Proceedings of the IVA'10 Proceedings of the 10th international conference on Intelligent virtual agents
Pages 28-34; Springer-Verlag Berlin, Heidelberg ©2010

Using Artificial Team Members for Team Training in Virtual Environments

Jurriaan van Diggelen, Tijmen Muller, Karel van den Bosch

TNO Defense, Safety and Security
Soesterberg, The Netherlands
{jurriaan.vandiggelen, tijmen.muller, karel.vandenbosch}@tno.nl

Abstract. In a good team, members do not only perform their individual task, they also coordinate their actions with other members of the team. Developing such team skills usually involves exercises with all members playing their role. This approach is costly and has organizational and educational drawbacks. We developed a more efficient and flexible approach by setting training in virtual environments, and using intelligent software agents to play the role of team members. We developed a general framework for developing agents that, in a controlled fashion, execute the behavior that enables the human player (i.e., trainee) to effectively learn team skills. The framework is tested by developing and implementing various types of team agents in a game-based virtual environment..

Keywords: human-agent teams, virtual environments, team training

1. Introduction

A team of experts does not necessarily constitute an expert team. A team's performance is not only determined by how well the individual team members perform their tasks, but also by how well they practice their *team skills.* Not surprisingly, training team skills is an essential part of the education of many professions demanding a tight collaboration [10]. For example, fire fighters not only learn how to operate a fire hose (task behavior), but also how they must communicate relevant information about the fire's progress to their colleagues (team behavior).

Usually, team skills are trained by exercises with a (potentially large) number of participants, each playing a dedicated role in a given scenario. Such large scale team exercises have a number of drawbacks. First, this type of team training requires ample resources (both human and financial) and is difficult to organize. Furthermore, it occurs frequently that not all members of a team are available for training at the same time and location, and that role players (e.g. staff personnel) need to take their place. It can also be that the team members vary in competence, thus having different training needs (e.g. a team consisting of novice and advanced trainees). For example, an urban patrol scenario contains many roles which are either not relevant for soldiers practicing their team skills (e.g. *local inhabitant)*, or for which some soldiers holding that rank have already completed their training.

We propose to approach this problem by combining recent developments in virtual environments (VEs) and intelligent software agents. We aim at developing a game-based platform for team training in which some team members are played by humans, and some are played by software agents. In this way, team training can be made less costly, as not all roles have to be fulfilled by humans anymore. Furthermore, training can be better tailored to the specific learning objectives of a specific individual, as we can control more precisely the team behavior of artificial team members than that of human team members (which may be trainees themselves).

Whereas computer-controlled characters in VEs are common in computer games, these characters are usually programmed with scripts which list a number of trigger/event rules. The type of behavior required for teamwork is too complex to be programmed in these scripting languages. Therefore, we have developed a framework for developing software agents whose behavior in a VE is based on concepts as goals, plans, and beliefs, and which is "aware" of its team.

The behavior of the software agent is directed by a teamwork model. Teamwork models have been studied for several decades by the AI community, e.g. [12]. Typical applications of these models are mixed human-agent teams [11], where computers assist humans in a flexible way, i.e. as a teammate. Our application to team training is novel, as our prime interest is not to develop a "perfect" team member which is always subjected to humans. Rather, we aim at developing a realistic team member which exhibits natural behavior and from which humans can *learn*. This makes human trainees better team players "in the real world" after they have completed their training. For example, a real team member sometimes forgets to communicate an important piece of information, for instance if he is too busy with other tasks. Implementing this property into an artificial team members allow us to design realistic experiences, from which the trainee can learn how to deal with such team task problems.

Our research objectives can thus be formulated as follows. Firstly, we aim at developing generic teamwork models tailored to training fundamental team skills. Secondly, we aim to implement these models to develop an agent-based artificial team member, situated in a virtual environment. This paper describes the concepts and design of our agent-based team training approach, followed by a proof-of-concept study, showing that it is a feasible alternative to human-based team training. We demonstrate our approach by presenting *Samurai*, a software agent that in a virtual environment performs a team task in a coordinated fashion with a human player (trainee).

The paper is organized as follows. The following section describes how teamwork is implemented in our system. Section 3 describes the implementation of Samurai and the integration of Samurai with the virtual environment. Section 4 gives the training case and event traces. A conclusion is given in Section 5.

2. Training of Teamwork

Teamwork is a topic of great complexity and breadth and no consensus has yet been reached on an exact definition. Most simply, a team is a group of people working together to achieve a common goal. This requires team members to maintain *common ground*, to be *mutually predictable*, and to be *mutually directable* [6]. Related literature reports different ways in which these properties can be implemented in artificial team members, with different levels of sophistication. We have implemented Samurai as follows. Samurai maintains common ground with the other team members by actively sharing relevant pieces of information with its team mates. Mutual predictability is achieved by applying an organizational structure to the team, which is commonly known by all participants, and describes how the different team members (both human and agent) fulfill their tasks. Mutual directability is implemented by a *request* protocol, which allows one agent to ask another agent to perform an action.

Whereas we believe these properties to be essential for team membership, *how* these properties reveal themselves in behavior may vary. By explicitly modeling such differences in artificial team members we can expose trainees to different types of team behavior. In this paper, we will focus on two characteristics of the team member's behavior: unprovocative and provocative. Unprovocative indicates that the team member behaves in accordance with the procedures and goals of the team as a whole. This behavior imposes no challenges to fellow team members recognizing and correcting any omissions. In contrast, provocative behavior of a team member conflicts with good team behavior, thus requiring other team members to stay alert and to bring about corrections. These characteristics are shown in the rows of Table 1, , for each of the three teamwork properties discussed above. This leads to eight different possible configurations for team training.

		Team property		
		Mutual predictability	**Common ground**	**Mutual directability**
Training	**Unprovocative**	Organization aware	Always shares relevant information	Always obeys a request
	Provocative	Not organization aware	Sometimes fails to share relevant information	Sometimes refuses a request

Table 1. Different team member characteristics

In the upper cell of the first column, unprovocative mutual predictability is described as (fully) organization aware. The team members have complete knowledge of their organization, which team member fulfills which roles in the team, and how they perform their work. In provocative training, the team members are not completely aware of the organization they are part of. This makes it more difficult for Samurai to predict the behavior of the trainee and the other way around. To overcome this issue, the trainee should learn to communicate with Samurai about his own role in the organization and also to find out about the role of Samurai himself.

In the second column, the different possibilities for common ground are shown. In unprovocative training, the team members always communicate relevant information with each other. In provocative training, they might sometimes fail to do this, because they forget it, or are too busy with other things. This requires the trainee to learn to actively collect relevant information.

In the third column, two possibilities for mutual directability are shown. On one extreme, the team members act as obedient servants, who blindly follow a request, regardless of what they are currently doing or what their own individual goals are. Whereas such behavior may appear convenient to the trainee, it is not necessarily in the team's interest, nor is it very realistic team behavior. A natural team member should be allowed to refuse a request, typically based on information that the requester does not have. This type of team member is willing to fulfill a request, when it believes this to be in the interests of the team. When it believes that its own plans are currently more important for the team, then the team member refuses the request.

3. Artificial Team Members

In this section, we will explain how we implemented the different types of team behavior outlined in the previous section. We will first explain the task behavior of an agent. Then, we will focus on team aspects.

The Cognitive Architecture

For the implementation of Samurai a cognitive architecture was developed in C++. The architecture consists of a framework for specifying practical reasoning rules, goals, beliefs, plans, together with an organization module including roles and role enactment, a dialogue module for processing natural language and a deliberation cycle.

Following Bratman's theory of practical reasoning [2], we model an agent's behavior by specifying *beliefs*, *goals* and *plans*. As is common for BDI-agent platforms (e.g. 3APL [6]), we use practical reasoning rules (or *PR-rules*) to determine how actions result from beliefs, goals and plans.

A PR-rule has the form *Head* ← *Guard* | *Body*. The head describes goals which form the activation event of the rule; the guard contains beliefs (using a predicate formula) that may or may not match with the agent's belief base; and the body describes the plan that is adopted when the rule fires. The activation and firing of rules is performed by the *deliberation cycle (*explained in Section 3.2).

As explained in Section 2, team members can know about each other's actions by using a shared organization model. Following well-known agent organization models, such as Moise [6], we define an organization as a number of *roles* (e.g. leader, scout) together with an *enactment model*, specifying which agents enact which roles. A role is specified as a set of goals and PR-rules. By enacting a role in the organization, the agent assumes the goals of the role (telling it *what* it should achieve), and it assumes the PR-rules (telling it *how* it should achieve the goals). Note that the agent may also have *individual* goals and PR-rules. These goals and PR-rules, contrary to the ones specified for its role, are not accessible to other agents.

A key factor in this approach is that the organization model is commonly known by all agents and is used by all to determine *information relevance*. As argued by Castelfranchi [4], information relevance for BDI agents should be examined in relation to their goals. Because the shared organization model allows agents to know some goals and PR-rules of other agents, they can proactively share information with other agents when they believe this helps them to achieve their goals.

This use of organization models not only serves the purpose of implementing behavior of individual agents in a team, but it also establishes the team properties 'mutual predictability' and 'common ground' among the team members. If a team consists of both virtual members (agents) and human members, then agents should be able to determine the role and tasks of the human players. As a consequence, the behavior of humans also needs to be captured in terms of goals and PR-rules. This allows a software agent to assess and reason about the goals and plans of humans too.

Deliberation Cycle

The deliberation cycle tells the agent what it should do next. This cycle is illustrated at the left hand side of the following figure:

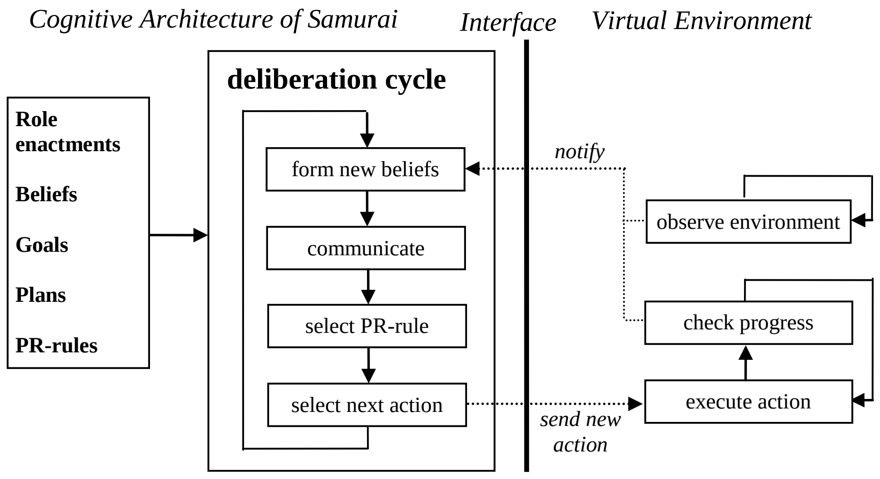

Figure 1. Cognitive architecture and interaction with Virtual Environment

Given this process, unprovocative behavior is achieved if Samurai starts fully aware of the team organization (i.e. appreciating the roles of all team members), immediately shares all relevant information and always follows requests. The provocative training modes described in Table 1 are implemented by restricting access to the organization model (*Provocative mutual predictability*), applying a random 'forget' function to the sharing of relevant information (*Provocative common ground*), or by making Samurai attribute low priority to unmotivated incoming requests (*Provocative mutual directability).*

Interaction with Virtual Environment

Samurai, as well as the other members of the team, move around in a virtual environment. SABRE [8] was chosen as virtual environment for our experiment, as it was used before by NATO to study military team behavior. Furthermore, Neverwinter Nights, the commercial game SABRE is based on, offers a low-level scripting language with such functionality as locating specific objects and retrieving their attributes, and communicating with the (human) player through text messages.

For Samurai to be able to interact (i.e. sense and act) with this virtual environment an interface is needed. The outline of this interface is shown in Figure 1: the left hand side shows Samurai's architecture, while the right hand side shows the observation and execution model that is executed in parallel. The action execution mechanism is implemented in the virtual environment, because this environment defines which actions Samurai can take. Consequently, the deliberation cycle runs in parallel with the action execution mechanism, so the process of deciding what to do next does not halt action execution and Samurai can choose to interrupt an action at any time.

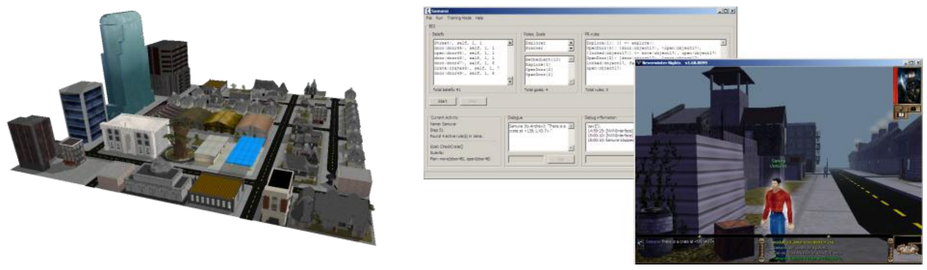

Figure 2. The environment (left), the Samurai/SABRE interface (right)

4. Team Training Case

Using the Samurai/SABRE software framework described above, we implemented a cognitive model for a team training scenario. The scenario takes place in a small city with residential and commercial areas. The task of the team is to find, identify and neutralize weapon crates while maintaining the goodwill of the local population. To accomplish this, the team has various resources at its disposal, such as weapon scanners to scan the contents of a crate and lock-picking sets to open locked crates and doors. Results are measured in terms of goodwill of the population – for example, goodwill increases when weapon crates are neutralized by collecting its weapons, but decreases when private crates (i.e. not containing weapons) are opened.

We defined the team, using several roles: an *explorer* explores the environment to find and locate crates; *weapon scanner* has a tool to check if a crate contains weapons; a *lock picker* has a tool to break the lock of a crate; a *weapon collector* can remove any found weapons from the crates.

SABRE is representative for a team task, as its team members require each other's help to achieve the team goaland tools are distributed among the team members. Additionally, team coordination and cooperation is necessary for high performance, because there is limited time to execute the mission.

Depending on the objective of the training, Samurai can be set to provocative or unprovocative for each of the team properties listed in Table 1. These settings will affect the processes during training and consequently the team performance. We have tested our prototype by performing a pilot experiment in which trainees interacted with Samurai in provocative and unprovocative mode. Demonstration videos can be found online [13].

5. Conclusion

In this paper, we have proposed a generic architecture for modeling virtual team members for team training. The architecture was used to implement a virtual team member called Samurai for a search task in the virtual environment SABRE. The team behavior of Samurai can be controlled on different dimensions, making him provocative or unprovocative on multiple team properties. Consequently, the difficulty of cooperating with Samurai can be adapted to the ability of the human trainee, causing different learning experiences.

In the future, we plan to investigate empirically the effects of different levels of provocativeness on the learning of the trainee. Also, we plan to extend the software framework, allowing our virtual team member to be used in other virtual environments as well, such as VBS2.

References

[1] Baker, D.P., Salas, E. (1992), Principles for measuring teamwork skills, Human Factors, Vol. 34 pp.469-75.

[2] Bratman, M.E., Israel, D.J. and Pollack, M.E. (1988), "Plans and resource-bounded practical reasoning" Computational Intelligence 4 349-355.

[3] Burton, R.M., DeSanctis, G., Obel, B. (2006), Organizational Design, Cambridge University Press.

[4] Castelfranchi, C. Guarantees for autonomy in cognitive agent architecture. In Intelligent Agents, volume 890 of LNAI, pages 56-70. Springer Berlin 1995.

[5] J. T. Doswell, "Pedagogical Embodied Conversational Agent," icalt, pp.774-776, Fourth IEEE International Conference on Advanced Learning Technologies (ICALT'04), 2004

[6] Hübner, J. F., Sichman, J. S., and Boissier, O., (2002). A Model for the Structural, Functional, and Deontic Specification of Organizations in Multiagent Systems. In Proc. of the 16th Brazilian Symposium on AI, LNCS, vol. 2507. Springer-Verlag

[7] Klein, G. Woods, D., Bradshaw, J.M. Hoffman, R.R. , Feltovich, P.J. (2004). Ten Challenges for Making Automation a Team Player. In Joint Human-Agent Activity," IEEE Intelligent Systems, vol. 19, no. 6

[8] Leung, A., Diller, D., & Ferguson, W. (2005). SABRE: A game-based testbed for studying team behavior. Proceedings of the Fall Simulation Interoperability Workshop (SISO). Orlando, FL, September 18-23, 2005.

[9] J. Rickel, W.L. Johnson (2002), Extending Virtual Humans to Support Team Training in Virtual Reality, in Exploring Artificial Intelligence in the New Millenium, Morgan Kaufmann Publishers, 2002.

[10] M. O. Riedl and A. Stern (2006) Believable Agents and Intelligent Scenario Direction for Social and Cultural Leadership Training. Proceedings of the 15th Conference on Behavior Representation in Modeling and Simulation, Baltimore

[11] Sycara, K., and Lewis, M. (2004). Integrating intelligent agents into human teams. In Team Cognition: Understanding the Factors that Drive Process and Performance,, 203-232. Washington, DC: American Psychological Association.

[12] Tambe, M. (1997). Towards Flexible Teamwork, Journal of Artificial Intelligence Research, pp. 83-124

[13] D. Traum, J. Rickel, J. Gratch, S. Marsella (2003), Negotiation over Tasks in Hybrid Human-Agent Teams for Simulation-Based Training, Proceedings of the second international joint conference on Autonomous agents and multiagent

[14] Cognitive Models Group, TNO, http://cm.tm.tno.nl/index.php/en/virtual-team-member