INTEGRATING KNOWLEDGE FROM VIRTUAL REALITY ENVIRONMENTS TO LEARNING SCENARIO MODELS
A Meta-modeling Approach

Nicolas Marion, Ronan Querrec and Pierre Chevaillier
LISyC, ENIB, UEB
CERV, 29280 Plouzané, France

Keywords: Learning scenario, VRLE, IMS-LD, Meta-modeling.

Abstract: This paper focuses on learning scenario modeling for Virtual Reality Learning Environments (VRLE). Learning scenario models used in computer-supported learning environments are usually not able to describe educational activities implying interaction of learners with a virtual environment. In this paper, we propose an IMS-LD extension making it possible to describe executable educational activities taking place in virtual worlds. Moreover, the model described in this paper is generic in the sense that it can create scenarios regardless of the nature of virtual environments or application domain.

1 INTRODUCTION

A learning scenario describes educational activities taking place during a training session, and is usually written by the trainer. This paper focuses on learning scenarios modeling for sessions where some learning activities use virtual reality learning environments (VRLE). A VRLE is a computer-supported learning environment (CSLE) that uses virtual reality (VR) technology in order to immerse learners in a virtual environment. In the context of learning scenarios modeling, the specificity of VRLE in comparison to classic CSLE is related to the nature of activities that take place in the scenario, that imply actions of actors in a virtual environment.

Educational modeling languages that describe learning scenarios usually define five types of information (Koper, 2001):

- Prerequisites: prerequisites describe knowledge or skills the learner should have in order to take advantage of the learning scenario.
- Learning objectives: learning objectives are knowledge or skills to be gained by learners that achieve the learning scenario.
- Activities: it consists in the description of activities that can be performed in the environment by the different actors, learners or teachers, and their scheduling.
- Roles: they describe the involvement of the users in the learning scenario, and activities they have to perform.
- Environments: an environment describes the context of execution of educational activities. It contains necessary resources to the execution of activities.

Existing learning scenario models in the domain of CSLE (Koper et al., 2003; Rodríguez-Artacho and Verdejo Maíllo, 2004), generally consider educational activities as "black boxes", described by a textual description. Activities are considered atomic, in the way that their execution is not described in the scenario. Only their inputs (resources) and outputs (outcomes) are taken into account.

On the other side, most VRLE include an authoring tool that allows the description of learning activities execution in a virtual environment (Munro, 2003; Gerbaud et al., 2008). Learning scenario models included in these authoring tools usually have two problems. 1) They don't describe the integration of the activity in virtual environment into a more global educational process. 2) They are generally not reusable because of their specificity in relation to a particular

Marion N., Querrec R. and Chevaillier P. (2009).
INTEGRATING KNOWLEDGE FROM VIRTUAL REALITY ENVIRONMENTS TO LEARNING SCENARIO MODELS - A Meta-modeling Approach.
In *Proceedings of the First International Conference on Computer Supported Education*, pages 253-258
DOI: 10.5220/0001976102530258

domain, like maths (Koedinger et al., 2004), to a particular task, like assembly operations (Brough et al., 2007), or to a particular pedagogical strategy, like discovery learning (Van Joolingen and de Jong, 2003).

Our learning scenario model proposition is grounded on the observation made by (Guéraud and Cagnat, 2006) that learning scenarios describing situations in which learners interact strongly with interactive educational tools (like virtual environments in our context) must provide information on the internal progression of educational activities. However, (Burgos et al., 2007) pointed out the difficulties encountered when trying to integrate simulations or games in an executable learning scenario. A full integration of the simulation in the learning flow should make it possible to receive information from the learning scenario (initial configuration, environment modification...) and to send information generated during its execution (actions performed, state of the world...). The constraint it brings is that a specific wrapper, linking the simulation and the learning scenario, should be created for each simulation. Then, even if the learning scenario model is as generic as classical CSLE models, we encounter the same problem as with VRLE authoring tools because the integration of the virtual environment requires specific development.

There is a need to provide models and tools allowing trainers to create learning scenarios integrating activities in virtual environment, regardless of the nature of those environments, and directly executable on a virtual reality platform. The learning scenario model we propose is based upon IMS-LD (Koper et al., 2003), a specification that has been normalized and widely accepted in the field of CSLE. IMS-LD allows to write scenarios describing the five types of information presented in introduction (prerequisites, learning objectives, activities, roles and environments).

In order to fully integrate a virtual environment in the learning flow, a both-way communication is necessary so that: 1) the virtual environment can be adapted to the profile of learners and activities they previously performed and 2) the learning flow can be modified based on what happens in the virtual environment in real time. It is thus necessary to extend IMS-LD so that the learning flow can take into account learners activity in the virtual environment. In section 2 we present the abstraction level of the model, and more precisely the virtual environment meta-model that it uses. In section 3 we show how IMS-LD can be extended to integrate activities in virtual environments in learning scenarios.

2 MODEL'S ABSTRACTION LEVEL

As previously said, the integration of a virtual environment in the learning flow requires a both-way communication between the learning flow and the virtual environment. To enable this communication, the learning scenario must be able to reference concepts used in the virtual environment. To do so, an approach consists in describing all those concepts in a domain-specific model of the application. This domain-specific model gathers information about types of entities that can be found in the environment, their properties, behaviors, *etc.* Thus, a learning scenario linked to such a model can express properties about the environment using domain-specific concepts. At this point, two abstraction levels can be identified: the virtual environment (level 0) and the model of this environment (level 1). In order for the learning scenario model not to be specific to one domain, the concepts of this domain (level 1) must be considered as interchangeable input data of the learning scenario model. For the learning scenario model to be able to handle these data, it uses a meta-model (level 2) that describes concepts used in domain-specific model.

To sum up, a learning scenario takes place in a virtual environment (level 0) and references the model of this environment (level 1). The scenario is expressed using a learning scenario model that uses the virtual environment meta-model (level 2). Figure 1 shows the links between learning scenarios, learning scenario model and the different virtual environments modeling level.

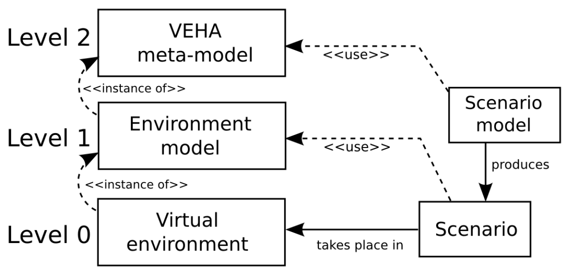

Figure 1: Representation of links between learning scenarios, learning scenario model and the different virtual environments modeling level.

This meta-modeling approach can be compared with the content layer of the PALO model (Rodríguez-Artacho and Verdejo Maíllo, 2004). Unfortunately, the meta-model they use contains only two concepts (entity and relationship), and is not expressive enough to describe complex virtual environments. For example, it cannot describe the dynamic behavior of entities, or actions that can be performed by users in a virtual environment.

The MASCARET (Buche et al., 2004) (Multi-Agent System for Collaborative, Adaptive and Realistic Environment for Training) project is intended to design realistic VRLEs. MASCARET contains several models that describes different parts of VRLE. In this paper, we describe two models that are used by the learning scenario model: the virtual environment meta-model and the organisational model.

Virtual Environment Meta-model: VEHA
VEHA (Marion et al., 2007) is a meta-model designed to define virtual environments, providing a semantics allowing artificial or human agents to build a representation of it.VEHA meta-model is based upon UML 2.1[1]. Figure 2 presents a overview of VEHA.

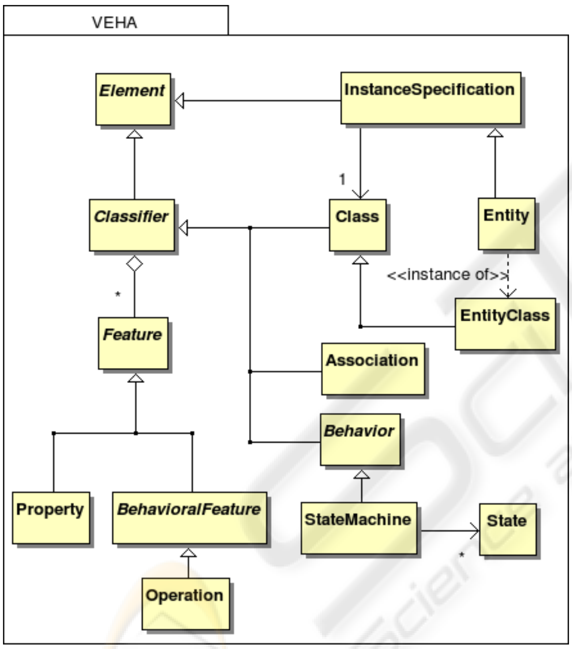

Figure 2: A part of VEHA meta-model.

Concepts of domain-specific models are represented by instances of the `Class` class. The structural (`Property`) and Behavioral (`BehavioralFeature`) properties of classes are associated with `Class` *via* the `Feature` class. Instances of these classes (entities actually evolving in the environment) are defined by the `InstanceSpecification` class.

To explicit knowledge about virtual reality, VEHA extends UML meta-model adding classes making it possible to model objects that are geometrically represented in the virtual environment, and which are therefore controllable by the user. A VEHA environment is a set of objects with graphical representations. The `Entity` class can represent such objects. Entities are instances of a particular class: `EntityClass`. It owns not only `Features`, but also information related to the shape, geometry, positioning, *etc.*(not represented on the figure).

VEHA provides an informed virtual environment meta-model that explicits characteristics of environments and entities that compose them. In addition, it allows the introspection of the domain-specific model (level 1), as well as the virtual environment (level 0). This meta-model can be used by a learning scenario model to express a learning scenario that references concepts of the domain. Thus, the scenario can describe properties of the environment that have to be observed by the learning flow during learners activity, as well as modifications to bring to the environment, based on the progress of the learning flow execution.

MASCARET's Organisational Model
MASCARET allows the creation of virtual environments in which humans and virtual agents can interact by playing roles within organisations. This is done *via* an Agent–Role–Organisation structure, described in (Buche et al., 2004). It is important for the learning scenario to reference organisational entities that exist in a virtual environment, in order to make it possible for users of the learning scenario to play specific roles in the virtual environment. The learning scenario model proposed in this article uses MASCARET, more precisely its virtual environment meta-model (VEHA) and its organisational model. The scenario model allows to create scenarios describing learning activities in virtual environments, as long as those virtual environments are described in MASCARET. MASCARET has already been used to create virtual environments for learning like GASPAR (Marion et al., 2007), SÉCURÉVI (Querrec et al., 2003) and a physics lab work (Baudouin et al., 2008).

3 LEARNING SCENARIO MODEL

In this section, we present how IMS-LD can be extended to integrate activities taking place in virtual environments. This section contains three parts, describing three aspects of learning scenarios that need to be extended to integrate activities in virtual environments: the pedagogical organisation model (3.1), the property model (3.2) and the environment model (3.3).

[1]Unified Modeling Language 2.1: http://www.omg.org/docs/formal/07-11-01.pdf

3.1 Pedagogical Organisation Model

This part focuses on the organisation of the different roles that perform activities in the scenario. In IMS-LD, every information of the scenario is contained in the `learning design`. A learning design contains a set of roles. Two types of role exist: learner roles and staff roles. In the scenario, roles are associated to activities that represent activities that users playing that role have to perform. Proposed model adds the notion of pedagogical organisation. This notions is based on MASCARET'S organisational model. Figure 3 presents the pedagogical organisation model we propose, as well as the part of MASCARET on which it is based.

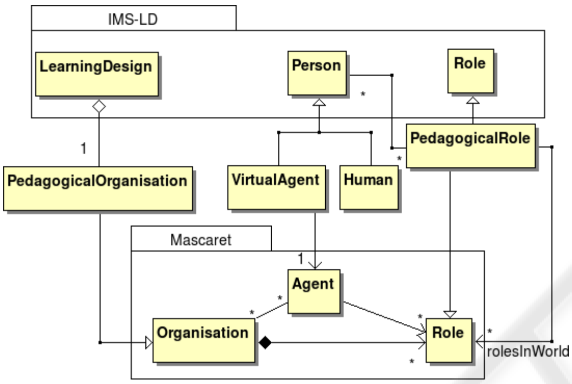

Figure 3: Pedagogical organisation class model.

Every `LearningDesign` contains one pedagogical organisation. Pedagogical organisations, are defined using the notion of organisation described in MASCARET. A pedagogical organisation is composed of pedagogical roles. In comparison to a MASCARET role, a pedagogical role adds two information:

- a link to one or many roles of the virtual environment (`rolesInWorld`). This linked can be used by the trainer to specify that a user will play one or several roles of the virtual environment (in addition to his pedagogical roles) in order to perform a collaborative activity.
- a link to one or several instances of the `Person` class, that represent the users of the scenario that will play this role at run-time, and perform associated activities. These persons can be humans or virtual agents.

This extension of IMS-LD organisational model allows the learning scenario to take into account the organisational structure of the virtual environment. The trainer can associate a pedagogical role to one or several roles in the virtual environment so that users playing that pedagogical role are automatically associated to the corresponding role in the virtual environment for a specific activity. an trainer or an intelligent tutoring sustem (ITS).

3.2 Properties Model

The principle of properties is that the trainer can describe variables on which tests can be done, and describe the progress of the scenario based on the results of those tests. This mechanism adds flexibility for the trainer to describe the scheduling of the scenario.

In IMS-LD specification, a property represents a variable that can have a type and a value. Properties defined by IMS-LD can have different scopes: local (same value for all users in a scenario), global (same value for every user in every scenario), personal (different value for every user) or role (same value for every user playing the same role).

In order to make it possible for the trainer to take into account the virtual environment in the learning scenario, it is necessary to extend the definition of property made by IMS-LD. In the learning scenario model, a property can not only represent a variable as in IMS-LD, but also an entity's property, an entity's state-machine or the fact that an action has been performed in the virtual environment. The properties model is based on UML's properties model, already implemented in VEHA. Figure 4 represents the class diagram of properties model.

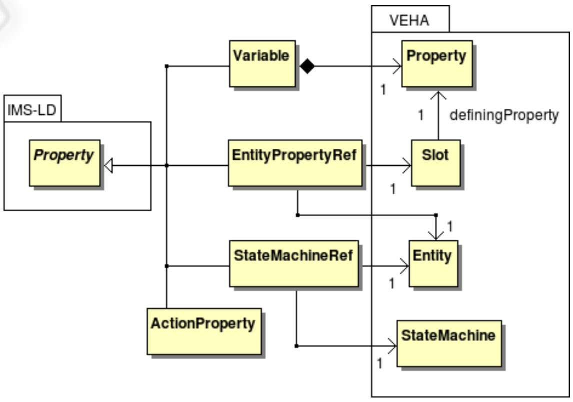

Figure 4: Properties model of proposed learning scenario model.

This figure shows four types of property:

- Properties as defined by IMS-LD are represented by the `Variable` class. A variable contains a VEHA `Property`, that defines the data type and initial value of the variable. During run-time, one or many slots (property instances) are instantiated for each variable (depending on the type of the variable; for example, for a personal value will be instantiated one slot by user).

- An `EntityPropertyRef` represents a link to a property of an entity in a virtual environment. The difference with a `Variable` is that an `EntityPropertyRef` references an existing slot, that belongs to an instantiated entity of the virtual environment.
- A `StateMachineRef` represents a link to a state-machine of an entity in a virtual environment. The value of such a property is the current state of the referenced state machine.
- An `ActionProperty` allows to know if a specific action has been performed in the virtual environment. `Actions` are defined in the domain-specific model and describe the set of actions that can be performed by actors (human or virtual) in the virtual environment. `ActionProperty` model is not detailed in this paper.

Those four types of property behave the same way in that that we can retrieve their value or modify their value, with the exception of `ActionProperties` from which we can only retrieve the value.

The properties defined by the trainer are used to define conditions in the scenario. These conditions associate an expression (defined by a set of properties and their values combined with boolean operators) and a pedagogical action.During run time, conditions are tested by the platform, and pedagogical actions associated with true conditions are triggered. As in IMS-LD, pedagogical actions can be: change the value of a property, send a notification or change the visibility of a component of the scenario (make available or unavailable activities, environment, resources, ...). In addition, we added a new action called `VRAction`. The `VRAction` class represents a set of pedagogical actions that can be applied to a virtual environment, given a certain context. The various actions are based on the list of pedagogical assistances defined by (Lourdeaux et al., 2002).
Properties, expressions, conditions and pedagogical actions make it possible for the trainer to add run time individualization to the scenario based on learners' actions. Our proposition adds three new types of property that allow the description of learners' expected activity in virtual environments, and pedagogical actions to perform.

3.3 Environment's Model

Every activity takes place in an environment. Figure 5 represents corresponding class model.

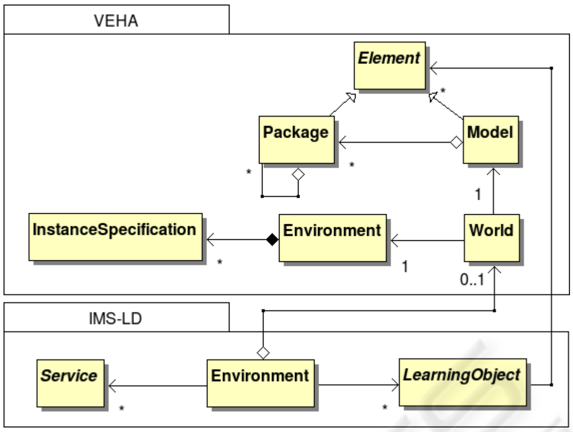

Figure 5: Environment's class model.

In IMS-LD, an environment contains learning objects and services. In the context of VRLE, the notion of environment is a bit different. In addition to learning objects and services, an environment contains a virtual world in which learners are immersed and act. To take this specificity into account, proposed model adds a virtual world to the environment *via* the `World` class defined in VEHA. A world represents an informed virtual environment created with VEHA and contains two main information:

- A domain-specific model for the application (`Model`, level 1). This model contains all the concepts described by a domain expert (as classes, associations, *etc.*), types of entities composing the environment, their properties, *etc.* the whole being gathered into packages.
- A virtual environment (`Environment`, level 0). This environment describes an actual instantiation of the domain-specific model and contains `InstanceSpecification` objects (*cf.* section 2).

Every information about the virtual environment can be referenced by the scenario and properties about entities of this environment can be defined (*cf.* 3.2). Proposed model adds a link between a learning object and VEHA elements. This association makes it possible to express links that can exist between virtual environment entities and learning objects outside the virtual world. For example, it can describe the association between a virtual tool and its instructions of use in PDF format. Then, this link can be used during simulation by an trainer or a pedagogical agent to provide struggling learners an appropriate resource.

4 CONCLUSIONS AND PROSPECTS

This paper describes a learning scenario model able to integrate VRLE in the learning process. The main advantage of this model, based on a meta-modeling approach is that it is generic, in that virtual environments can be integrated, regardless of their nature or domain. This model extends IMS-LD on several aspects. First, it makes it possible to take into account the organisational aspect of virtual environments. Then, three types of properties have been added (`EntityPropertyRef`, `StateMachineRef` and `ActionProperty`) so that conditions about virtual environments' state can be written. Finally, the model integrates virtual worlds in IMS-LD environments, and thus makes it possible to create links between learning objects and elements of the virtual environment. The learning scenario model described in this paper has been used to create a learning scenario for SÉCURÉVI (Querrec et al., 2003), a MASCARET-based virtual reality application designed to train firefighters. The main prospects of this work focus on the evaluation of this model by didactics expert of different domains. The goal of this evaluation is to check that the scenario model is flexible enough to take into account characteristics specific to different domains as well as characteristics specific to different pedagogical strategies.

ACKNOWLEDGEMENTS

This study has been carried out with the financial support from Région Bretagne.

REFERENCES

Baudouin, C., Chevaillier, P., Le Pallec, A., and Beney, M. (2008). Feedback on design and use of a virtual environment for practical lab work. In *Proceedings of the Virtual Reality International Conference, VRIC 2008*, pages 117–125.

Brough, J., Schwartz, M., Gupta, S., Anand, D., Kavetsky, R., and Pettersen, R. (2007). Towards the development of a virtual environment-based training system for mechanical assembly operations. *Virtual Reality*, 11(4):189–206.

Buche, C., Querrec, R., De Loor, P., and Chevaillier, P. (2004). Mascaret : A pedagogical multi-agent system for virtual environment for training. *Journal of Distance Education Technologies*, 2(4):41–61.

Burgos, D., Tattersall, C., and Koper, R. (2007). Re-purposing existing generic games and simulations for e-learning. *Computers in Human Behavior*, 23(6):2656–2667.

Gerbaud, S., Mollet, N., Ganier, F., Arnaldi, B., and Tisseau, J. (2008). GVT: a platform to create virtual environments for procedural training. In *Proceedings of IEEE VR*, pages 225–232.

Guéraud, V. and Cagnat, J.-M. (2006). Automatic semantic activity monitoring of distance learners guided by pedagogical scenarios. In *Innovative Approaches for Learning and Knowledge Sharing*, pages 476–481. Springer.

Koedinger, K., Aleven, V., Heffernan, N., McLaren, B., and Hockenberry, M. (2004). Opening the Door to Non-programmers: Authoring Intelligent Tutor Behavior by Demonstration. *LECTURE NOTES IN COMPUTER SCIENCE*, pages 162–174.

Koper, R. (2001). Modeling units of study from a pedagogical perspective. *Educational Technology Expertise Centre, Open University of the Netherlands, First Draft, version*, 2.

Koper, R., Olivier, B., and Anderson, T. (2003). IMS Learning Design Information Model. *IMS Global Learning Consortium*.

Lourdeaux, D., Burkhardt, J.-M., Bernard, F., and Fuchs, P. (2002). Relevance of an intelligent agent for virtual reality training. *International Journal of Continuous Engineering and Life-long Learning*, 12(1/2/3/4):131–143.

Marion, N., Septseault, C., Boudinot, A., and Querrec, R. (2007). Gaspar : Aviation management on an aircraft carrier using virtual reality. In *Cyberworlds 2007 proceedings*, pages 15–22.

Munro, A. (2003). Authoring simulation-centered learning environments with RIDES and VIVIDS. In *Authoring tools for advanced technology learning environments*, pages 61–91. Kluwer Academic Publishers.

Querrec, R., Buche, C., Maffre, E., and Chevaillier, P. (2003). SécuRéVi : virtual environment for firefighting training. In Richir, S., Richard, P., and Taravel, B., editors, *5^{th} virtual reality international conference (VRIC'03)*, pages 169–175, Laval, France. ISBN: 2-9515730-2-2.

Rodríguez-Artacho, M. and Verdejo Maíllo, F. (2004). Modeling educational content: The cognitive approach of the palo language. *Educational Technology & Society*, 7(3):124–137.

Van Joolingen, W. and de Jong, T. (2003). SimQuest, authoring educational simulations. In *Authoring tools for advanced technology learning environments*, pages 1–31. Kluwer Academic Publishers.