delft University of Technology[1], Utrecht University[2], TNO Institute of Defence, Security and Safety[3], The Netherlands

Abstract. In this paper we focus on explaining to humans the behavior of autonomous agents, i.e., explainable agents. Explainable agents are useful for many reasons including scenario-based training (e.g. disaster training), tutor and pedagogical systems, agent development and debugging, gaming, and interactive storytelling. As the aim is to generate for humans plausible and insightful explanations, user evaluation of different explanations is essential. In this paper we test the hypothesis that different explanation types are needed to explain different types of actions. We present three different, generically applicable, algorithms that automatically generate different types of explanations for actions of BDI-based agents. Quantitative analysis of a user experiment (n=30), in which users rated the usefulness and naturalness of each explanation type for different agent actions, supports our hypothesis. In addition, we present feedback from the users about how they would explain the actions themselves. Finally, we hypothesize guidelines relevant for the development of explainable BDI agents[1].

1 INTRODUCTION

Explaining to users how AI systems come to their conclusions is an area of research with a history in expert systems and planning (see e.g., [1][2]). In this paper we focus on explaining to humans the behavior of autonomous agents. Explainable agents that use natural language for their explanations are useful in many domains. In scenario-based training (e.g. disaster or military training) the agents in the training should be able to explain the rationale for their actions so that students can understand why the training unfolds as it does [3]. In tutor and pedagogical systems, natural dialog between the user and system has been shown to increase the training effect of such systems [4]. Debugging tools for BDI agent programs might benefit from a natural way of interaction involving asking why agents perform certain actions instead of looking at execution traces and internal mental states [5]. In gaming and interactive storytelling [6][7], having automatic mechanisms to generate explanations of agent actions (the "story") could enhance the flexibility and appeal of the storyline.

Humans understand and explain (vocalize) their own and others' behavior in terms of *folk psychology*, that is, in terms of its underlying mental states like beliefs, desires and intentions [8]. To automatically generate similar explanations of agent behavior,

[1] Acknowledgements. 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), as well as STW (NWO) VICI-project 08075.

it is convenient to have explicit representations of agent beliefs, goals and plans. This can be accomplished by using a BDI-based (belief desire intention) agent programming approach. Behavior in BDI agents is motivated by goals (desires), and selected based on whether or not an agent believes a particular behavior will satisfy a goal or subgoal. Behavior is then committed to (an action or sequence of actions is planned) transforming it into an intention. The outcome of a BDI agent's reasoning, i.e., its actions, can then be explained by the goals and beliefs that were responsible for it. Our approach to generating explanations is based on using the already available (relations between) mental constructs in the agent program that generates the agent behavior. It was found, that humans usually provide action explanations that only contain one or two mental concepts [9]. Thus, in particular when agents are complex, providing as explanation the complete trace of beliefs and goals underlying an action is undesirable. Instead, an explanation based on a selection of beliefs and goals underlying the action is needed.

Our hypothesis is that different actions require different *types of explanations*, i.e., an interaction effect exists between type of explanation and action on the perceived quality of an explanation. We present a study in which users evaluate three algorithms that each automatically generate a different type of explanation for 10 different agent actions. For each action and explanation type subjects rated usefulness and naturalness.

In Section 3 we distinguish different action types, and we present three generically applicable algorithms for automatically generating different explanation types for BDI agent actions. In Section 5 we present a quantitative analysis of a user evaluation experiment (n=30) to assess the usefulness and naturalness of the generated explanation types for different agent actions. We also present feedback from the users about how they would explain the actions themselves. Finally, in the discussion we hypothesize guidelines for the kind of information that should be modeled in the BDI agent if meaningful explanations are to be generated. First we discuss related work in the next section.

2 RELATED WORK

In the introduction we have mentioned several application domains of explainable agents. Most of the related work is in virtual training systems. We now briefly review explainable agent approaches in this domain.

Debrief is the first system that explains agent behavior [10]. Debrief is implemented as part of a fighter pilot simulation and allows trainees to ask an explanation 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 which factors must have been responsible for the agent's decisions.

Another account of explainable agents is the XAI (eXplainable Artificial Intelligence) explanation component [11]. The XAI system has been incorporated into 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, e.g. its location or health.

A second version of the XAI system was developed to overcome the shortcomings of the first. It is claimed that the new XAI system supports domain independency, mod-

ularity and the ability to explain the motivations behind agents' actions. The system is described in [12] and [3], where it is applied to a tactical military simulator, and a virtual trainer for soft skills such as leadership, teamwork, negotiation and cultural awareness, respectively. For the generation of explanations, the system depends on information that is made available by the simulation.

Both Debrief and the first XAI system lack the ability to provide explanations involving the motivations behind an agent's actions. The XAI system only provides information about an agent's physical state, and not about its mental. Debrief does provide explanations in terms of an agent's beliefs, but never gives explanations including its underlying goals and intentions. The second XAI system can provide explanations in terms of an agent's goals, but only if those are represented as such in the simulation, which is often not the case [13]. If the agent's goals are not represented in the simulation, a hand-built XAI representation of the behaviors has to be made. Consequently, changes in the agent specification must also be reflected in the explanation component.

3 EXPLAINABLE AGENT MODEL

In this section we describe an explainable agent model that can provide different types of explanations about agent behavior. The basic principle of the model is that the mental concepts responsible for an agent's action are also used to explain that action. Because not all mental concepts underlying an action are needed to explain that action, we also present three different explanation algorithms that select a mental concept that is most appropriate to generate an explanation.

As mentioned in the introduction, BDI-based agent programming languages allow for the explicit representation of an agent's mental state, and actions are the result of a deliberation process on the agent's mental concepts. In our study, we have used the BDI language GOAL [14]. A GOAL agent program consists of six different sections, including the agent's knowledge, beliefs, goals, action rules, action specifications and percept rules. Together, the knowledge, beliefs and goals of an agent make up its mental state. Although GOAL distinguishes itself from other BDI-based languages in the exact way agents are specified and executed, we would like to stress that the explanation approach presented in this paper can also be applied to other BDI-based agents.

To explain agent behavior by the underlying mental concepts, we need two things. First, the agent's past goals and beliefs must be accessible when the explanation is constructed. Second, when there is a request to explain an action, the proper goals and beliefs explaining that action must be selected. We have implemented an explanation module that satisfies these two requirements.

3.1 Tree-based behavior log

The explanation module includes a mechanism to construct a behavior "log", to which an agent's goals and beliefs are updated. The explanation module can be connected to any GOAL agent, and during run-time of the agent, the explanation module examines and logs the execution of the agent program.

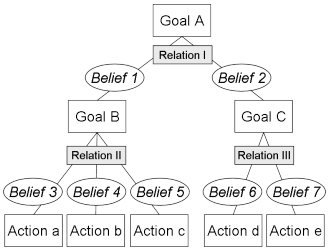

Fig. 1. Example behavior "log" (goal tree) of a BDI agent.

The behavior "log" in the explanation module is a tree structure that is constructed while the agent reasons and performs actions based on its agent program (so formally it is not a log, as in a timed list of actions). It is made such that it automatically construct a goal tree based on the actual behavior of the agent, see e.g. Figure 2 representing a particular execution of the agent program as used in the experiment (please also see our notes at the end of Section 3.2). The algorithm (in text) is as follows: The agent's initial goal becomes the top node of the tree (Goal A in Figure 1). If the program decides to adopt a goal in order to achieve another goal, this is represented as a subgoal (Goal B and C). The adoption conditions of a goal, i.e., beliefs that determine whether the agent program should adopt a subgoal, are represented along the branches of the tree (Belief 1-7). The agent's actions form the leaves of the tree (Action a-e). This algorithm automatically constructs a tree structure that is different depending on the actual behavior and choices of the agent.

In addition to this tree, one has to supply the behavior log with goal-relation information. Currently we add this manually, but this information could be explicitly represented, or extracted from the agent program. Goals can have three different relations to their subgoals (relation I-III): *all*, *one* and *seq*. A goal with an *all* relation to its subgoals/actions means that all subgoals/actions must be fulfilled in arbitrary order to achieve the goal, relation *one* means that exactly one of the subgoals/actions must be fulfilled to achieve the goal, and relation *seq* (from sequential) means that all subgoals/actions must be fulfilled in a particular fixed order. Based on these relations, we distinguish the following three types of actions.

- *All* action: relation to parent goal is of type all
- *One* action: relation to parent goal is of type one
- *Seq* action: relation to parent goal is of type seq

To summarize, we distinguish three different action types, where the action type depends on the relation to an action's parent goal and its siblings. In the next section we present three explanation algorithms that generate different types of explanations.

3.2 Explanation algorithms

When a user requests an action explanation, an explanation algorithm is applied to the behavior log. Based on the log, the algorithm determines the goals and beliefs that are reasons for the action. Then, it selects beliefs and goals relevant for the explanation. We propose three algorithms for constructing three different types of explanation.

Algorithm I. The first explanation algorithm explains actions by the goal that motivated the selection of the action. It generates a sentence that looks like "Because I want to <goal>". We expect that this algorithm delivers useful explanations for actions of the type *all*, meaning that the action and all its sibling actions have to be executed in order to achieve their parent goal. For example, if relation II in Figure 1 would be of the type *all*, we expect that action b is best explained by goal B.

Algorithm II. The second algorithm explains an action by its enabling condition, i.e. the belief because of which it was executed. It generates a sentence that looks like "Because I believe that <belief condition>". We expect that these explanations are useful in particular for actions of the type *one*, meaning that only one of a goal's children actions needs to be executed to achieve it. In Figure 1 for example, if relation III would be of the type *one*, we expect that belief 6 provides the explanation for action d. Namely, belief 6 determined that action d was chosen to achieve goal C and not action e.

Algorithm III. In the third algorithm, an action is explained by the first action or task that must follow after the action. Thus, if an action is part of a sequence of actions that must be executed in a particular order to achieve a goal, the action can be explained by the next action in the sequence. It generates a sentence that looks like "Because I want to <next goal>". We expect that this algorithm will deliver most useful explanations for actions of the type *seq*. For instance, if relation II is of the type *seq*, action b is explained by action c according to this algorithm. In other words, action b enables the execution of action c. If an action is not part of such a sequence, the algorithm considers the parent goal of the action, and checks whether this goal is part of a sequence of goals. In Figure 1, if relation II is not of the type *seq*, relation I is considered and if that is a *seq* relation, goal C is given as the explanation for action b. If the top goal is reached without finding a relation of the type *seq*, the top goal is provided as an explanation.

Note that the execution of GOAL agents that are designed according to a hierarchical goal model will result into a goal tree, i.e. there is one main goal and each goal has a limited number of subgoals or actions. As the explanation module automatically constructs a goal-condition-subgoal structure based on the execution trace of the agent, other agent programs may result into less regular tree-shaped graphs, e.g. one main goal with many subgoals, several separated trees when multiple independent initial goals are present, or several partly connected trees when multiple dependent initial goals are present. In principle, the explanation algorithms can be applied to all kinds of goal graphs to generate explanations, but we expect that the explanation algorithms will in general deliver more useful explanations when applied to a proper tree. The assumption of a hierarchical goal model is plausible, as it is based on existing knowledge elicitation methods. Namely, hierarchical task analysis (HTA), which is a well-accepted cognitive task analysis technique [15].

Also note that explanations could be asked for during runtime, as the goal tree is build up continuously. Although in this paper we assume the agent has executed its complete program, as long as the tree contains enough information for the explanation algorithm to generate an explanation, it does not need to be complete.

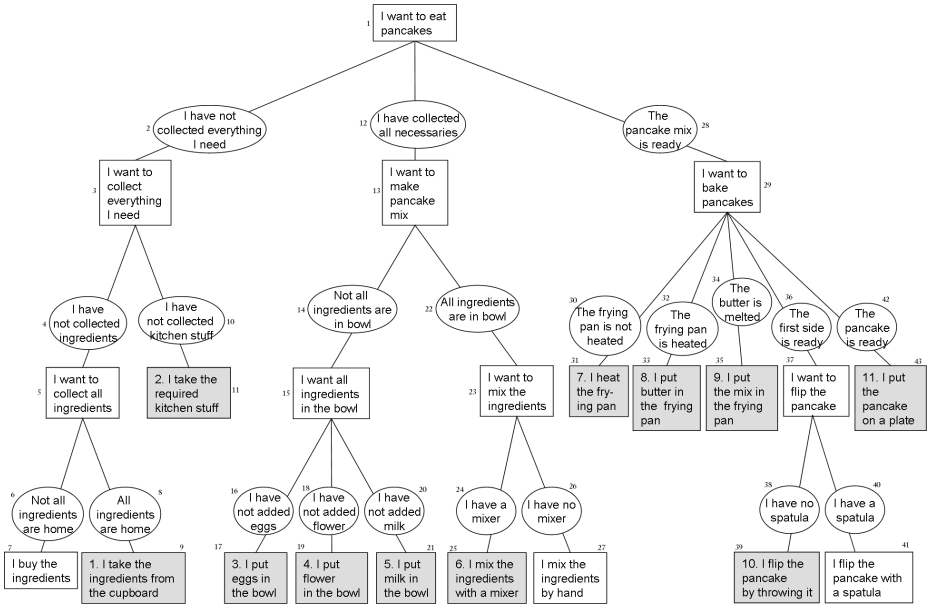

Fig. 2. Cooking agent behavior log. Grey boxes denote the 11 actions used in the experiment.

4 EXPERIMENTAL SETUP

To evaluate how users perceive the different explanation types for different actions, we have to test these in an application domain. We have chosen for a cooking agent that bakes pancakes and explains its actions. The reason for choosing a domain like this is that for average users to evaluate whether an explanation is useful and natural, the user must be familiar with the domain. He/she has to judge the explanation. This excludes more sophisticated domains such as disaster or negotiation training, as users are typically less familiar with these. Picking a domain limits the generalizability of our results, and we will come back to this issue in the discussion.

The cooking agent (Figure 2) was programmed in GOAL, and executed. To evaluate the effect of the different explanation types for the three action types, the agent program was constructed such that it included actions of all types. Action 2, 3, 4 and 5 are of type *all* (actions that all need to be executed), action 1, 6 and 10 are of type *one* (mutually exclusive actions), and action 7, 8, 9 and 11 are of type *seq* (actions that all need to be executed in a particular order). For all three explanation types, a list of explanations for all actions was generated. Post analysis excluded action 11 from the statistical result analysis as this action was misplaced in the tree (see Results section).

To investigate our hypothesis, we followed a between subject 10x3 design (10 actions, 3 algorithms) with dependent variables usefulness, naturalness. Subjects were randomly assigned to the different conditions with exactly 10 subjects per condition (n=30, 12 female, age(avg=32, stdev=9), cooking skills (5-point Likert scale, avg 3.6), average education level between Bachelor and Master, subjects were a balanced mix of family, friends, colleagues and students of the first two authors). All subjects scored all actions for a particular condition, resulting in 10 measurements per action per con-

dition. The first two authors each administered 15 tests, no effect of experimenter bias was found during analysis of the data.

The procedure for gathering feedback from the subjects was organized as follows. Subjects were told to read the instructions (stating that the study was about developing smart agents for virtual training purposes), after which they received the first feedback form. On this form subjects wrote down their own explanations for the 11 actions listed on the form (see also the gray boxes in Figure 2), as if they were the cook explaining how to bake pancakes to a student. This feedback was aimed at extracting the "ideal" explanations as perceived by the user, and to help subjects get into the right context. We do not evaluate this qualitative data in this paper. When finished, subjects received the second form. This form asked for 5-point likert feedback on the naturalness of each action's automatically generated explanation (1=not natural, 5=very natural). Subjects took the role of observer when judging the naturalness of the explanation. Naturalness was explained as follows: "With a natural explanation we mean an explanation that sounds normal and is understandable, an explanation that you or other people could give". When finished, a similar form was presented for 5-point likert feedback about the usefulness of the explanations. Subjects were asked to imagine they were the student learning to cook while judging the usefulness. Useful was explained as follows: "Indicate how useful the explanations would be for you in learning how to make pancakes". Finally, subjects were presented with the goal tree (the graphical representation of the behavior log as shown in 2). We asked users to indicate all elements in the tree they deemed useful for giving an explanation of each of the 11 actions, by putting the action number next to the element. Subjects were asked to imagine they were the cook while numbering elements. This feedback was aimed at extracting information about what could be a good and feasible version of an explanation algorithm, given our way of automatically generating tree-based behavior logs.

5 RESULTS

To test our main hypothesis, i.e., different actions require different types of explanations, we performed a 10x3 2-way MANOVA with explanation type (3 conditions) and action (10 conditions) as independent variables, and usefulness and naturalness as dependent variables. The MANOVA test is used to identify if signifiant differences in means of dependent variables are introduced by variation in independent (experimental) variables. Values of independent variables define groups, in our case 3x10=30 groups. Analysis showed a main effect of algorithm type ($F(4, 538) = 3.973, p < 0.01$), a main effect of action ($F(18, 538) = 1.917, p < 0.05$), and an interaction effect between action and algorithm ($F(36, 538) = 2.638, p < 0.001$). Post hoc testing (Tukey) for the influence of action alone on naturalness and usefulness revealed no significant differences between the actions on both measures. This indicates that the actions are equal with respect to explainability, meaning that no action is easier to explain than another. The same post hoc testing for the influence of algorithm type revealed only a significant effect on the perceived usefulness. Algorithm I (parent goal as explanation) performed significantly better ($p < 0.01$) than the other two algorithms ($Mean(I) = 3.1, Mean(II) = 2.5, Mean(III) = 2.5$). This indicates that there

is a significant influence of explanation type on the perceived usefulness of the explanation, and that explaining an action with its parent goal (Algorithm I) is the best default method. However, the interaction effect indicating that different actions need different explanations (supporting our main hypothesis), is more important, as we will see next.

In Figure 3 an overview is given of the average naturalness and usefulness of the actions per algorithm type. In Figure 4 an overview is given of the number of times subjects indicated a particular element in the tree-based user feedback.

As can be seen, actions 1, 2, 6 and 9 score high on both measures when the parent goal is given as explanation (Algorithm I), while actions actions 3, 4, and 5 score high on both measures when the next action or goal is given as explanation ("I want to mix the ingredients", Algorithm III), and actions 7 and 10 score high when the enabling condition (belief) is given as explanation (algorithm II). Action 8 does not score well on either of the algorithms. Action 11 is explained well by Algorithm III (next goal/action), but this is a side effect of two factors. First, action 11 was misplaced, it should have been under "I want to eat pancakes", as also indicated by the tree-based user feedback. Second, Algorithm III defaults to the top level goal when no next steps are available in the sequence, which in our case happened to be the most logical option for explanation. We exclude action 11 from our analysis.

Actions 2, 3, 4 and 5 are actions of the type *all*; they are all needed in arbitrary order to achieve the parent goal. For 3, 4 and 5, the parent goal is not very descriptive, when the action has already been read (I put X in the bowl - because I want to put all ingredients in the bowl). As can be seen in Figure 4 subjects included in their own choice of elements the goal numbered 13 ("I want to make pancake mix"), indicating that subjects indeed need a more descriptive goal. Action 2 is well explained by its parent goal, as indicated by the naturalness and usefulness feedback as well as the tree-based feedback.

Actions 1, 6 and 10 are actions of the type *one*. Action 1 and 6 score high on using the parent goal as explanation, but in addition to that they seem to require extra information for an adequate explanation. In Figure 3 we can see that for action 1 and 6 subjects use the goal two levels up in the hierarchy. Action 10 is well explained by Algorithm II (enabling condition). This is reflected in the tree-based feedback, as for action 6 and 10 subjects use the enabling conditions for the action and for the parent goal. Action 6 thus has a rather complex explanation structure using two goals and two conditions.

As indicated by the tree-based feedback, enabling conditions in combination with the parent goal are also used for action 7, 8, and 9; all three actions are actions in a sequence, type *seq*. However, action 8 and 9 use only the enabling condition for the action itself, while action 7 uses both the enabling condition for the action itself as well as the enabling condition for the action's parent goal. We will interpret these results in more detail in the discussion.

Finally, we have conducted correlations between the subject demographics and usefulness and naturalness. We found four significant correlations. Two of the correlations were positive: the one between usefulness and naturalness ($p < 0.001, r = 0.491$), and the one between cooking skill and usefulness ($p < 0.001, r = 0.145$). The first correlation is as expected: natural explanations are more useful and vice versa. The second is somewhat counterintuitive: more experienced cooks judge the explanations slightly

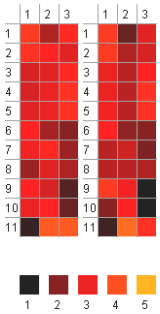

Fig. 3. Average naturalness (left) and usefulness (right) of actions (1-11) per condition (1-3).

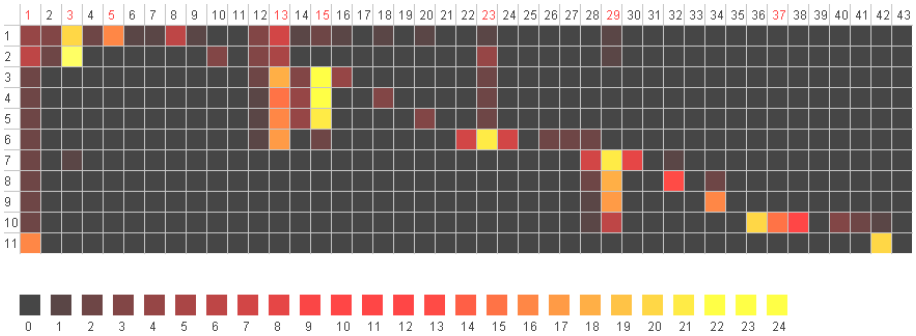

Fig. 4. Distribution of tree elements used to generate explanations for different actions (1-11) as given by the subjects. Elements number from 1 to 43 and refer to numbers in Figure 2.

more useful. This could be due to the fact that a better cook is better able to understand the explanation in the first place, but as the correlation coefficient is rather small, we do not pay further attention to this in this paper. Furthermore, we found two negative correlations: between action number and naturalness ($p < 0.001, r - 0.200$), and between action number and usefulness ($p < 0.01, r = -0.178$). As actions were always scored from top to bottom, and this corresponds to the action number, this might indicate two different things: for the later actions it is more difficult to automatically generate explanations, or, subjects got tired of scoring explanations. This issue needs future experiments.

6 DISCUSSION

We first discuss the results in more detail. Then we summarize the discussion by hypothesizing guidelines for the development of explainable BDI agents that generate explanations based on their behavior and mental processes. We end the discussion with several limitations of our study, such as the choice of domain and the choice of particular actions, subgoals and the linkage between them in the goal tree.

Our results indicate two things. First, the results support our main hypothesis: different actions need different explanation types, as indicated by the 2-way ANOVA showing

significant interaction between action and type of explanation. Second, our expectations on how action types and explanation algorithms are related are too simplistic. We expected that *all* actions (AND relation with siblings) would be explained best by the action's parent goal, that *seq* actions (AND and sequence relation with siblings) would be explained best by the next action/goal in the sequence, and *one* actions (XOR relation with siblings) would be best explained by their enabling condition. Looking at the tree-based feedback, most of the actions seem to need at least one additional element for explanation, in addition to their parent goal. The kind of additional information seems to depend on the action's role in the process and the action's type (seq, all, or one).

First consider the actions of type *all*: action 2, 3, 4 and 5. Of all actions, only action 2 is explained well by only one element, its parent goal. Action 3, 4 and 5 are well explained by the next action in the sequence (Figure 3), but when subjects produce their own tree-based feedback (Figure 4), they choose for a combination of the parent goal and the parent's parent goal. We currently can not explain this inconsistency, but it does indicate that neither the enabling condition nor the parent goal are descriptive enough in this particular case.

Now consider actions 1, 6 and 10 which are of type *one*. The way this type of action is modeled in the tree is such that the parent goal presents a choice, while the enabling condition of the action's parent explains why the choice has to be made. For this action type, the parent goal is not descriptive enough to provide a satisfying explanation. Instead, both the enabling condition of the action and the enabling condition of the parent goal are needed (Figure 4).

Finally, consider the actions 7, 8, 9, and 10 which are part of the same sequence (note that 7, 8 and 9 are of type *seq*, but 10 is of type *one*). According to the tree-based feedback (Figure 4), these actions should be explained by their parent goal and their enabling condition, contrary to our expectation that such actions would need the next action/goal in the sequence. In addition, action 7 and 10 also need the enabling condition of their parent's goal in their explanations. A possible explanation for this difference is that action 8 and 9 are in the middle of a sequence. Their parent goal explains what is to be done, and the enabling condition explains where we are in the process. Action 10 does need its parent goal and its enabling condition because it is an action of type *one*. The enabling condition of its parent goal needs to be given because it is also, though implicitly, part of the sequence involving action 7 to 10. Action 7 can be explained in the same way. It is the first action of a next phase in the process (baking). Phase in this case is defined as either preparation for baking, or baking. The parent goal of action 7 is about that next phase, but it does not explain why we ended up in this phase. This is what the parent goals' enabling condition is about, hence, action 7 needs again two enabling conditions (it's own and that of its parent goal).

According to studies in psychology, humans explain intentional behavior using reasons while they explain unintentional behavior using causes [16]. Furthermore, when behavior was made possible by opportunity, skill or by removal of an obstacle, people tend to use a description of enabling factors for explaining the behavior (e.g., why does a person start driving when waiting for a traffic light? Because the light turns green). Obviously, all of our agent behavior is intentional, but for a human, actions of the type *one* (OR, XOR) could well be considered driven by opportunity in our case (having

ingredients at home or not, having a mixer or not). It is therefore in line with [16] that these actions need their enabling condition for explanation. Also the actions in sequence 7-10 need an enabling condition. When performing an action sequence, the whole sequence is intentional, but the actions within the sequence are controlled by external factors or the logic of the process. These can thus be considered non-intentional, and it is therefore again in line with [16] that also these actions need their enabling condition.

6.1 Guidelines

We now sum up this discussion and present several guidelines relevant for the development of explainable BDI agents. The guidelines are hypotheses, and should be tested in further research. First, as the parent goal of an action seems essential in its explanation, explanation methods should first attempt to use this. This also suggests that explainable-agent programmers should make these parent goals as meaningful as possible in light of an explanation. Second, actions that start a new phase in a process need additional explanation in the form of the enabling conditions for the action and the parent goal. Third, care should be taken when explaining XOR choices (*one* action type) using a common parent goal as ”abstract action”, because such a parent goal is often non descriptive. This means that either the explanation method must take this into account (e.g., by using agent-program meta information), or such choices should be modeled differently. Fourth, sequenced actions need to be ”chained” using their enabling condition, so that the user can position the action in the sequence.

6.2 Limitations and Future work

We have chosen a domain that is well-known to the subjects because we wanted all subjects to be able to judge the naturalness and usefulness of the explanations. Our current aim was not to investigate if these explanations actually result in, e.g., a better training session. In future work we plan to perform similar experiments with subjects that are not familiar with the domain (e.g., a disaster training) to test whether generated explanations increase the understanding of these subjects.

Furthermore, the particular agent program used to represent beliefs, goals and resulting action selection, produces a particular hierarchical goal structure. Although we expect similar structures are ubiquitous in programs, more research is needed on relaxing these structural constraints.

A similar issue is the particular instantiation of our BDI program. Our results might be limited to our specific goal tree. However, we have taken care to construct the goal hierarchy such that it contains duplication of action types at different places. Therefore, we feel that similar results for action explanation at two different places indicates that the result is generic for that action type.

7 CONCLUSION

In this paper we have presented a study involving user evaluations of explanations about agent behavior. We distinguished three action types and three algorithms automatically

generating different explanation types. We investigated which explanation types are preferred for which actions. Our hypothesis that different actions require different types of explanations, as generated by different explanation algorithms, was supported by the results. We found that an action should always be explained by its parent goal, and depending on the action type, particular additional information is needed. We have abstracted this and other findings into four guidelines relevant for the development of explainable BDI agents and explanation algorithms.

References

1. Cortellessa, G., Cesta, A.: Evaluating mixed-initiative systems: An experimental approach. In: ICAPS'06. (2006) 172–181
2. Gilbert, N.: Explanation and dialogue. The Knowledge Engineering Review **4**(03) (1989) 235–247 10.1017/S026988890000504X.
3. Core, M., Traum, T., Lane, H., Swartout, W., Gratch, J., Van Lent, M.: Teaching negotiation skills through practice and reflection with virtual humans. Simulation **82**(11) (2006) 685–701
4. Graesser, A.C., Chipman, P., Haynes, B.C., Olney, A.: Autotutor: an intelligent tutoring system with mixed-initiative dialogue. Education, IEEE Transactions on **48**(4) (2005) 612–618
5. Broekens, J., DeGroot, D.: Formalizing cognitive appraisal: from theory to computation. In Trapple, R., ed.: Cybernetics and Systems 2006, Vienna, Austrian Society for Cybernetics Studies (2006) 595–600
6. Cavazza, M., Charles, F., Mead, S.J.: Character-based interactive storytelling. IEEE Intelligent Systems **17**(4) (2002.) 17–24
7. Theune, M., Faas, S., Heylen, D.K.J., Nijholt, A.: The virtual storyteller: Story creation by intelligent agents. In: TIDSE 2003: Technologies for Interactive Digital Storytelling and Entertainment, Darmstadt, Fraunhofer IRB Verlag (2003) 204–215
8. Keil, F.: Explanation and understanding. Annual Reviews Psychology **57** (2006) 227–254
9. Harbers, M., Van den Bosch, K., Meyer, J.: A study into preferred explanations of virtual agent behavior. In Ruttkay, Z., Kipp, M., Nijholt, A., Vilhjlmsson, H., eds.: Proc. of IVA 2009, Amsterdam, Netherlands, Springer Berlin/Heidelberg (2009) 132–145
10. Johnson, W.: Agents that learn to explain themselves. In: Proc. of the 12th Nat. Conf. on Artificial Intelligence. (1994) 1257–1263
11. Van Lent, M., Fisher, W., Mancuso, M.: An explainable artificial intelligence system for small-unit tactical behavior. In: Proc. of IAAA 2004, Menlo Park, CA, AAAI Press (2004)
12. Gomboc, D., Solomon, S., Core, M.G., Lane, H.C., van Lent, M.: Design recommendations to support automated explanation and tutoring. In: Proc. of BRIMS 2005, Universal City, CA. (2005)
13. Core, M., Lane, H., Van Lent, M., Gomboc, D., Solomon, S., Rosenberg, M.: Building explainable artificial intelligence systems. In: AAAI. (2006)
14. Hindriks, K.: Programming Rational Agents in GOAL. In: Multi-Agent Programming: Languages, Tools and Applications. Springer (2009) 119–157
15. Schraagen, J., Chipman, S., Shalin, V., eds.: Cognitive Task Analysis. Lawrence Erlbaum Associates, Mahway, New Jersey (2000)
16. Malle, B.: How people explain behavior: A new theoretical framework. Personality and Social Psychology Review **3**(1) (1999) 23–48