A Pedagogical Scenario Language for Virtual Learning Environment based on UML Meta-model

Application to Blood Analysis Instrument

Frédéric Le Corre, Charlotte Hoareau, Franck Ganier, Cédric Buche and Ronan Querrec
UEB/ENIB/CERV, 25 rue Claude Chappe, F-29490 Plouzané, France
{*lecorre, hoareau, ganier, buche, querrec*}*@enib.fr*

Keywords: Intelligent Tutoring System, Virtual Environment, Generic, PedagoGical Scenario, Knowledge Base, Learning, Training.

Abstract: Training to learn the use and maintenance of biomedical devices have various constraints. In order to complete these trainings, we proposed to use virtual reality based on pedagogical scenarios and Intelligent Tutoring Systems (ITS). In this paper, we first established the existing pedagogical scenario models and ITS. Subsequently we presented our proposal of a formal model based on the concept of learning organization by extension of UML in order to describribe some pedagogical scenario and ITS. The use of this model is illustrated by an application of a virtual biomedical analyzer with the aim of learning the technical procedures of the device. Finally, we performed two experiments in order to verify the efficiency of virtual reality training.

1 INTRODUCTION

In the biomedical domain, the traditional training method consists of several learners in a classroom with a live instructor manipulate the device alternately: method one-to-many which is known to be irrelevant (Bloom, 1956; Bloom, 1984) This is the case for the STA-R® instrument (Figure 1) produced by STAGO who sponsored this work. Unfortunately there are several constraints inherent in this training method.

Figure 1: Picture of the biomedical analyzer STA-R®.

First, there is always a potential biohazard while dealing with biological fluids. Second, learners have to use reagents to prepare the blood tests, which increase the cost and variability of the learning sessions. In some cases, the biomedical instrument is transported to the site of the learning session, which could damage it and again can impact the cost. Sometimes, the learning session takes place in the customer's site; therefore the instrument cannot be available for training. Finally the use of diagnostic device is time-consuming for the learner because of inherent time needed during the analytical.

Considering the above, the use of informatics tools for biomedical training sounds interesting. The objective of this training is to manipulate the device according to well-defined procedures. However the successful completion of these procedures requires biology knowledge. A learning strategy like "learning while doing" is relevant : the learner has to manipulate the device. Therefore, we consider that virtual reality is a major contribution in this context. Virtual reality has proven to be efficient to solve those kinds of issues (Mikropoulos and Natsis, 2010; Okutsu et al., 2012). Pedagogical situation using virtual reality are called Virtual Learning Environment (VLE).

However in a lot of professional environment and particularly in biomedical sector, there is a lot of staff turnover. It became then pretty common for employee to not go through the classical learning sessions. The objective of this paper is to propose a solution to implement the pedagogical strategy of the trainer inside the VLE so that the learners could train without having the initial training. Moreover, each individual learner can have its own basal level. So, our VLE have to be able to adapt itself to the learner in order to be really efficient. In medical field, some work were conducted like an application for medical problem-based learning (Suebnukarn and Haddawy, 2007).

This paper is organized in five sections. In section 2 we determined the existing solutions in the context of pedagogical scenarios and ITS. In section 3 we proposed our model based on a definition of a language for a pedagogical scenario (Koper, 2001). The ultimate goal for STAGO was to make an application based on the biomedical devices: VIRTUALANALYZER, presented in section 4 and we also conducted two experiments to verify the efficiency of virtual reality training. Finally, we concluded, in section 5, by listing the characteristics of our model that needed a further improvement and detailed the terms of an incoming experiment.

2 RELATED WORK

We focus on the acquisition of declarative and procedural knowledge on complex systems (many types of heterogeneous elements) in virtual reality. In this field, the most popular and representative tutor is STEVE (Rickel and Johnson, 1998), an animated pedagogical agent. STEVE belongs to the ITS field. Classically, ITS are structured on four models : domain, pedagogical, learner and interface (Wenger, 1987). One of the classic problems in ITS is to provide a generic language to describe domain knowledge. In our case, this knowledge is complex and from a high level of expertise. We have already proposed a meta-model MASCARET (based on SYSML, an UML extension) in order to acquire the system specifications to learn, directly from the conception(Querrec et al., 2013). This method allows us to generate the domain model and execute it in the virtual environment. It is not conceivable to rewrite this knowledge in the domain model, so it will be directly imported. Similarly, these specifications will directly lead to the generation of the virtual environment. A second problem in ITS is the link between the rules governing the pedagogical behavior of the tutor and the pedagogical course of training. Under the training procedures on complex systems, work has already proposed learning scenarios templates (explanation of the system and subsystem, explanation and organization of the procedures, repetition more or less guided). In our more specific context, learning scenarios have already been defined, it is therefore necessary to import it. In this paper, we propose a model to explain this pedagogical scenario in the pedagogical model related to the domain knowledge.

2.1 ITS

Our work belongs to ITS field. Classically ITS are structured around four modules:

- Domain : knowledge on the job to learn
- Pedagogical : knowledge on the pedagogical strategies
- Learner : knowledge representation of the learner. Very often, this knowledge is a subset of the domain knowledge where the learner had access.
- Interface : knowledge representation that the tutor may have on the actions that the learner achieves and the actions that the tutor can make in return.

This paper focuses exclusively on the pedagogical and domain models. Work on these models aim to make them generic, adaptative and individual (hence the link with the learner model). In the ITS domain, many research works were made during the past few years: some projects aimed at developing the generic part of ITS (Sanchez and Imbert, 2007; Shi and Lu, 2006; Sorensen and Ramachandran, 2007). Some projects aim at individualizing the simulation for each learner. This could be done for example through emotional agents (Ailiya et al., 2010) or by Hollnagel classification (El-Kechai and Desprès, 2007). At lower scale, some have highlighted the adaptability characteristic of ITS (Dos Santos and Osorio, 2004).

Typically, these models are defined from cognitive architectures like STEVE which is based on SOAR[1], or CTAT (Koedinger and Heffernan, 2003) which is based on ACT-R (Adaptive Control of Thought-Rational). Knowledge expressed in this system is defined as a set of rules like in ANDES (Vanlehn et al., 2005). In this context we have already proposed PEGASE which is based on the four classical models (domain, learner, pedagogical, inteface) (Woolf, 1992). PEGASE also contains an error model and a definition of the pedagogical model (usable independently of the exercise to do) in order to generalize it. One last important part of PEGASE is its adaptability, allowed by the auto-modification of the pedagogical model. PEGASE proposes a pedagogical model based on a hierarchical classifier system. This system organizes knowledge while taking the abstraction of the data involved into account. It structures knowledge according to three levels, from rules based on abstract knowledge of educational methods (the pedagogical approach), to the rules based on concrete knowledge

---

[1] http://sitemaker.umich.edu/soar/home

of virtual reality (pedagogical techniques), via an intermediary level (pedagogical attributes).

A major difficulty in the design of these models is writing these rules. For this there are three technical :

- Authoring tools like in CTAT (Koedinger et al., 2004), but can not be as generic as desired due to the proposed interfaces.
- Model-tracing which observe the actions of experts to build the domain model.
- Meta-model definition like the KBT-MM model of Murray (Murray et al., 2003) which has however a too high level of abstraction.

From this meta-model, Murray proposed a specialization called EON. Knowledge is then defined using ontologies. Many researches focused on some languages to describe the ontologies, like Resource Description Framework (RDF) or Web Ontology Language (OWL). The major problem of theses languages is that they do not allow to describe the dynamics of the situation when we precisely needed it to teach how works the instrument. We are interested in this approach but at an intermediate level compared to KBT-MM. Indeed, we propose a model in the context of learning procedures (handling, maintenance and diagnostic) on complex systems (industrial). Moreover, these systems are so complex that it is not possible to rewrite the system specifications (for the knowledge base or for the conception of the virtual environment).

MASCARET is an extension of the meta-model SYSML. It enables to describe the system structure by blocks, attributes and compositions from SYSML. The reactive behaviors of the system elements are described by state machines and the domain procedure are described by activities. MASCARET is a kind of SYSML interpreter for virtual reality and provides an operational semantic for every element of the meta-model. This enables to make the knowledge explicit for the agents during the simulation and to automatically execute the system entities behaviors. As we said in section 1, on this kind of system, work on the training structuring has already been done and it is therefore appropriate to incorporate the pedagogical scenario knowledge in the pedagogical model.

2.2 Pedagogical Scenario

Prior to 2000, teaching scenarios were based primarily on documentary approaches like *Learning Object Metadata*[2] (LOM) or *Sharable Content Object Reference Model*[3] (SCORM). From the 2000s Koper (Koper, 2001) initiated the use of educational modeling languages like *Educational Modeling Language* (EML). The main feature of EML is, that unlike documentary approaches, EML focus on the description of activities and not on educational resources.

Afterwards, Koper (Koper et al., 2003) contributed to the IMS-Learning Design (IMS-LD) standard, based on EML. IMS-LD focused on the concept of learning unit as a base element of the description of the learning process. Indeed, in IMS-LD a scenario is considered as a series of educational activities. Each of these activities is described by a text or a set of documents explaining the purpose of the activity, the task to achieve, the instructions to be followed, etc.

However, IMS-LD has its own limitation, like this review of Ferraris (Ferraris et al., 2005) based on the lack of expressivity regarding the description of the interactions between users in collaborative tasks. These models from VLE do not fully meets our requirements, so we had to look more specifically the existing models in virtual reality for training.

In the literature we could identify several virtual reality models for training. The FORMID (FORMation Interactive Distance) (Guéraud et al., 2004) project focus on pedagogical scenario activities in which learners interact with an interactive pedagogical object. Another proposal is called GVT (Generic Virtual Training) and is intended for procedure learning. The disadvantages of FORMID and GVT are that they are not generic, they are not reusable and there is no distinction between the activity scenario for the procedures of the environment and the pedagogical scenario.

Marion (Marion et al., 2009) proposed a model of a pedagogical scenario called POSEIDON, which aims to be directly reusable in different environments. POSEIDON covers various points like: educational objectives, prerequisites, education activities, education organizations and environments. We have decided to base our own pedagogical scenario model on POSEIDON because of this generic side. However the negative point of POSEIDON is its lack of link with an ITS. Our goal is to create this link so that the ITS can use the pedagogical scenario knowledge to reason.

3 MODEL

Classically, a pedagogical scenario is composed of pedagogical objectives and prerequisites, pedagogical organization (set of roles), pedagogical activities and an environment (Koper, 2001). Following that definition, we considered that the trainer activity is a domain activity just like any other. Thereby we could

---

[2] http://www.lom-fr.fr

[3] http://www.scorm.com

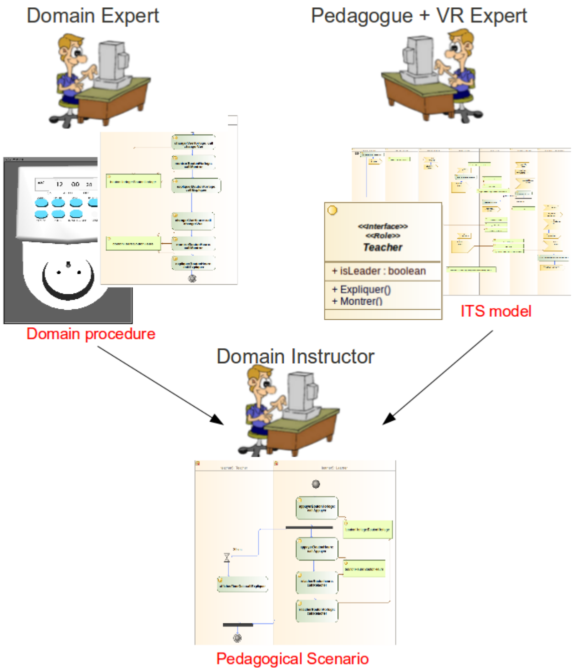

Figure 2: Workflow for application conception.

propose a formal model of the concept of a pedagogical scenario by extended UML. A pedagogical organization is considered like collaboration composed of Roles. A role is the UML concept of Interface. This means that a role lists a set of services without actually providing an implementation. The agent (artificial or human) who will play this role will propose its own implementation. Within this pedagogical organization, the role takes part to the pedagogical scenario as an activity arranging the pedagogical actions which could modify the virtual environment (composed of the system or the pedagogical resources). These concepts enable to create a pedagogical scenario and its development is now presented.

3.1 Workflow

In order to achieve a complete virtual reality training application (application, ITS, scenario), several people of various fields must participate. The picture 2 summarizes all the contributors described in this section.

First, an educationalist instructor assisted by a virtual reality expert defines the ITS model. Indeed, his instructor qualities enable him to define the best pedagogical strategies for the ITS, and the virtual reality expert knowledge allows to implement the pedagogical actions specific to virtual reality (set an entity in transparence, switch the point of view,...) The knowledge is and must be generic. They are described and implemented only one time and are reused for all pedagogical scenarios in all kind of fields. Second, another contributor must participate to the application development: the application domain expert who describes the entire domain model. The expert knowledge is centered on the field on which the application is based on. Finally, the domain teacher imports the ITS model, described by the educationalist instructor, and the domain model, described by the domain expert, in order to produce some pedagogical scenarios. This three step development enables to separate the pedagogical scenario conception and the system conception. Thereby, our pedagogical scenario model is generic and can be applied to other instruments and in other fields.

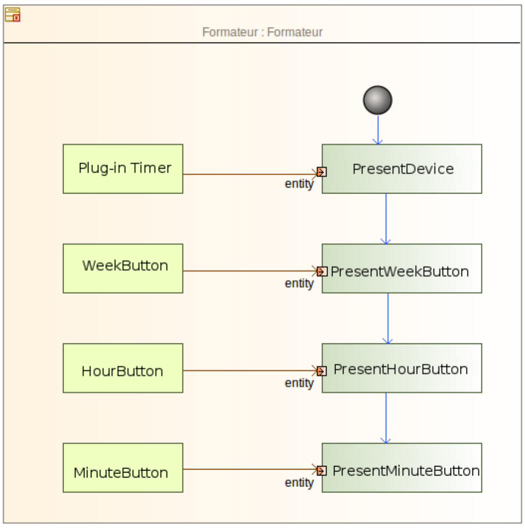

Figure 3: UML model of pedagogical scenario for system knowledge acquisition.

3.2 Pedagogical Scenario and ITS

Thus, the trainer could describe some pedagogical scenarios to learn from the system. This can be of two types: for knowledge acquisition or for procedure acquisition (use, maintenance and diagnostic). Based on an example of a plug-in timer application, we can present a pedagogical scenario for knowledge acquisition (Figure 3) including a trainer role who is responsible for presenting the plug-in timer and each of its buttons (Figure 4).

The “present” operation could be defined by the educationalist instructor in this manner:

- set all the entities in transparence except for the one to present
- change the point of view on the entity to present

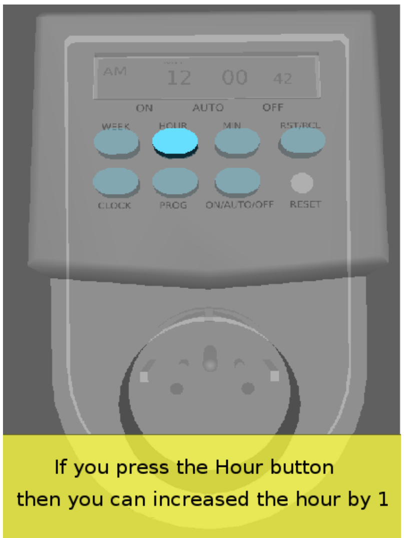

Figure 4: Picture of a plug-in timer in the virtual environment with the “present“ operation

- display the annotation from the domain model on the entity to present

So, the trainer should rely on this pedagogy and the domain model in order to create his pedagogical scenario. We could also describe some pedagogical scenarios for procedure acquisition: for example the teacher role presents the procedure and the learner role has to carry out the domain procedure.

Moreover, we could describe ITS behaviors in the same manner as the pedagogical scenarios described previously. This kind of modelisation ensures a high modularity of the system, which enables to create ITS behaviors more or less complex. For example, in picture 5, the trainer described the learner role which had to carry out a procedure of setting time of the plug-in timer. The teacher role monitors the environment, highlight the learner errors and perform an “undo“ on the current action (back to the previous state). The advantage is that the trainer can easily remove or add some actions, making our system very modular.

In the same way, we could create another application: VIRTUALANALYZER which is applied to our biomedical training problematic.

4 APPLICATION

The proposed model has been applied to the problem of training presented in section 1 in order to realize a virtual reality application for STAGO.

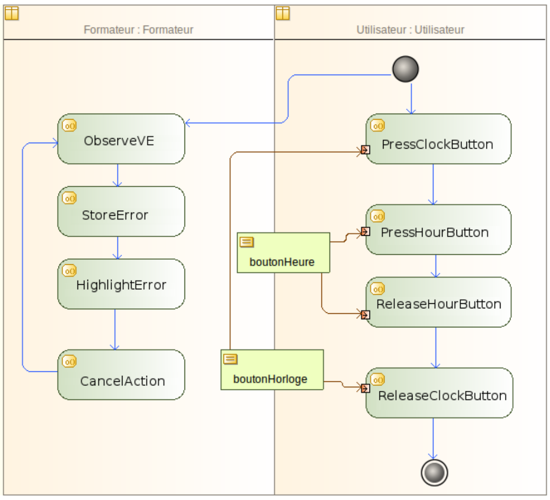

Figure 5: UML model of a ITS behavior

4.1 Virtual Environment

While looking for replicating the real STA-R$^{\circledR}$ in a virtual environment, we made a virtual reality application with the instrument in a 3D environment and the control software modeled on ANDROID (Figure 6).

Figure 6: Virtual STA-R$^{\circledR}$ application.

The environment includes a pre-operative blood work procedure. This procedure is composed of 125 basic actions and is coupled with a simple ITS behav-

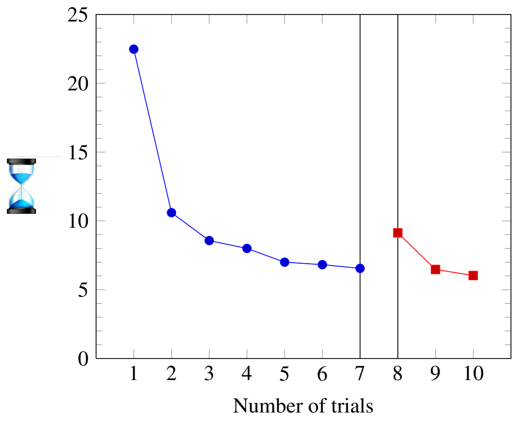

Figure 7: Learning curve: total time.

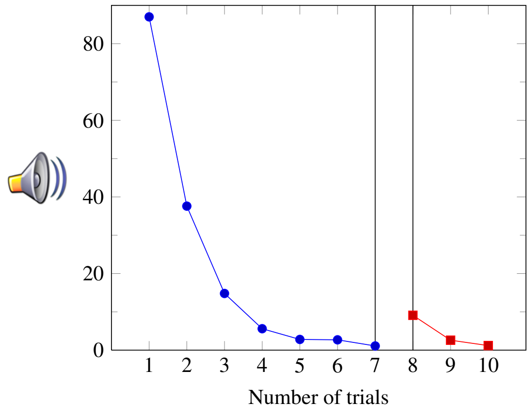

Figure 8: Learning curve: instructions.

ior: vocal assistance and blocking wrong actions. In order to verify the efficiency of virtual reality training, we performed two experiments. The first one aimed at verifying the procedure learning with 12 computer science students during two days. The second experiments goal was to verify the knowledge transfer on a real STA-R® with 58 biology students during ten days.

4.2 Experiments

4.2.1 Learning

First, the learners had to perform seven times the procedure described previously. After a week time, the learners came back to complete the second session of the experiment: they had to do three times the same procedure. In order to evaluate the learning procedure we collected behavioral data: total time of the procedure achievement, number of audio instructions consultation, number of wrong actions (Figure 7 and 8).

The total time of the procedure achievement and the number of audio instructions consultation decreased over repetitions. After seven days, the learners partially consulted the instructions only during the first try, and then their performance became similar to the latest ones of the first session. This is a typical learning curve. Thereby, this study confirmed the usability of VIRTUALANALYZER for learning procedure. However, a large number of try would be necessary to obtain a stabilization of performance and a lower number on instruction consultation at the 8th try, certifying the perfect acquisition of the procedure and its storage in long-term memory. Learning by using a virtual environment is favorable only if the skills acquired through this device can be used in a real situation. We attempted to verify this assumption in the following experiment.

Table 1: Results of the knowledge transfer experiment.

	Traditional	Virtual	Control
Achievement	100%	100	72%
SOS	0.2	0.7	3.1
Interventions	1.4	2.3	3.9
Consultations	9.5	1.9	48.7
Total time	19'48"	30'02"	39'09"

4.2.2 Knowledge Transfer

This experiment aimed to verify the knowledge transfer on a real STA-R® with 58 biology students. The students were divided into three groups: control, traditional teaching and virtual teaching. Each of these groups had a theoretical training on STAGO, hemostasis, biological tests and the instrument during one hour. The traditional group had a classical training provided by a trainer from STAGO (six learners for one instrument) and the virtual group had a training with VIRTUALANALYZER (one computer for each learner): each group trained for two hours while the control group had no training. Thereafter, each learner had to individually perform the entire procedure on the real STA-R®.

In order to evaluate the transfer procedure, we collected some data: procedure achievement, number of paper document consultations, number of technician interventions, number of technician call for help and total time (Table 1).

This experiment showed no significant differences between the data of the traditional method and the virtual method, except for the total time. This difference can be explained by several ways: our android interface is very different from the real interface of the STA-R® and the learners have no notion of time of the working instrument. We hope to reduce this time to a value close to the traditional one, by modifying how to provide some information in our application. The knowledge acquired with our virtual environment is transferable in a real STA-R®.

5 CONCLUSION AND FUTURE WORKS

STAGO wanted to use virtual reality and virtual environments for training for their biomedical diagnostic devices. Thereby, we proposed a training application for a STAGO instrument called VIRTUALANALYZER. This enabled us to verify the quality of the learning during a training based on this application and also to check if the knowledge is transferable in a real environment. Therefore, our work met the objectives defined by STAGO. However, the conducted experiment did not enable to verify the contribution of our model on the learning. Indeed, the experiments are based on the application rather than a complex ITS behavior. This is why we would like to evaluate, in the future, the contribution of the ITS with a complex behavior (like the PEGASE one) on the application VIRTUALANALYZER. We also pointed out that the language ergonomic for describing the pedagogical scenarios is not intuitive for the trainer. Although the interface is graphical and formalized, the fact that it relies on UML concepts does not help the ergonomic and does not facilitate its use by domain trainers. Finally, the long-term goal for STAGO would be to deploy this work to their whole range of instruments.

REFERENCES

Ailiya, Shen, Z., and Miao, C. (2010). An emotional agent in virtual learning environment. In *Transactions on Edutainment IV, LNCS 6250*, pages 22–33.

Bloom, B. (1956). Taxonomy of educational objectives. In *Educational Researcher*, volume 1.

Bloom, B. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. In *Educational Researcher*, volume 13, pages 3–16.

Dos Santos, C. T. and Osorio, F. S. (2004). Integrating intelligent agents, user models, and automatic content categorization in a virtual environment. In *ITS 2004, LNCS 3220*, pages 128–139.

El-Kechai, N. and Desprès, C. (2007). Proposing the underlying causes that lead to the trainee's erroneous actions to the trainer. In *EC-TEL : European Conference on Technology Enhanced Learning*, volume 4753, pages 41–55.

Ferraris, C., Lejeune, A., Vignollet, L., and David, J.-P. (2005). Modélisation de scénarios pédagogiques collaboratifs. In *Conférence EIAH*.

Guéraud, V., Adam, J.-M., Pernin, J.-P., Calvary, G., and David, J.-P. (2004). L'exploitation d'objets pédagogiques interactifs à distance : le projet FORMID. *STICEF*, 11:103–163.

Koedinger, K. R., Aleven, V., Heffernan, N., McLaren, B., and Hockenberry, M. (2004). Opening the door to non-programmers: Authoring intelligent tutor behavior by demonstration. *INTELLIGENT TUTORING SYSTEMS Lecture Notes in Computer Science*, Volume 3220/2004:7–10.

Koedinger, K. R. and Heffernan, N. (2003). Toward a rapid development environment for cognitive tutors. In *in Proceedigns of the International Conference on Artificial Intelligence in Education*, pages 455–457. IOS Press.

Koper, R. (2001). Modeling units of study from a pedagogical perspective. In *Educational Technology Expertise Centre, Open University of the Netherlands*.

Koper, R., Olivier, B., and Anderson, T. (2003). Ims learning design information model. In *IMS Global Learning Consortium*.

Marion, N., Querrec, R., and Chevaillier, P. (2009). Integrating knowledge from virtual reality environments to learning scenario models. a meta-modeling approach. In *International conference of computer supported education*, pages 254–259.

Mikropoulos, T. and Natsis, A. (2010). Educational virtual environments: A ten-year review of empirical research (1999-2009). In *Computer and Education*, volume 56, pages 769–780. Elsevier.

Murray, T., Blessing, S., and Ainsworth, S. (2003). *Authoring Tools for Advanced Technology Learning Environments: Towards cost-effective adaptive, interactive and intelligent educational software*. Kluwer Academic Publishers.

Okutsu, M., DeLaurentis, D., Brophy, S., and Lambert, J. (2012). Teaching an aerospace en- gineering design course via virtual worlds: A comparative assessment of learning outcomes. In *Computer and Education*, volume 60, pages 288–298. Elsevier.

Querrec, R., Vallejo, P., and Buche, C. (2013). MASCARET: creating virtual learning environments from system modelling. In *Engineering Reality of Virtual Reality (ERVR'13)*, volume 8649, pages 8649–04, San Francisco. SPIE.

Rickel, J. and Johnson, W. L. (1998). Steve: A pedagogical agent for virtual reality. In *Second International Conference on Autonomous Agent*, pages 332–333.

Sanchez, L. and Imbert, R. (2007). An agent-based adaptable and configurable tutoring module for intelligent virtual environments for training. In *Edutainment 2007, LNCS 4469*, pages 499–510.

Shi, R. and Lu, P. (2006). A multi-criteria programming model for intelligent tutoring planning. In *KES 2006, Part I, LNAI 4251*, pages 780–787.

Sorensen, B. and Ramachandran, S. (2007). Simulation-based automated intelligent tutoring. In *Human Interface, Part II, HCII 2007, LNCS 4558*, pages 466–474.

Suebnukarn, S. and Haddawy, P. (2007). Comet: A collaborative tutoring system for medical problem-based learning. *IEEE Intelligent Systems*, 22(4):70–77.

Vanlehn, K., Lynch, C., Schulze, K., Shapiro, J. A., Shelby, R., Taylor, L., Treacy, D., Weinstein, A., and Wintersgill, M. (2005). The andes physics tutoring system: Lessons learned. *Int. J. Artif. Intell. Ed.*, 15(3):147–204.

Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems*. Morgan Kaufmann, Los Altos, California.

Woolf, B. (1992). Building knowledge based tutors. In *Computer Assisted Learning : the 4th International Conferences (ICCAL'92)*, pages 46–60.