



# A methodology for the design of pedagogically adaptable learning environments

Julien Saunier <sup>1</sup>, Mukesh Barange <sup>1</sup>, Bernard Blandin <sup>2</sup>, Ronan Querrec <sup>3</sup>

<sup>1</sup> Normandie Université - LITIS, 685 Avenue de l'université, 76800 Saint-Etienne du Rouvray

<sup>2</sup> CESI-LINEACT, 169 Rue Georges Auric, 34070 Montpellier

<sup>3</sup> ENIB, Lab-STICC, Technopole Brest-Iroise, 29238 Brest

**Abstract** - In the last decades, the industry has profoundly integrated the use of digital resources in their production process. However, these assets are rarely re-used for the training of the users, operators and technicians that have to interact with these objects. Furthermore, although training and learning environments are classical applications of virtual reality, the design of these environments is generally ad hoc, i.e. dedicated to specific operations on specific objects, hence requiring the intervention of programmers whenever a modification of the pedagogical scenario is required. In this article, we propose a methodology to design adaptable virtual environments, by separating the role of the different protagonists that play a part in the creation of learning environments. In particular, its goal is to allow the teachers to implement different scenarios according to the level of the trainees and to the pedagogical objectives without the intervention of computer scientists. An example of adaptable wind turbine environment is shown, with three different learning situations: simulator, safety training and preventive maintenance training.

**Index Terms** - Virtual reality; pedagogical scenario; UML meta-model; learning environment.

## I. INTRODUCTION

This work takes place in the context of the EAST (Scientific and technical learning environments) project, which aims at stimulating the interest of young people for science through virtual reality environments, based on industrial assets.

Although there is an increasing production and use of digital resources in the industry, these resources are rarely used in the context of human learning and training. A major technical issue to use industrial assets for training purposes is the lack of generic tools to integrate pedagogical contents with these assets. The EAST project aims at proposing a methodology and tools to facilitate the design of activity training in virtual environments (VEs). This methodology is based on the use of 3D models of industrial equipment or objects on which or through which a pedagogical activity is performed,

and on an analysis of the activity targeted by the learning action.

A major challenge to achieve this objective is the integration of (1) the semantics of the VE components, (2) their behavior, (3) how humans interact with them, and (4) pedagogical assistance provided through the scenario or continuous help. We propose to perform this integration through the use of a unified abstract description of all these elements that compose the learning environment.

Although the advantages of using virtual reality (VR) technologies to develop professional skills in safe and low-cost settings have already been demonstrated, their design (and the modifications of existing VR environments) necessitates computer scientists, even for modeling and implementing expert actions or pedagogical actions. However, computer scientists introduce their own biases in their modeling and implementation choices that impact the final learning environment.

In the following, we first present the state-of-the-art related to the design and implementation of pedagogical scenarios in section II. Then, in section III, we present our methodology that separate the responsibilities of the actors playing a role in the design of the environment. We show in section IV how it is applied in the context of a wind turbine. Finally, we conclude and discuss some open issues for virtual reality environments for human learning.

## II. MODELING OF EDUCATIONAL

### SITUATIONS IN VIRTUAL ENVIRONMENTS

Technical training in industrial systems is one of the privileged areas of application of VR (Mikropoulos and Natsis 2011). In this section, we present the related works of this domain.

### II.1 VIRTUAL REALITY FOR TRAINING

The advantage of VR in this area and the impact of immersion modes (Desktop, Helmet VR etc.) and interactions (mouse, Motion Capture etc.) in learning has already been evaluated (Boud, Baber, and Steiner 2000; Gavish et al. 2011; Stevens, Kincaid, and others 2015). Beyond the relevance of VR for training, one of the disadvantages of VE design approaches, as raised above, is the inevitable intervention of computer scientists during all phases of implementation. This requires them to understand the relevant

---

E-mail: julien.saunier@insa-rouen.fr

profession, which can lead to misinterpretations and approximations, and to design the pedagogical scenarios, which can introduce implicitly their own representation of pedagogy and of the learning process. To overcome these problems, outsourcing these expertises is a proposed solution. In this way, the representation of the activity and of how it should be taught becomes external data to the computer program, while this knowledge becomes explicit when running the VR simulator. Such systems are named informed or intelligent VEs.

Informed VEs (Kallmann and Thalmann 1999) advocate the use of artificial intelligence techniques to represent explicitly the task knowledge in the VE. Several models have been proposed, such as Smart Objects (Lugrin and Cavazza 2006) which aim to integrate the knowledge necessary to the interaction within the virtual environment, or STORM (Mollet and Arnaldi 2006) that allows to define interaction not only between an agent and an object but also between multiple objects. Moreover, the collaborative version of STORM (Saraos Luna, Gouranton, and Arnaldi 2012) allows to define a collaborative manipulation of one object by various users. Concerning user activities, several models to express the human activities to be performed in virtual environments have been proposed. The LORA scenario language (Language for Object Relation Application) equates the virtual environment with the prescribed procedure to be performed by the user (Mollet and Arnaldi 2006). ACTIVITY-DL (Barot et al. 2013) formally called HAWAII-DL (Edward et al. 2008) is used in ergonomics to describe human activities.

The disadvantage of these models or languages is that they cover only a part of the representation of the system and require to assemble several of them in order to cover the whole pedagogical scenario. Generally, the design of these languages has been guided more by the possibility of being automatically interpreted by a computer, and not by its own power of expression. In our case what interests us is not only a language that is directly executable by a computer (especially the dynamic aspect of the system), but also that this language allows to describe the different components (structural, static and dynamic) of a system by the expert, to avoid the biases that may be introduced by the computer scientists. This is one major issue that our project intends to tackle.

## II.2 PEDAGOGICAL SCENARIO MODELING

Current methods for the modeling of pedagogical scenarios can be divided into two groups: 1) design methods for the learning activity; 2) operationalization methods of the learning activity.

Among the methods of design of the learning scenarios, we can cite LDL (Martel et al. 2006) and Isis (Sanagustín, Emin, and Hernández-Leo 2012). LDL (Learning Design Language) focuses on content management, enabling to describe and design collaborative learning situations. Isis (Intentions, Strategies, and interactional Situations) is rooted in previous work on active learning situations, and proposes a specific identification of the intentional, strategic, tactical and operational dimensions of a learning scenario. The operationalization methods of the learning activity allow the translation of elements defining the learning activity

(action, actors, and resources) into machine language (usually XML). Educational Modeling Language (EML) (Koper 2001), its standardized version IMS-LD (Koper and Tattersall 2005) and a prototype of visual EML (Retbi, Idrissi, and Bennani 2013) describe the content and process within a unit of study from a pedagogical perspective in order to support reuse and interoperability. Considering another perspective, the MASCARET-CHRYSAOR model (Le Corre et al. 2012), which extends the meta-model based MASCARET approach (Chevaillier et al. 2011), is tailored for the operationalization of learning environments of types micro-world simulator or serious games, because they allow describing activity scenarios taking into account the interactions with the virtual environment and with the objects within it.

These two groups are complementary methods. Although Isis is operationalized via IMS-LD, it has been noted in (Koper and Tattersall 2005) that IMS-LD is suitable for the development of distance learning scenarios, which leads to pedagogy oriented intelligent tutoring system (ITS). This approach is therefore not relevant to ITSs that supports the learning environments of micro-worlds simulators or serious games types, although they offer the greatest opportunities for the creation of new learning modalities. In the second family of approaches, the composite MASCARET-CHRYSAOR approach allows this kind of pedagogical settings. However, these approaches have not formalized so far the links with design methods for learning activities, and they do not make their models accessible to teachers and trainers. Thus, our methodology proposes to combine Isis approach for designing learning activities and its operationalization through the MASCARET approach.

## III. DESIGN METHODOLOGY FOR PEDAGOGICAL SCENARIOS IN VIRTUAL REALITY

The main principle of our methodology is to externalize both the expert knowledge on the application domain and the pedagogical elements from the virtual environment. The virtual environment itself only contains the 3D components of the scene. All the other information is considered as data, produced by the protagonists of the learning environment design process.

In order to achieve this, we propose to define a unified modeling language that enables to represent the two aspects of the learning situation: its contents (expert gestures or procedures) and its presentation (pedagogical scenario). In this way, the pedagogical scenario is a particular expert procedure – from the pedagogical field – that is intertwined with application (expert) procedures. This enables both procedures to share a common modeling language, to be instantiated from the same meta-model, and to be integrated in a common sequence that is the pedagogical scenario.

We have chosen UML<sup>a</sup> as our modeling language because its formalization enables it to be automatically interpreted by the virtual environment player (in our case MASCARET) and its graphical representation can be manipulated by non-

<sup>a</sup><http://www.uml.org/>

computer scientists. Furthermore, the curriculum of french teachers contains the creation and use of SysML diagrams, which is an extension of the UML specification.

In our methodology, the pedagogical scenario formal model is also an UML extension. The pedagogical organization of a scenario is a collaboration made up of roles. Roles are UML interfaces, which means that they provide a set of services without implementing them. The agent playing a role supplies the actual instantiation of those services. By agent, we mean either an artificial agent or a human protagonist. Each role participates to the pedagogical scenario, which is an activity putting together pedagogical actions that modify the state of the virtual environment (the system and the pedagogical resources).

In figure 1, we show the four main roles in our design methodology:

- The **education specialist** defines the pedagogical actions that can be used to guide or correct the trainee in the virtual environment settings, as well as the pedagogical action forms (typical sequences of actions, reactions and interactions with the objects of the system). These actions are independent from the application domain, from the technological environment and from the pedagogical strategies. However, they depend on the type of learning environment, for example interactive simulations. The output from the education specialist is a library containing a set of pedagogical actions described as UML objects.
- The **job expert**, who knows the activity which has to be learned, formalizes the sequence of actions and the interactions with the objects of the environment. He also describes good practices and procedures that have to be learnt and the different behaviors (proactive or reactive) of the objects. This description is independent from the execution platform and from the 3D objects that represent the objects. The output from the job expert is thus the set of actions, procedures and dynamics that can be done to, with or by the objects and agents of the learning environment. These elements are also UML objects.
- The **designer** creates the virtual environment, *i.e.* the objects and their behavior (based on the job expert specifications) based on heterogeneous sources such as industrial assets, pictures and behavior libraries. He also enables the instantiation of the learning environment by interfacing the geometries and scripts with the pedagogical roles of the different actors. The output from the designer is therefore a mapping from the expert knowledge to a 3D environment, without *a priori* on the pedagogical scenarios to be instantiated.
- The **teacher** (or trainer) defines pedagogical scenarios (the sequence of situations in which the trainee acts in the environment) and the pedagogical assistance provided by the system in real time. To define these scenarios, the teacher manipulates:
  1. the environment and the objects it contains (created by the designer),
  2. the potential actions of the learner on the objects and the good practices (defined by the job expert),
  3. the models, the pedagogical action forms and generic pedagogical actions (defined by the education specialist).

3. the models, the pedagogical action forms and generic pedagogical actions (defined by the education specialist).

The output from the teacher is UML diagrams composing these different elements in one or more consistent pedagogical scenarios to be used by trainees.

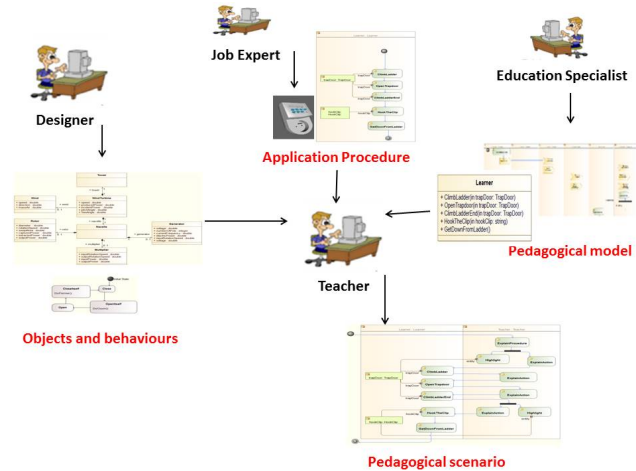


Figure 1: Pedagogical scenario design workflow

Although these roles are separate, some of them may be played by one person, e.g., the job expert and the teacher / trainer.

The final model produced by the teacher, after the intervention of these different roles, is then exported from the UML editor to be interpreted by the MASCARET module in the VR platform. In this way, creating and playing a new scenario does not require the intervention of a computer scientist. The models semantics are detailed in (Le Corre et al. 2012).

As we mentioned before, the data produced by the job expert and the applicative teacher are dependent on the application domain. However, it is not the case of the data produced by the education specialist: this pedagogical library can then be re-used in different settings. The use of a common language based on UML model for both job activity and pedagogical activity description also enables to extend these libraries *a posteriori*. The library developed until now contains the following action types:

- Pedagogical actions on the VE: *highlight an object, play an animation*
- Pedagogical actions on user interactions: *change the point of view, block a position*
- Pedagogical actions on the structure of the system: *describe the structure or an element of this structure; display an entity's documentation*
- Pedagogical actions on the system dynamics: *explain the objectives of a procedure, explain an action*
- Pedagogical actions on the scenario: *display a pedagogical resource such as a video or text document, explain the objective of the current scenario*

Indeed, these actions have to be instantiated by the designer in the platform chosen for the specific application. The MASCARET interpreter<sup>b</sup> is available for the Unity3d platform and already contains these pedagogical actions.

#### IV. EAST PROJECT: APPLICATION TO A WIND TURBINE

The first step of our methodology is achieved by the job expert; it is the definition of the system structure and of the procedures and actions that can be applied to it by technicians.

**Applicative model.** Figure 2 (A) illustrates a part of the UML model defining the structure of the windmill. The system consists of a series of equipment which collects and transforms the energy from the wind into electrical power. The system is directly described in the model, which enables to easily modify (outside of the VE) its behavior, for example to define several wind turbine models using the SysML<sup>c</sup> formalism. Any UML editor may be used to achieve this task.

The classes and attributes are documented in order to generate pedagogical actions (such as *explaining the system*). For example, figure 2 (B) represents the description of the voltage attribute of the class Generator.

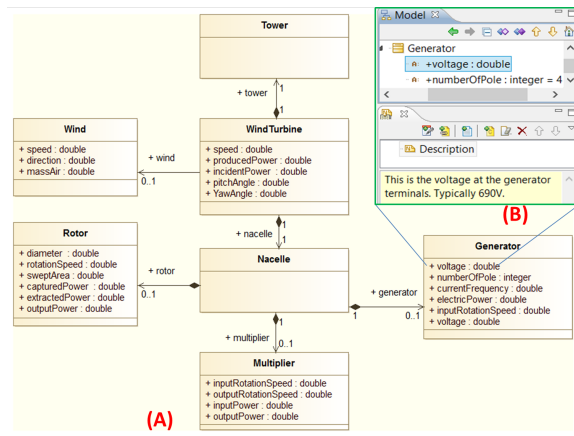


Figure 2: Windmill structure

The designer then implements a view of these objects to be played in the virtual environment; using a 3D modeller (Fig. 3).

Concerning the procedures defined by the job experts, several learning activities have been implemented within the same wind turbine environment: system physics presentation, safety training, and maintenance training. The UML description enables to describe complex activities through loops, sequences, parallelism, events, etc. The hierarchical structure is used and interpreted by MASCARET as a knowledge base to reason on the procedure and track the activity of the trainee.

<sup>b</sup><https://github.com/querrec/Mascaret>

<sup>c</sup><http://www.omgsysml.org/>

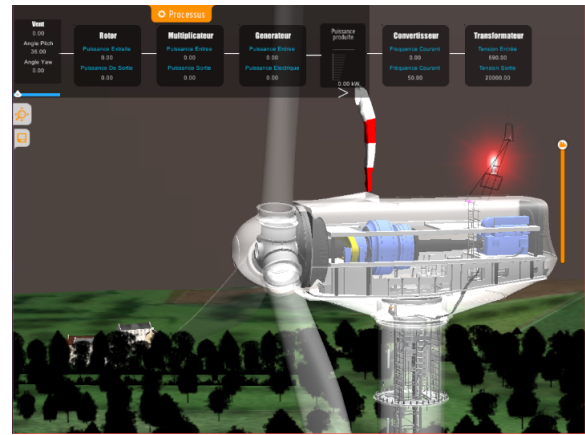


Figure 3: Virtual environment representation

**First scenario: Simulation.** The objective of this scenario is to present the transformation of wind energy into electricity, depending on variables such as wind force, rotor orientation, and internal multiplier and generator characteristics.

This scenario uses pedagogical actions to present the structure of the system, highlight and put in transparency elements. It also allows the trainee to modify the external variables (wind, orientation) to study their impact on the system output. Fig. 4 shows the different variables that are automatically calculated when the trainee changes the value of properties of the system.



Figure 4: Changing properties of the system

**Second scenario: Safety.** The safety scenario is designed to teach trainees the safety measures to intervene in a wind turbine, such as ladder climbing, opening / closing trapdoors, and hooking to safety rings. The interaction mode chosen is a metaphor using either a keyboard (through lateralized keys) or a joystick (through triggers) for VR headsets. Fig. 5 is an extract from the scenario with two parts (or roles) played in parallel: the trainee and the virtual teacher.

The principle is to repeat the procedure until it is acquired by the trainee (transferred to its long-term memory). How-



ever, pedagogical assistances are adapted according to the number of sessions of the trainee.

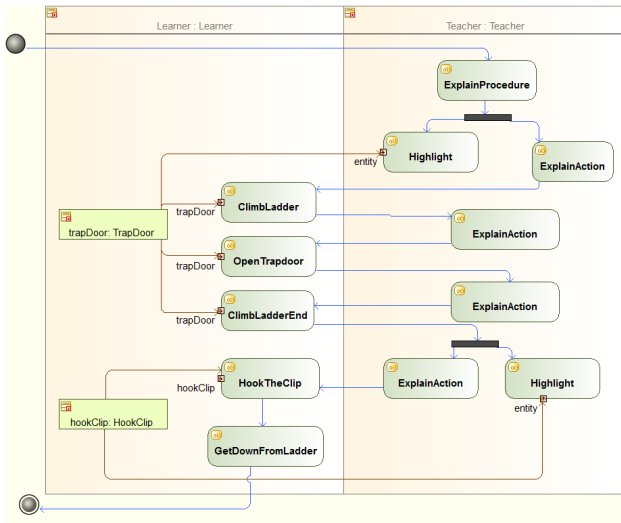


Figure 5: Extract from the safety procedure

The pedagogical scenario shown in figure 5 is dedicated to a first session with the virtual environment, and therefore contains a number of explanations on the environment and actions, because the learner does not know *a priori* all the actions and resources to be used.

These pedagogical elements are not necessary in further scenarios when the trainee is familiar with the virtual environment. In this case, the teacher only has to remove the corresponding steps of the scenario, *i.e.* the actions contained in the "teacher" role of the UML diagram, to remove a part or all of the explanations and generate the new scenario. In the same way, if the teacher wants to focus the training on specific parts of the wind turbine in the context of special sessions, he can modify the procedure to add or remove specific parts of the scenario, only by the manipulation of graphical boxes.

The first action of the pedagogical scenario shown in figure 5 explains the objective of the procedure so that the learner knows that his objective is to attain the top of the wind turbine. During his ascent, he has to keep a stable balance on the ladder and always keep his lifeline attached to a runner or to a safety ring. This explanation exploits the UML description of the scenario, and can thus be modified by the teacher without changing the implementation of the VE.

Finally, the part of the scenario shown in figure 5 has two more pedagogical actions, executed in parallel; the first action enables to explain the next action to be performed, (climbing to the trapdoor), while the second enables to highlight the trapdoor (changing its color to a salient red as shown in figure 6). These pedagogical actions are generic and can be used on any element of the environment.

**Third scenario: Maintenance** The aim of the third scenario is to teach a maintenance procedure. The particular case chosen is the verification of the multipliers gears wear



Figure 6: Extract from the safety procedure

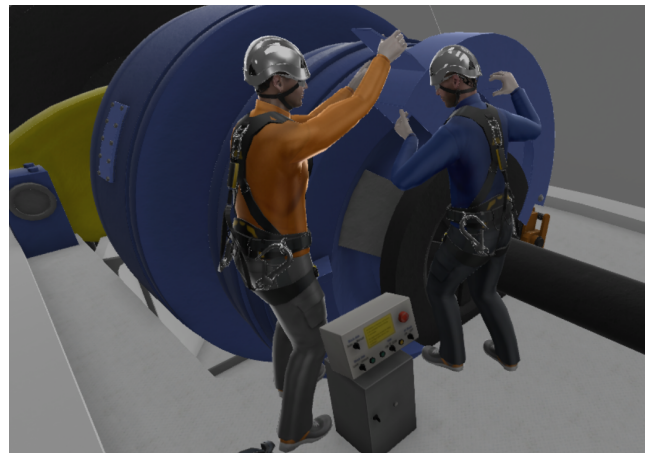


Figure 7: Collaborative action between two technicians

with an endoscope. This procedure necessitates the intervention of two technicians. Hence, we have introduced a virtual agent that has to cooperate with the trainee to achieve the scenario goal, as illustrated in figure 7. Here, the user named Florian plays the role of the trainee, and the role of the technician agent is played by Sebastien which is a virtual agent. The whole scenario lasts around 15 minutes.

The first part of the scenario takes place in the company workshop, where the technicians must discover their task, verify if the weather is acceptable for a maintenance operation, and choose suitable tools to take to the wind turbine. The second part is the arrival on the site, and consists in preparing the maintenance task: calling the wind turbine operator, stopping the wind turbine. The third part is the maintenance operation itself: a series of collaborative actions on the multiplier such as opening the access cover and using an endoscope, all the while respecting safety checks and actions. The final part of the scenario is dedicated to putting the different parts back in working order and the relaunching

of the wind turbine.

The cooperation between the trainee and the virtual agent that accompanies him is supported through shared actions and shared plans that are synchronized through dialogues. The virtual agent exhibits both the reactive and proactive dialogue capabilities that rely on the semantic modeling of the virtual environment and of the task activity using the MAS-CARET meta-model (Chevaillier et al. 2011). Reactive conversation capabilities allow the agent to understand and respond to the questions asked by the trainee. Thus, the trainee can seek the information in order to understand and progress towards the achievement of the goal. The virtual agent can then use any data contained in the different models regarding the technician activities and associated roles, actions, resources and other objects in the environment, their properties and operations. For example, in figure 8 (B), the trainee asks about the current action of the technician agent. Furthermore, the trainee can also request the agent to perform some actions, e.g. in this case the trainee asks the virtual agent to choose the spray bottle as shown in figure 8 (C).



Figure 8: Reactive conversation behavior

The agent can also proactively communicate with the trainee in order to establish or maintain the cooperation, to satisfy the anticipated need of information of the trainee, or to handle the resource sharing. For example, in figure 9 (A), when the trainee does not start the next action that should be performed by him, the technician agent proactively asks him to do that action. Furthermore, the agent can provide help to the trainee, by taking in charge a part of the pedagogical actions such as explaining the next task or showing an object. For example, in figure 9 (B), the trainee does not recognize the endoscope, and asks the technician agent about it. The agent uses a pedagogical action, in this case *turning towards* the target object and *highlighting* it, along with the dialogue utterance in order to pro-

vide required information. More details on the architecture of the virtual agent may be found in (Barange et al. 2014; Barange, Pauchet, and Saunier 2016).



Figure 9: Proactive and pedagogical actions

## V. CONCLUSION AND PERSPECTIVES

The purpose of this work is to redefine the roles of the different actors playing a part in the design of VR learning environments. The main idea is that the job expert and the education specialist should be put forward in the design of such environments, and that the teacher / trainer should be able to reuse and adapt the pedagogical scenarios without the help of computer scientists.

In order to achieve this, we propose a methodology that externalizes the applicative models and the pedagogical models from the application. These models, expressed through UML diagrams, are then interpreted by the virtual environment to instantiate scenarios based on the meta-model of the system and of the assistance decided by the teacher. This methodology also allows to develop generic pedagogical actions that can be reused for different scenarios and environments.

Massive experiments are being conducted to assess the relevance of the learning environments in real settings. More than 3,000 students have already used the wind turbine simulator, ranging from 9th grade to engineering schools, in scenarios which are adapted to the students' grades, from the discovery of a technical system to the design of such a system.

In this article, we have also shown an example of application through a wind turbine environment with three different scenarios. Another environment (a power plant with cogeneration unit) has been developed as part of the EAST project. They all share a part of the UML meta-model.

In the future, we plan to evaluate the virtual agent that cooperates with the trainee through two types of actions: cooperative actions and pedagogical assistance based on dialogue and show by example actions.

Another perspective of this work is to design applicative models with higher degrees of genericity. For example, parts of the maintenance procedure are similar in several application domains. Furthermore, we plan to study the re-usability of the pedagogical primitives in the context of augmented reality, where real and virtual elements are mixed up.

#### REFERENCES

- [1] Barange, M.; Kabil, A.; De Keukelaere, C.; and Chevaillier, P. 2014. Collaborative Behaviour Modelling of Virtual Agents using Communication in a Mixed Human-Agent Teamwork. *Int. Journal On Advances in Intelligent Systems* 7(3 & 4):423–438.
- [2] Barange, M.; Pauchet, A.; and Saunier, J. 2016. Agents pédagogiques interactifs pour l'apprentissage d'une tâche procédurale. In *Workshop Affect, Compagnon Artificiel, Interaction*.
- [3] Barot, C.; Lourdeaux, D.; Burkhardt, J.-M.; Amokrane, K.; and Lenne, D. 2013. V3S: A virtual environment for risk-management training based on human-activity models. *Presence: Teleoperators and Virtual Environments* 22(1):1–19.
- [4] Boud, A. C.; Baber, C.; and Steiner, S. J. 2000. Virtual reality: A tool for assembly? *Presence: Teleoper. Virtual Environ.* 9(5):486–496.
- [5] Chevaillier, P.; Trinh, T.-H.; Barange, M.; Devillers, F.; Soler, J.; De Loo, P.; and Querrec, R. 2011. Semantic modelling of virtual environments using MASCARET. In *Proceedings of the Fourth Workshop on Software Engineering and Architectures for Realtime Interactive Systems SEARIS'11*.
- [6] Edward, L.; Lourdeaux, D.; Barthes, J.-P.; Lenne, D.; and Burkhardt, J.-M. 2008. Modelling autonomous virtual agent behaviours in a virtual environment for risk. *International Journal of Virtual Reality* 7(3):13–22.
- [7] Gavish, N.; Seco, T. G.; Webel, S.; Rodriguez, J.; Peveri, M.; and Bockholt, U. 2011. Transfer of skills evaluation for assembly and maintenance training. In *BIO Web of Conferences*, volume 1, 00028. EDP Sciences.
- [8] Kallmann, M., and Thalmann, D. 1999. Direct 3d interaction with smart objects. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST '99*, 124–130. New York, NY, USA: ACM.
- [9] Koper, R., and Tattersall, C. 2005. *Learning design : a handbook on modelling and delivering networked education and training*. Springer.
- [10] Koper, R. 2001. Modeling units of study from a pedagogical perspective: the pedagogical meta-model behind eml.
- [11] Le Corre, F.; Fauvel, C.; Hoareau, C.; Querrec, R.; and Buche, C. 2012. Chrysaor: an agent-based intelligent tutoring system in virtual environment. In *International Conference on Virtual Learning*, 39–45.
- [12] Lugin, J.-L., and Cavazza, M. 2006. Ai-based world behaviour for emergent narratives. In *Proceedings of the 2006 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology, ACE '06*. New York, NY, USA: ACM.
- [13] Martel, C.; Vignollet, L.; Ferraris, C.; and Durand, G. 2006. Ldl: a language to model collaborative learning activities. In *Proc. of the 2006 World Conference on Educational Multimedia, Hypermedia and Telecommunications (ED-MEDIA'2006)*.
- [14] Mikropoulos, T. A., and Natsis, A. 2011. Educational virtual environments: A ten-year review of empirical research (1999-2009). *Comput. Educ.* 56(3):769–780.
- [15] Mollet, N., and Arnaldi, B. 2006. Storytelling in virtual reality for training. In *Edutainment*, 334–347.
- [16] Retbi, A.; Idrissi, M. K.; and Bennani, S. 2013. A visual educational modeling language to help teachers in learning scenario design. In *Proceedings of World Academy of Science, Engineering and Technology*, number 73, 601.
- [17] Sanagustín, M. P.; Emin, V.; and Hernández-Leo, D. 2012. Considering the space in the design of learning activities: the isis and 4spices models applied to science inquiries. In *IEEE 12th International Conference on Advanced Learning Technologies*, 159–163. IEEE.
- [18] Saraos Luna, A.; Gouranton, V.; and Arnaldi, B. 2012. Collaborative Virtual Environments For Training: A Unified Interaction Model For Real Humans And Virtual Humans. In *Learning by Playing. Game-based Education System Design and Development*, 1–12.
- [19] Stevens, J. A.; Kincaid, J. P.; et al. 2015. The relationship between presence and performance in virtual simulation training. *Open Journal of Modelling and Simulation* 3(02):41.