Adaptability properties of a Collaborative Virtual Environment for Training

Stéphanie Gerbaud, Valérie Gouranton, Bruno Arnaldi

To cite this version:

Stéphanie Gerbaud, Valérie Gouranton, Bruno Arnaldi. Adaptability properties of a Collaborative Virtual Environment for Training. Edutainment, 2009, Banff, Canada. pp 316-327, 10.1007/978-3-642-03364-3 . hal-00444137

HAL Id: hal-00444137

https://hal.archives-ouvertes.fr/hal-00444137

Submitted on 5 Jan 2010

Adaptability properties in collaborative virtual environments for training

Stéphanie Gerbaud[1,3], Valérie Gouranton[2,1,3], and Bruno Arnaldi[2,1,3]

[1] INRIA, Centre Inria Rennes - Bretagne Atlantique, Rennes, France
[2] INSA de Rennes, IRISA (- UMR6074), Rennes, France
[3] Université Européenne de Bretagne, France
[4] {stephanie.gerbaud,valerie.gouranton,bruno.arnaldi}@inria.fr

Abstract. In order to be used in various contexts of training, a desirable property for a virtual environment for training, especially a collaborative one, is **adaptability**. In this paper, we identify two levels of adaptability the environment should offer. The first level concerns the parametering of the components of the application, especially the scenario and the actors' behavior. The second level appears during the execution of the application where the actors should be able to adapt the scenario execution to the context. The models proposed in this paper fully satisfy those requirements. The scenario, the virtual world, the actors and the training session can be parametrized. Moreover, an action selection mechanism enables the actors to dynamically choose an action to perform depending on both the current context and the past actions. Finally we validate our models by presenting an illustrative scenario within the GVT environment.

Key words: virtual environment, training, adaptability

1 Introduction and related work

Virtual training can be applied to various training situations so that the requirements for a virtual environment change depending on the training context: number and type of people involved (from individual to collaborative training, supervised by a trainer, working with virtual humans), the object of the training (e.g. procedure, technical gestures, decision skills), the pedagogical expectations, etc. For instance in figure 1, we can see two different training situations: on the left we can see an extract from a military maintenance procedure; a trainee uses a remote control while his virtual partner is pulling a cable. On the right, the trainee and his partner (real or virtual) collaboratively assemble a piece of furniture. In an ideal world, a virtual environment for training should be able to take into account those different demands and show adaptability even dynamically. As an introduction, let us take an actual instance of adaptation. Due to an error, the trainee injured his hand but he must perform a sequence of action that requires both his hands. He starts on this sequence but he finds himself in a difficult situation where he can no longer perform any action. His partner detects that and interrupts what he was doing in order to help him. This reaction can be obvious for a real partner, but we would like a virtual partner to automatically detect that situation as well and to behave the same way.

Fig. 1. Different training situations

A virtual environment for training is composed of three major elements: the virtual world, the scenario and the actors (either real or virtual). The adaptation can then occurs on each of these elements. Now we will detail different level of adaptation proposed in virtual environments for training.

Scenario description Depending on the scenario language used, a scenario may be very precise whereas another may turn out to be more flexible. Indeed, three major ways exist to transcribe a procedure to accomplish into a scenario. The first one consists in writting all the actions to perform. This way of description provides a scenario not adaptable at all but it enables the system to control that the trainee strictly respect the reference procedure. For instance, it was the choice made in the first version of LORA [1] since it was designed for training on military maintenance procedures. The second way is to underspecify the scenario by describing only key actions while letting basic actions implicit. A reasoning algorithm is then used to infer on the actions to perform in order to satisfy the preconditions of key actions. In the MASCARET model [2] used in SecuReVi [3] the scenario describes *actions trade* (e.g. for firemen: *sprinkling a fire*) while letting implicit actions at a generic semantic level for human (e.g.: *going at a point*). This method enables the actors to adapt their actions to the context while respecting the scheduling of key actions. The third way is to use plans of action with constraints and goals. In STEVE [4] a naval maintenance procedure is described thanks to tasks, each of one being written as a standard plan representation [5]. A task is composed of steps and makes explicit causal links and ordering constraints between these steps. This type of description makes the re-adaptation of the scenario possible when a non-predicted event happens. Nevertheless, this flexibility can be a problem when the aim is to learn an exact procedure since the system can not ensure that the trainees will follow a pre-specified sequence of steps.

Distribution of scenario actions We have seen that is is possible in virtual environments for training to describe a scenario adaptable to the context. When these environments are collaborative, the scenario could also be adaptable concerning the distribution of scenario actions between actors. Such flexibility is conceivable when the application makes possible either a specification of severall roles allowed to perform a scenario action or a specification of one role but with more than one person playing the same role. In STEVE [4], both a trainee and his tutor can perform a given scenario action since the tutor plays the same role as his associated trainee. Nevertheless, there is no ambiguity about whom will perform this action (the trainee has priority and if he asks for help, his tutor may perform the action). In order to be fully exploited, the flexible distribution of scenario actions between actors must be associated with a mechanism in charge of dynamically determining the best candidate for an action depending on the current context. Such a mechanism is sometimes proposed for teamwork simulation (e.g. in [6]) but not in collaborative virtual environments for training where the distribution of scenario actions between actors is static rather than dynamic [7].

Virtual humans' behavior When a collaborative virtual environment for training offers virtual humans, another type of adaptation concerns the behavior of these virtual humans. Indeed, various authors suggest that virtual humans should play pedagogical roles in order to enhance the training. Chou et al.s [8] identified two categories of pedagogical roles: teacher (coach, tutor, guide) and learning companion (collaborator, troublemaker, etc). In STEVE [4], agents can play two roles: either tutor (who can answer trainee's questions or show the actions to perform) or substitute for missing team members (who simulates the behavior of an expert trainee). This last pedagogical role is the only one proposed in SecuReVi [3]. Even if the learning companion category contains many different roles, in existing virtual environments for training only one role is available which consists in following the procedure and performing the scenario actions expected at the right moment. As a consequence, virtual humans in virtual environments for training do not show much adaptability in their behavior since they only perform expected actions, without making any mistake.

After this short state of the art about adaptability in virtual environments for training, we will expose in section 2 the adaptability requirements we have identified; then we will describe the models we developed in order to satisfy those needs in section 3 followed by an illustrative scenario in the GVT environment in section 4.

2 Needs for adaptability

In our state of the art, we have noticed a lack of adaptability in virtual training applications and especially in collaborative ones. Indeed, such training applications should answer various level of demand from the application (depending on the training domain) as well as from the users (trainees and trainer). For instance a military training demands that trainees respect a strict sequence of actions whereas a do-it-yourself training expects the trainees to adapt themselves to the situation. In order to answer a wide range of requirements from various training situations, a virtual environment for training and especially a collaborative one needs to be able to show adaptability. The first adaptability level consits in enabling the session creator to set the parameters of the application in order to adapt them to the training session he would like to create. This level will be detailed in section 2.1. A second level of adaptability appears during the runtime, when the application is able to dynamically adapt itself to a changing context, as mentioned in section 2.2.

2.1 First adaptability level: parametering

Virtual world The virtual world is made of behavioral objects. The session creator should be able to easily create a reactive virtual world adapted to the training session, by adding objects and giving them interaction capabilities [9].

Virtual humans' behavior A specific behavioral object that sometimes appears in collaborative virtual environments is the virtual human. In a training session, the trainer should be able to parametrize the virtual humans' behavior (for instance by giving them a pedagogical role) depending on the training level of the trainee. For example, an inexperienced trainee could be associated with collaborative partners who could help him when needed whereas an expert trainee could be face with disturbing partners (who tends to monopolize the required tools for instance) in order to evaluate his capability to complete the procedure in a perturbing situation.

Scenario The author must transform the procedure to teach into a scenario written in a scenario language. We have shown that depending on the language used, the resulting scenario can turn out to be more or less flexible. The author should have a scenario language that let him decide on the level of adaptability of the scenario, and on the way to make it flexible. We have pointed up two ways to make a scenario adaptable in our state of the art: by underspecifying the scenario and by allowing severall people to perform a given action.

Application configuration The session author should define the performances to evaluate in a specific training session: for instance the displacements (the actors should minimize their movements), the number of actions (the actors should avoid to exchange tools and try to reuse the ones they already own), the continuity in the scenario actions (the actors should complete a sequence of actions before switching to another), etc. Thus, the actors should choose what action to perform depending on those performance criteria.

2.2 Second adaptability level: dynamic adaptation

Once the configuration of the training session is made, we can execute it. The second level of adaptability then appears.

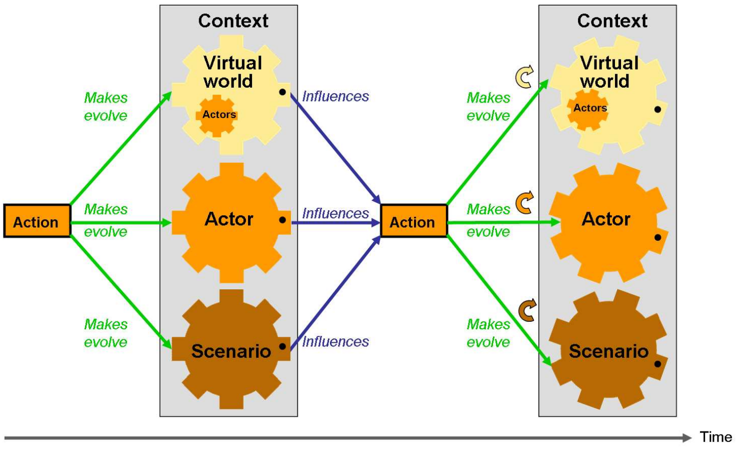

Fig. 2. Adaptation to the context (one actor involved)

Adaptation to the context The adaptability concerns here the actions performed by the virtual humans or suggested to the trainees that should depend on the context. On figure 2, a cog wheel represents a state that can evolve. Thus, the context is made of the current state of the world (including the current state of the actors) and the current state of the scenario. On this diagram, we can see that the action performed by an actor is influenced by the current context (blue arrows) and then makes this context evolve (green arrows which make the cog wheels turn). Thus, we expect the actors to adapt the scheduling of actions (scenario actions or implicit actions) to the context. For instance if a trainee already owns a hammer, he should not be asked to take a hammer before nailing.

Adaptation to the activity of other actors Since we are in a collaborative context of training, the virtual humans should also adapt their actions to the activity of their partners (their previous actions) in order to optimize the distribution of the actions between them and to avoid conflictual situations (e.g. if two actors plan to perform the same action). On figure 3, we can see the influence of the context like on figure 2, but here two actors are involved (the context contains two actor states). In addition to the context, the action of an actor is also influenced by the previous action of his partner; Once performed, this action will then influence the next action of the partner, etc. The actors should also try to help their partners when they are in a difficult situation where they can not perform any action (you can see an example of *implicit collaboration* in [7]).

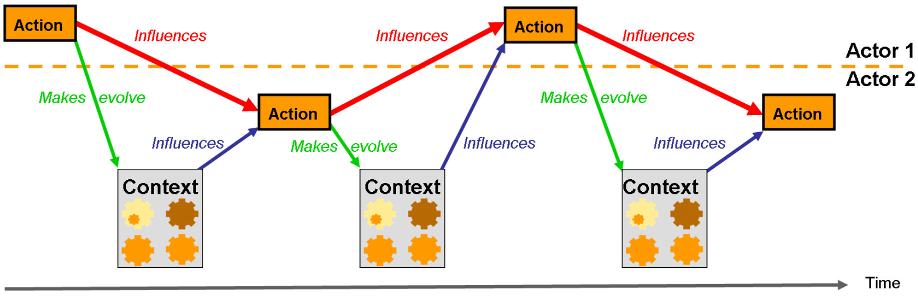

Fig. 3. Adaptation to the activity of partners (at least 2 actors involved)

Substitution of a trainee by a virtual human Last, for pedagogical use, it could be interesting to offer a trainee the possibility to be temporarily replaced by a virtual human in order to see the actions to perform.

3 Models involved in those adaptability properties

In order to offer these different levels of adaptability, both during the configuration of the application as well as during its execution, we developed models illustrated on figure 4. On this diagram, the parametrized elements appear on hatched areas whereas the rack-and-pinion gear represents the dynamical part where the different states jointly evolve during the training session, thanks to the action selection process also represented.

3.1 Configuration of the application

To configure the application, we must parametrize different components: the scenario, the virtual world, the actors and the training session. Those parameters appear in hexagons on figure 4 and in bold in the following sections.

Description of the scenario actions We extended an existing scenario language, LORA [1], used to describe individual scenarios for virtual training applications, in order to make the writing of collaborative scenarios possible. There

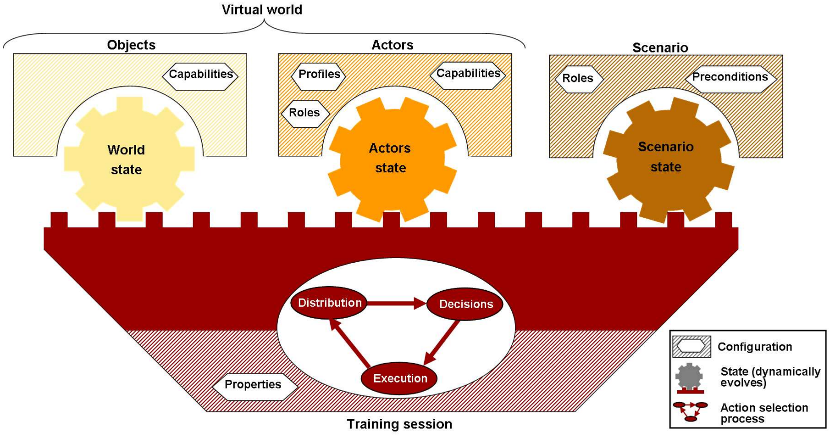

Fig. 4. Models involved in the adaptation

are two major additions [10]. The first one consists in specifying the **roles** allowed for each action, associated with a priority. Thus, we propose a flexible distribution of scenario actions between actors contrary to existing virtual environments for training which imposed a fixed distribution (given that only one role can be associated to each scenario action) [7]. The second one consits in underspecifying the scenario, by making implicit some basic actions: taking a tool and putting an object down. Instead of writting all the actions in the scenario, the author only describes key actions and use **preconditions** in order to specify the state of the hands required for these actions. The actors can then perform these basic actions whenever they want. In short, the scenario describes a fixed scheduling of the key actions while letting the actors dynamically determining the distribution of these actions between them and the moment when they want to perform basic actions.

Parametering of the virtual world In order to give the trainees a world they will be able to interact with, we must incorporate behavioral objects in it. Behavioral objects are objects endowed with **interaction capabilities**, which enables them to interact with each other and with trainees (real or virtual). The objects can then benefit from a generic interaction mechanism based on generic interaction links. These links can make any objects enter in interaction as long as they have compatible capabilities (see [9] for more details).

Parametering of an actor In order to be able to interact, an actor is a behavioral object and, as such, he also has **interaction capabilities**. An actor also has a **role** in order to know what he should do in the scenario. The role gives the actor who plays it some additional capabilities (for instance the capability to give order for a leader and the capability to control a pressure level for a technician). The behavior of an actor is defined in his **collaborative profile**. It is composed of a set of weighted behavioral rules and could correspond to a pedagogical role such as those mentioned in our state of the art. Some basic profiles are available but it is easy to create new ones by combining different behavioral rules. We can for instance create an actor who tends to make the procedure progress or on the contrary who try to disturb a trainee.

Parametering of the training session The creator of the training session must define general **properties** for this session. These properties define global trends or performance criteria for a specific session. The actors will then try to respect those properties when they will select an action to perform. For instance, the creator can specify if it is more important to complete a scenario sequence of actions before switching to another or to reuse an object an actor already owns.

3.2 Dynamic execution: action selection mechanism

Each of the different components which can be parametrized also has a state which dynamically evolves. The set of these states at a given time (objects, actors and scenario) forms the current context. The actors have to decide what action to perform depending on that context. Moreover, due to our description of the scenario, there may be more than one actor allowed to perform a scenario action. The action selection mechanism is the mechanism in charge of dealing with the parametering described in the previous section in order to dynamically adapt the execution of the training session to the current context. The action selection mechanism aims at first determining the best candidate depending on the properties defined in section 3.1; then choosing an hypothetical action for each actor; at last performing an action chosen by either a real user or a virtual human. Those steps are illustrated on figure 4. Now, we will shortly describe each one of these steps. More complete information about the action selection mechanism can be found in [7].

Action distribution The action distribution module aims at analyzing all active scenario actions, that is all scenario actions that correspond to actions currently allowed in the procedure. It ranks the different actors (real and virtual) for each possible scenario action according to their abilities to make the procedure progress, taking into account the properties mentioned in section 3.1. Consequently, this module is able to dynamically evaluate who the best candidate for each action is. Then, this information is transmitted to the actors in order to suggest them a good way of sharing the scenario actions in order to make the scenario progress.

Decisions The module of action distribution proposes a global distribution of actions between actors, respecting the scenario requirements. But it is only a suggestion and it is up to each actor to make an individual choice thanks to their own decision-making module (one decision per actor). This module uses the behavioral rules contained in the actor's collaborative profile (see section 3.1) to choose one action.

Execution of the action Once an actor has decided to perform an action, this action is executed in the same way, no matter if this decision comes from a real user or from a virtual human. Indeed, an actor can be either a real user (trainee or trainer) or a virtual human. For both types of actor, the previous steps in the action selection mechanism are exactly the same and so is the execution step. The only difference remains in the link between the decision step and the execution step. For a virtual human the selected action will automatically be performed, whereas for a real user, this action may be suggested as a pedagogical piece of advice, but it is up to the user to select an action to perform (see figure 5). Since this link is the only difference in the modelling of the behavior of a real user and a virtual human, a real user can easily and dynamically be replaced by a virtual human, and reversely, only by changing the connection between the decision-making step and the execution step. The execution of the action leads to a new context and then the action selection process loops back.

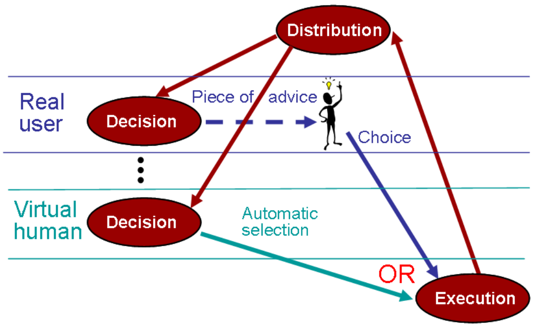

Fig. 5. Connection of a real user or a virtual human

4 Validation in GVT

4.1 Industrial application: the GVT project

The GVT[5] (Generic Virtual Training) project is developed in a research/industry collaboration, involving three partners: INRIA and CERV laboratories and Nexter Systems. This latter partner is a French company specialized in military equipment, such as the Leclerc tank. GVT is a software platform for building virtual training applications in which trainees have to learn a procedure (as

---

[5] http://www.gvt-nexter.fr/

opposed to technical gestures). The commercialized version of GVT proposes individual virtual training on procedures such as maintenance procedures or diagnosis procedures. A prototype is also available for virtual training on collaborative procedures where real users and virtual humans collaborate (see figure 1). The models proposed in section 3 have been implemented within this prototype. More information about GVT can be found in [11] and [12].

4.2 Illustrative scenario: cupboard assembly procedure

As an illustrative example of the adaptability possibilities offered by the models we have introduced in this paper, we take a "do-it-yourself" scenario which consists in collaboratively assembling a kitchen cupboard. This scenario, which actually runs under the latest version of GVT, had been written based on true visual assembly instructions for a kitchen cupboard (from a well known Swedish trademark) and contains more than 100 actions including collaborative ones (e.g to hold planks). This scenario is interesting because the assembly instructions describe the different actions to perform and the number of people required (at least two) but they did not make clear the distribution of those actions between the involved people. The distribution is thus flexible and should be adapted to the context.

For this scenario, the configuration phase consisted in traducing the visual assembly instructions into a scenario written in LORA, with preconditions when tools or free hands are needed and with several sequences of actions that could possibly be done in parallel. We did not specified any role for scenario actions, which means that any role is permitted. We have created different collaborative profiles: active learning companion, lazy partner and troublemaker. The active learning companion simulates the behavior of a trainee, he performs the action he is expected to and helps the trainee when needed. The lazy partner only performs actions when one of his partners is at a deadlock where he can not perform any action (for instance he is waiting for a partner to perform a collaborative action or he has his two hands busy but he is not allowed to free them for the moment). The troublemaker tends to steal objects to the trainee and tries to perform unexpected actions (which can lead to conflict with the trainee on the choice of actions).

Let us take a short sequence in this scenario where there is some screws to place on a plank and then to screw. These two sequences of actions (to place the screws and to screw them) may be performed in parallel. If the trainee is faced with a lazy partner, he will have to perform all these actions. On the contrary, if he has an active learning companion, his partner will try to perform complementary actions. If the trainee takes the screwdriver, the virtual human will automatically deduce (thanks to the distribution step) that the trainee is willing to screw the screws and thus he will take a screw in order to place it (see figure 6, on the left). Next, if the trainee decides to put the screwdriver down and to take a screw, the virtual human will deduce that the trainee is willing to place the screws and then he will take the screwdriver in order to screw the placed screws (see figure 6, on the right). We can see that the virtual

Fig. 6. Adaptation in GVT

human dynamically adapts his actions to the actions performed by the trainee in order to have a clever distribution of actions between them. If the trainee (an expert trainee) has a disturbing partner, when he will take the screwdriver, the virtual human will steal it to him but he will not use it; it is up to the trainee to adapt his behavior in order to choose other actions to perform or to take the screwdriver back and use it before the virtual human is able to steal it again.

We can extract from this scenario more sequences which illustrate the adaptability of our models. For instance in [7] the trainee is hurt and can not perform any action; as a result his virtual partner interrupts what he was doing in order to help him.

5 Conclusion

The models we proposed in section 3 enable our collaborative virtual training environment to be adaptable in different ways and to answer the needs identified in section 2. Indeed, two levels of adaptation are provided: the first one consits in parametering the application taking into account the demands from the session creator. The second level appears during the execution of the training session where the actors adapt their actions to the current context and also to the previous actions of their partners. All these ways of adaptation enable our virtual environment for training presented in section 4, GVT, to be used in various training situations: individual or collaborative training, procedure training (from military maintenance procedures to furniture assembly procedures) diagnosis training, etc. Moreover, among those different possibilities of adaptation, some of them are new in virtual environments for training. The first one is the parametering of virtual humans, which enables the trainer to give them pedagogical roles (including roles such as disturbing partner that have not been proposed in such environments yet). The second one is the possibility of specifying more than one role for a scenario action. Finally, our action selection process makes possible a dynamic and clever distribution of the actions between the actors, contrary to existing training environments where this distribution is fixed. Future works will consist in studying the pedagogical use of these models. We could for instance provide new opportunities to the trainer (such as to dynamically adjust the collaborative profile of a virtual human) or use the result of the action selection process in order to provide to the trainee more complete explanations about the better action to choose in a specific context.

References

1. Nicolas Mollet and Bruno Arnaldi. Storytelling in virtual reality for training. In Zhigeng Pan, Ruth Aylett, Holger Diener, Xiaogang Jin, Stefan Göbel, and Li Li, editors, *Edutainment*, volume 3942 of *Lecture Notes in Computer Science*, pages 334–347. Springer, 2006.
2. Cédric Buche, Ronan Querrec, P. De Loor, and Pierre Chevaillier. Mascaret: Pedagogical multi-agents system for virtual environment for training. In *CW '03: Proceedings of the 2003 International Conference on Cyberworlds*, pages 423–430, Washington, DC, USA, 2003. IEEE Computer Society.
3. Ronan Querrec, Cédric Buche, E. Maffre, and Pierre Chevaillier. SécuRéVi : virtual environments for fire-fighting training. In Simon Richir, Paul Richard, and Bernard Taravel, editors, *5th virtual reality international conference (VRIC'03)*, pages 169–175, Laval, France, May 2003.
4. Jeff Rickel and W. Lewis Johnson. Virtual humans for team training in virtual reality. In *Proceedings of the Ninth International Conference on Artificial Intelligence in Education*, 1999.
5. Stuart J. Russell and Peter Norvig. *Artificial intelligence: a modern approach.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1995.
6. Yu Zhang, Sen Cao, and Richard A. Volz. Towards a role-based framework in agent teams. In *Workshop of Role-Based Collaboration, The 20th International Conference in Computer Supported Cooperative Work (CSCW'06)*, 2006.
7. Stéphanie Gerbaud and Bruno Arnaldi. Scenario sharing in a collaborative virtual environment for training. In *VRST '08: Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, pages 109–112, Bordeaux, France, 2008. ACM.
8. Chih-Yueh Chou, Tak-Wai Chan, and Chi-Jen Lin. Redefining the learning companion: the past, present, and future of educational agents. *Computers & Education*, 40(3):255–269, April 2003.
9. Nicolas Mollet, Stéphanie Gerbaud, and Bruno Arnaldi. Storm: a generic interaction and behavioral model for 3d objects and humanoids in a virtual environment. In *IPT-EGVE the 13th Eurographics Symposium on Virtual Environments*, pages 95–100, July 2007.
10. Stéphanie Gerbaud, Nicolas Mollet, and Bruno Arnaldi. Virtual environments for training: from individual learning to collaboration with humanoids. In *Edutainment*, pages 116–127, Hong Kong, China, June 2007.
11. Nicolas Mollet, Stéphanie Gerbaud, and Bruno Arnaldi. An operational vr platform for building virtual training environments. In *Edutainment*, pages 140–151, Hong Kong, China, June 2007.
12. Stéphanie Gerbaud, Nicolas Mollet, Franck Ganier, Bruno Arnaldi, and Jacques Tisseau. Gvt: a platform to create virtual environments for procedural training. In *IEEE Virtual Reality Conference. VR '08*, pages 225–232, Reno, NV, USA, March 2008.