A Cognitive Model for the Generation and Explanation of Behaviour in Virtual Training Systems

Maaike Harbers[1,2], Karel van den Bosch[2], Frank Dignum[1], and John-Jules Meyer[1]

[1] Institute of Information and Computing Sciences, Utrecht University, P.O.Box 80.089, 3508 TB Utrecht, The Netherlands
maaike,dignum,jj@cs.uu.nl
[2] TNO Defence, Security & Safety, Kampweg 5, 3796 DE Soesterberg, The Netherlands
maaike.harbers,karel.vandenbosch@tno.nl

Abstract. Instructors play a major role in many of the current virtual training systems. Consequently, either many instructors per trainee are needed, which is expensive, or single instructors perform highly demanding tasks, which might lead to suboptimal training. To solve this problem, this paper proposes a cognitive model that not only generates the behaviour of virtual characters, but also produces explanations about it. Agents based on the model reason with BDI concepts, which enables the generation of explanations that involve the underlying reasons for an agent's actions in 'human' terms such as beliefs, intentions, and goals.

1 Introduction

Virtual systems have become common instruments for training in organizations such as the army, navy and fire brigade. The following paragraph describes a possible training scenario of such systems.

To practise his incident management skills, fire-fighter F receives training in a virtual environment. He and his virtual colleague G receive a call from the dispatch centre informing that there is a fire in a house, and as far as known, there are no people in the house. While driving to the incident, they receive information about the exact location, scale of the incident, wind direction, etc. At the incident, F and G start unrolling and connecting the fire hoses, but suddenly, G sees a victim behind a window and runs towards the house. F did not see the victim and assumes that G is going to extinguish the fire. However, F sees that G is not carrying a fire hose, and wonders whether G forgot his hose, or ran to the house for another reason. What should F do?

Virtual training systems provide an environment which represents those parts of the real world that are relevant to the execution of the trainee's task. An interface

allows trainees to interact with the environment. In typical virtual training, a trainee has to accomplish a given mission, while a scenario defines the events that occur. Despite the predefined scenario, the course of the training is not completely known, because the trainee's behaviour is not exactly predictable. To provide effective training, it is important that the environment reacts to the trainee's actions in a realistic way, so that trainees can transfer the skills obtained in the virtual to the real world.

In most of the current systems for virtual fire-fighting training, virtual characters and other elements in the environment do not behave autonomously (e.g. [1]). Instead, instructors play a major role, managing changes in the environment, e.g. the size of a fire, and impersonating other characters in the scenario, e.g. a trainee's team-mates, police or bystanders, in such a way that learning is facilitated as much as possible. Besides controlling the environment and behaviour of virtual characters, instructors provide instruction, guiding, and give feedback to the trainee. As both of these tasks require ample attention of an instructor, more than one instructor is needed to train one trainee. Making use of even more instructors per trainee, however, is a costly and unpractical solution.

A way to alleviate the tasks of an instructor is by using artificial intelligence to perform (part of) the instructor's tasks. A lot of research has been done on intelligent tutoring systems (ITS) (for overviews, see [2, 3]). Successful applications mostly concern the training of well-structured tasks, such as programming and algebra [3]. Designing ITS for complex, dynamic, and open real-world tasks transpires to be difficult because it is not possible to represent the domain by a small number of rules and the space of possible actions is large.

Cognitive models are used to generate realistic behaviour of virtual characters (e.g. [4]). For achieving simple behaviour, cognitive models are successfully applied, but generating complex character behaviour is more difficult. One of the problems is that in complex situations, it is not always clear why a character acts the way it does. Even the system designers and the instructors, who should be able to explain its behaviour, can only speculate about the character's underlying reasons. Without knowing the motivation of a character's actions, it is more difficult for a trainee to understand the situation and learn from it.

This paper presents an approach for virtual training with fewer instructors. We propose a cognitive model that does not only generate behaviour, but also explains it afterwards, which should result in self-explaining agents. The paper has the following outline: we first discuss what explanations of virtual characters should look like (section 2), and then propose a cognitive model able to generate such explanations (section 3). Section 4 discusses the two uses of the model: generation and explanation of actions. Subsequently, requirements for the implementation of the cognitive model (section 5) and related work (section 6) are discussed. We conclude the paper with a discussion and suggestions for future research (section 7). We use the scenario at the beginning of the paper to illustrate the proposed principles and methods.

2 Self-explaining agents

Early research on expert systems already recognized that the advice or diagnoses from decision support systems should be accompanied by an explanation [5–7]. We transfer this idea to virtual training systems, where virtual characters should not only perform believable behaviour, but also provide explanations about the underlying reasons of their actions along with it. However, since there are many ways to explain a single event [8], the challenge is to develop a method to identify which explanations satisfactorily answer a trainee's questions. The field of expert systems distinguishes rule trace and justification explanations [5]. Rule trace explanations show which rules or data a system uses to reach a conclusion. Justification explanations, in addition, provide the domain knowledge underlying these rules and decisions, explaining why the chosen actions are appropriate. Research shows that users of expert system often prefer justification to rule trace explanations. Similarly, explanations of virtual characters should not only mention how, but also why a certain conclusion has been reached.

Humans use certain concepts when they explain their behaviour, they give explanations in terms of goals, beliefs, intentions, etc. Virtual agents should use similar vocabulary in order to produce useful explanations. To facilitate the generation of explanations in 'human' terms, agents can already make use of these concepts for the generation of their behaviour. Agents based on a BDI model reason with concepts such as goals and beliefs. Moreover, it has been demonstrated that BDI agents are suited for developing virtual non-player characters for computer games [9]. Because BDI agents use relevant concepts for the generation of their behaviour, explanations that give insight into their reasoning process are helpful to the trainee. For example, a BDI-based virtual character could explain its actions by referring to its underlying goals.

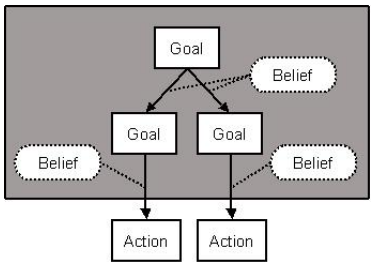
3 The cognitive model

As argued in the previous section, we will use the BDI paradigm to develop self-explaining agents. This section provides a model describing how beliefs, desires (goals), intentions (adopted goals) and actions relate to each other in our approach. The three basic elements in the model are goals, actions and beliefs. As illustrated in figure 1, a goal can be divided into sub-goals or achieved by executing a single action. Beliefs relate to the connections between goals and subgoals, and goals and actions. The three subsections each discuss one of them, and explain the relations to the other elements in the model more extensively.

3.1 Goals in the model

A goal describes (part of) a world state. All possible goals of an agent are present in its cognitive model and depending on the current world state, one or more of its goals become active. An agent's knowledge about the current world state is represented in its beliefs, determining which goals become active. For example,

Fig. 1. The relation between goals, actions and beliefs



a fire-fighter would only adopt the goal to extinguish a fire, if he believed that there (possibly) was one.

Goals relate to their sub-goals in four different ways. The **first** possibility is that all sub-goals have to be achieved in order to achieve the main goal, and the order of achievement is not fixed. For instance, the goal to deal with a fire in a house is divided into the sub-goals to extinguish the fire, and to save the residents. The completion of both sub-goals is essential to the achievement of the main goal, and both orders of achievement are possible. In the **second** relation, the achievement of exactly one sub-goal leads to the achievement of the main goal. The different sub-goals exclude each other in the sense that if an agent adopts one of the sub-goals, the other sub-goal(s) cannot simultaneously be active goals. For example, a main goal to extinguish a fire has the sub-goals to extinguish the fire with water and with foam. A fire-fighter has to choose between water and foam, but he cannot use both. In contrast to the first example, neither of the sub-goals is necessary, but both are possible ways to achieve the main goal. In the **third** goal/sub-goal relation, the achievement of one sub-goal leads to the achievement of the main goal, but the different sub-goals do not exclude each other. For example, sub-goals of the goal to find a victim are to ask other people, and to search the victim yourself. Different from the choice between water and foam, in this example an agent can decide to adopt both strategies. In the **fourth** relation, for achieving the main goal, all sub-goals have to be achieved in a specific order. For instance, the goal to rescue a victim in a burning house has the sub-goals to find the victim and to bring him to a safe place. The main goal is only achieved if the two sub-goals are achieved in the correct order.

The four goal/sub-goal relations are indicated by adding subscripts to the sub-goals, denoting the relation to their main goal. The different sub-goals are G_{and}, G_{xor}, G_{or} and G_{seq-i}, in the order as discussed in the previous paragraph, where *and*, *xor* and *or* refer to the logical operators, and *seq* refers to sequential. The i in G_{seq-i} stands for the position of a sub-goal in the sequence of sub-goals. It should be noted that the logical operators are biased, as some of the sub-goals are adopted more often than others. Each of the sub-goals can also be a main goal with new branches of sub-goals starting from it. Thus, the name of a goal is always derived from the relation to its main goal and the other sub-goals of

that main goal. Some goals can have more than one sub-script, i.e. if they relate to more than one main goal in different ways.

3.2 Actions in the model

Goals are specified by sub-goals, which are specified by new sub-goals, etc. At some point in the goal tree, goals are no further specified and can be achieved by performing one single action. Our model does not allow connecting more than one action to a goal for simplicity reasons. The different relations between two levels can be expressed at the goal/sub-goal level, so distinguishing different goal-action relations is unnecessary.

Basic actions should be carefully chosen in the model. If two actions are always followed by each other, e.g. the action to connect a fire hose and to open the tap, these two actions can be represented as one action in the model. A division of two actions (and thus two sub-goals because goals relate to only one action), while they are never used separately, is unnecessary. The determination of the smallest distinguishable actions does not only depend on the execution of actions, but also on the explanation of actions.

3.3 Beliefs in the model

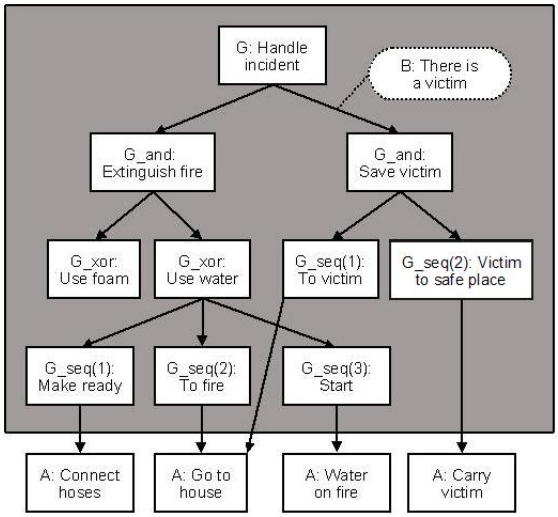
In our model, an agent's beliefs determine which of its goals become active, and thus which actions it takes. Agents have two types of beliefs: general and observation. General beliefs involve all them of which the truth does not depend on a particular situation, for example, the belief that combustion requires oxygen. Observation beliefs are context-specific and involve them about an agent's current situation. Observation beliefs are only true in a particular situation, for example, the belief that there is somebody in the burning house. It should be noted that observation beliefs are often interpretations of observations, which means that additional knowledge has been added. For example, the belief that a house is burning is an interpretation of observations such as smoke, heat, sounds of fire, etc. In general, it is hard to draw a line between observations and interpretations, but we will take a practical approach and consider such interpretations as observation beliefs.

In simple worlds, e.g. a block world, it is possible that an agent has an internal representation of all possible information about the current state of the environment. In more complex worlds, such as in virtual training systems, the environment contains lots of rapidly changing information and representing all of it is unfeasible. In our model, only beliefs that influence the agent's actions or that are needed for explaining them are made explicit.

4 Uses of the model

In this section, we illustrate how our model can be used for the generation and the explanation of behaviour. Therefore, we use the scenario at the beginning of

Fig. 2. The cognitive model of fire-fighter G



the paper. Figure 2 represents the internal state of virtual fire-fighter G in the scenario.

4.1 Generation of behaviour

The virtual characters in a training scenario fulfil specific roles with corresponding goals and tasks. For example, a dispatch centre operator should properly inform the leading fire-fighter, a policeman ensure order and safety, and a fire-fighter bring incidents to a succesful conclusion. The characters maintain their overall goal during the complete training session. To achieve their main goal, agents have to adopt proper sub-goals, sub-sub-goals, and finally perform the corresponding actions. The selection of goals depends on an agent's beliefs and the relation between that goal and its main goal (G_{and}, G_{xor}, G_{or} or G_{seq-i}). In the fire-fighting domain there are many rules prescribing what a fire-fighter should do in different situations, in other words, most decisions of fire-fighters are based on the matching of patterns with these rules [10]. The procedural nature of the fire-fighting domain facilitates the construction of a goal tree; the selection rules correspond to the actual rules in the domain. The selection mechanisms for the four different relations distinguished will be discussed later in this subsection.

Our approach to agent behaviour generation resembles planning methods based on hierarchical task networks (HTNs) [11]. In HTN planning, an initial plan describing the problem is a high-level description of what is to be done. Plans are refined by action decompositions, which reduces a high-level action to a set of lower-level actions. Actions are decomposed until only primitive actions remain in the plan. These initial plans, action decompositions and primitive actions correspond to our model's top-goals, divisions of goals into sub-goals and

actions, respectively. Russell and Norvig take an approach in which they combine HTN planning with partial-order planning (POP) [11]. POP refers to any planning algorithm that can place two actions into a plan without specifying which one comes first. The cognitive model also allows for POP, namely, sister sub-goals of the type G_{and} can be achieved in an arbitrary order. Further, Russell and Norvig argue that non-deterministic domains require conditional planning, and executing monitoring and re-planning [11]. Our model involves conditional planning by observation beliefs determining which goals become active, and execution monitoring and re-planning by sensing actions and belief updates.

In figure 2, fire-fighter G's main goal to bring the incident to a succesful end is divided into two sub-goals of the type G_{and}: to extinguish the fire and to save the victim. In principle, if a goal with G_{and} sub-goals is active, all its sub-goals are active, but only if the sub-goals do apply to the current situation. For example, only when fire-fighter G in the scenario obtained the belief that there was a victim the sub-goal to save the victim became active. Ideally, all active sub-goals are aspired simultaneously, but this is not always possible due to limited resources (e.g. there are insufficient fire-fighters available). The agent's model contains preferences for sub-goals in order to make choices between sister sub-goals. In the scenario, fire-fighter G judged the sub-goal to save the victim as more important than to save the building. Such preferences correspond to the rules fire-fighters were trained to use.

Before G saw the victim, its only goal was to extinguish the fire and G had to adopt either the G_{xor} sub-goal to use foam or to use water. G_{xor} sub-goals are possibilities of which one option has to be selected to achieve the main goal. In practice, one of the G_{xor} sub-goals will be the default way to achieve the main goal. For instance, water is used to extinguish a fire, unless some special circumstance arises, e.g. that there is no water. It should be noted that the xor relation is biased, as one of the G_{xor} sub-goals is selected more often than the others. Denoting a default sub-goal matches the way humans make decisions. Literature on natural decision making [12], which concerns decisions in complex, time-constrained, and sometimes stressful circumstances, stresses that humans in these conditions do not consider all possible options and make a choice. Instead, experienced decision makers use their experience deciding what to do, without considering other options; only if they encounter problems with their first choice, they will start considering other options.

If enough resources are available, all G_{or} sub-goals are adopted. When resources are constrained, only one or a few of the G_{or} sub-goals are adopted. Beliefs about preferences define which sub-goals are adopted by default in the last case. Subsequently, when one of the G_{or} sub-goals is achieved, all the others are dropped.

When a goal with sub-goals of the type G_{seq-i} is active, the first sub-goal of the sequence becomes active. After successful achievement of a G_{seq-i} goal, the next sub-goal, $G_{seq-i+1}$, becomes active. In our example, fire-fighter G has to connect hoses (G_{seq-1}) before going to the house (G_{seq-2}) and successfully putting water on the fire (G_{seq-3}).

4.2 Explanation of behaviour

For the generation of behaviour, the cognitive model is used in one direction: from goals to sub-goals to actions. A trainee interacting with a virtual character sees the actions of the character, but not its reasoning. If the trainee asks the character to explain why he performed a particular action, the model is used the other way: from actions to the underlying sub- and main goals. In an explanation, the beliefs and goals that were responsible for a particular action are made explicit. As the goal-tree was constructed according to the procedures in the fire-fighting domain, these procedures will be part of the explanations. However, a long trace of beliefs (observation and general) and goals can underlie one action, but a trainee often does not need all information. Out of the many goals and beliefs, the information that is most useful for the trainee needs to be selected. The particular information that answers a trainee's question depends on why the trainee asked for an explanation. In the next two paragraphs, two ways in which a trainee can be confused are described.

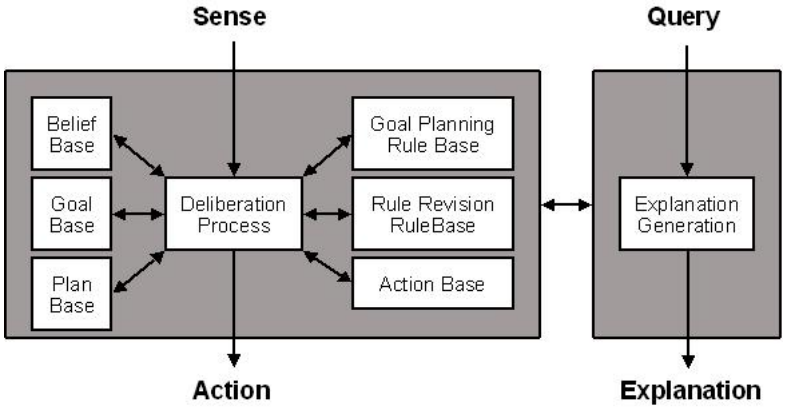
The first cause of confusion involves constructions with two goals that can be achieved by the same action or that have a connection to the same sub-goal. In the example, fire-fighter F does not understand why G runs to the house. Figure 2 shows that G's action can be performed in order to achieve two different goals, either to extinguish the fire, or to save the victim. Usually, by explaining for which of the two (or more) goals the action was performed, the trainee's confusion will be resolved. In the example, G can explain his action by stating that he ran to the house to save the victim. Another way would be to explain the observation belief that triggered the goal, in this case the belief that there is a victim. A third possibility is to provide the general belief why the observation belief makes the agent adopt the goal; in the example, saving people has priority over extinguishing fires.

Another type of confusion arises when, although the trainee understands for which goal an agent executes an action, he is surprised by the approach the agent takes. For instance, the trainee does not understand why an agent uses foam, if water is available (e.g. because of a possible chemical reaction of water with the substances involved). In such a case, it could be useful to ask the trainee what action he expected the agent to execute. With that information, one could go upwards in the goal hierarchy from the actual and the expected action, until both reasoning lines meet. The provided explanation informs the trainee why one and not the other line was chosen. Such an explanation could include an observation belief, e.g. there is a sticker informing about the properties of the burning substance, or a general belief, e.g. the picture on the sticker means that the substance has a chemical reaction with water.

5 Implementation

The implementation of our cognitive model requires an agent programming language in which beliefs and goals are explicitly represented. Moreover, because

Fig. 3. The 2APL architecture extended with an explanation module



explanations often involve goals *and* beliefs, an agent needs the ability to combine the two and reason with them. For instance, if fire-fighter G had the goals to extinguish a fire and save a victim, it would save the victim first. An explanation of G's action consists of more than just one goal or belief, namely, its two goals and its belief about their priority.

We plan to implement our model in 2APL [13], a programming language in which beliefs and goals are explicitly represented. Other languages that support BDI agents are, for example, 3APL and Jason/AgentSpeak [14]. 2APL agents can test whether they have a particular goal or belief, and add and delete them. However, 2APL agents cannot reason about their own goals and beliefs, which is crucial to the generation of explanations. Allowing an agent to reason with its beliefs and goals would lead to modifications to them, which could result in undesired loops. A solution is to add an explanation module to the 2APL architecture. In such a module, an agent can reason with its goals and beliefs without hindering the generation of actions by changing their original content. Figure 3 shows 2APL's architecture with a belief and a goal base (left part), and an explanation module added to it (right part).

The generation of explanations in the explanation module is not arbitrary. If beliefs and goals are distinguished, logical reasoning with the content of either beliefs or goals can be implemented, e.g. in a Prolog program. However, we argued that combinations of goals and beliefs are needed for explanation generation. So an agent has to distinguish goals and beliefs from each other, but also make connections between the two. Furthermore, agents should be able to reason about the beliefs and goals of other agents, possibly even in a nested way. This is more complicated to achieve in a Prolog program, and possibly other types of logics, e.g. modal logic, would be required.

6 Related work

Section 2 already discussed some of the literature on explanation research in general, here we will focus on self-explaining agents. One of the first systems in

which intelligent agents were able to provide explanations for their actions was Debrief [15]. 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. With the observations, Debrief determines what factors were responsible for the decisions. In contrast to our approach, Debrief derives what must have been the agent's underlying beliefs. However, as illustrated by the example in this paper, different reasons can be responsible for the same action. By making beliefs and goals in the reasoning process explicit, the actual reasons for performing an action can be given.

The XAI explanation component [16] was developed for a training tool for commanding a light infantry company and allows trainees to ask questions after a training session. The trainee can select a time and an entity, and ask the system about the entity's state, e.g. the agent's location or his health, but not about the motivations behind his actions. A second version of the XAI system [17, 18] was developed to overcome the shortcomings of the first; it claims to support domain independency, modularity, and the ability to explain the motivations behind entities' actions. This second XAI system is applicable to different simulation-based training systems, and for the generation of explanations it depends on information that is made available by the simulation. Unfortunately, it was found that most simulations do not represent agents' goals, and action preconditions and effects.

Another research area relating to our work is practical reasoning. In that context, Atkinson, Bench-Capon and McBurney [19] proposed the following reasoning scheme to construct an argumentation for an action. *In the current circumstances, we should perform action A, to achieve new circumstances S, which will realize some goal G, which will promote some value V.* Similar to our approach, Atkinson et al's reasoning scheme provides the reasons why an action was chosen. A difference is that they attach only one goal to an action, whereas in our model goals can branch into sub-goals. Another difference is that in their scheme states are represented as circumstances, and in our model states are represented in the beliefs of an agent. In complex environments, it is impossible to represent all circumstances, so instead we used a set of beliefs from the agent's perspective. A final difference is that their scheme contains values, which account for the fact that people may rationally disagree upon an issue. The differences in Atkinson et al's reasoning scheme and our cognitive model can be explained by the purposes for which they are used, namely, practical argumentation and generation of explanations of behaviour in the context of virtual training. Two argumentation partners can have different standpoints without one of them being right; they discuss to persuade each other. In contrast, explanation is not so much meant to change ones opinion as well as to provide insight. Especially in the context of instruction, an instructor is supposed to be objective and his explanations are assumed to be right.

7 Discussion and future research

In the present paper we have proposed a cognitive model for the generation and explanation of behaviour. The use of explicit goals and beliefs in an agent's reasoning process distinguishes our model from most other approaches of behaviour generation. Moreover, explanations generated by other accounts of explaining agents [15, 16] do not refer to the agent's beliefs and goals. We showed that the two types of confusions discussed in section 4.2 can only be clarified by referring to the agent's internal reasoning process, which stresses the importance of a representation in goals and beliefs.

The explanations generated by the model presuppose that the trainee already has basic knowledge about the domain. We made this assumption because novices start with learning separate procedures, and only advanced trainees practice with scenarios in which different procedures have to be applied. In scenario training, the trainee practices in giving priority to the right events and recognizing which procedures are required in which situation.

Effective training not only requires generating suitable explanations, but also selecting the appropriate time and form of delivery. For example, it should be decided whether explanations are given when a trainee asks for it or on the agent's initiative. Further, they can be given only after completion, but also during a training session. If so, the scenario could be paused or move on. Pedagogical knowledge is needed to decide when and how feedback should be presented to the trainee. However, these questions are beyond the scope of this paper.

The fire-fighting domain is characterized by many procedures, numerous aspects that have to be taken into account, and the requirement of fast decision making. These characteristics also apply to domains as crisis management and military command, so the cognitive model would also be useful in virtual training systems for these domains.

In the future, we plan to implement the proposed cognitive model in 2APL. The input of domain experts will be used for a more comprehensive cognitive model than the example in this paper. Further, we plan to develop an explanation module in 2APL, so that a 2APL agent can reason about its goals and beliefs. Once the model has been successfully implemented, it can be connected to a virtual training system for fire-fighting. Experiments with fire-fighter trainees should show whether the generated behaviour of the virtual characters in the training is believable, and whether the explanations they provide are useful. Moreover, such experiments should provide more insight in general to an agent's believability and usefulness in training systems.

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), and by the research programs 'Cognitive Modelling' (V524) and 'Integrated Training and Instruction' (V406), funded by the Netherlands defence organisation.

References

1. Houtkamp, J., Bos, F.: Evaluation of a virtual scenario training for leading firefighters. In B. Van de Walle, P. Burghardt, C.N., ed.: Proceedings of the 4th International ISCRAM Conference. (2007) 565–570
2. Murray, T.: Authoring intelligent tutoring systems: An analysis of the state of the art. International Journal of Artificial Intelligence in Education (1999) 98–129
3. VanLehn, K.: The behavior of tutoring systems. International journal of artificial intelligence in education (2006) 227–265
4. Van Doesburg, W.A., Heuvelink, A., Van den Broek, E.L.: Tacop: A cognitive agent for a naval training simulation environment. In M. Pechoucek, D. Steiner, S.T., ed.: Proceedings of the Industry Track of AAMAS 2005. (2005) 34–41
5. Ye, R., Johnson, P.: The impact of explanation facilities on user acceptance of expert systems advice. Mis Quarterly **19** (1995) 157–172
6. Dhaliwal, J., Benbasat, I.: The use and effects of knowledge-based system explanations: theoretical foundations and a framework for empirical evaluation. Information systems research **7** (1996) 243–361
7. Gregor, S., Benbasat, I.: Explanation from intelligent systems: theoretical foundations and implications for practice. MIS Quarterly **23** (1999) 497–530
8. Mioch, T., Harbers, M., Van Doesburg, W., Van den Bosch, K.: Enhancing human understanding through intelligent explanations. In Bosse, T., Castelfranchi, C., Neerincx, M., Sadri, F., Treur, J., eds.: Proceedings of the first international workshop on human aspects in ambient intelligence. (2007)
9. Norling, E.: Capturing the quake player: using a bdi agent to model human behaviour. In J.S. Rosenschein, T. Sandholm, M.W.M.Y., ed.: Proceedings of AAMAS 2003. (2003) 1080–1081
10. Lipshitz, R., Sender, A., Omodei, M., McLennan, J., Wearing, A.: What's burning? the r.a.w.f.s. heuristic on the fire ground. In Hoffman, R.R., ed.: Proceedings of the Sixth International Conference on Naturalistic Decision Making, CRC Press (2007)
11. Russell, S., Norvig, P.: Artificial Intelligence A Modern Approach. Second edn. Pearson Education, Inc., New Jersey, USA (2003)
12. Klein, G.: Sources of power: how people make decisions. MIT Press Camb. MA (1998)
13. 2APL. (URL: http://www.cs.uu.nl/2apl/)
14. Bordini, R., Dastani, M., Dix, J., Fallah-Seghrouchni, A., eds.: Multi-agent programming: languages, platforms and applications. Springer, Berlin (2005)
15. Lewis Johnson, W.: Agents that learn to explain themselves. In: Proceedings of the Twelfth National Conference on Artificial Intelligence. (1994) 1257–1263
16. Van Lent, M., Fisher, W., Mancuso, M.: An explainable artificial intelligence system for small-unit tactical behavior. In: Proceedings of IAAA 2004, Menlo Park, CA, AAAI Press (2004)
17. Gomboc, D., Solomon, S., Core, M.G., Lane, H.C., van Lent, M.: Design recommendations to support automated explanation and tutoring. In: Proceedings of the Fourteenth Conference on Behavior Representation in Modeling and Simulation, Universal City, CA. (2005)
18. 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** (2006)
19. Atkinson, K., Bench-Capon, T., McBurney, P.: Computational representation of practical argument. Synthese **152** (2006) 157–206