MASCARET: Creating Virtual Learning Environments from System Modelling

Abstract

The design process for a Virtual Learning Environment (VLE) such as that put forward in the SIFORAS project (SImulation FOR training and ASsistance) means that system specifications can be differentiated from pedagogical specifications. System specifications can also be obtained directly from the specialists' expertise; that is to say directly from *Product Lifecycle Management* (PLM) tools. To do this, the system model needs to be considered as a piece of VLE data. In this paper we present Mascaret, a meta-model which can be used to represent such system models. In order to ensure that the meta-model is capable of describing, representing and simulating such systems, MASCARET is based SysML[1], a standard defined by Omg.

Keywords: Virtual reality, System modelling, Virtual Learning Environment, Product LifeCycle Management.

1. Introduction

Virtual reality has been used for many years in teaching. Virtual learning environments (VLE) are primarily designed to be used in professional training where they are predominantly used to teach learners how to use or maintain industrial systems [1]. These systems are extremely complex and a high level of expertise is required to be able to design such systems. They are usually created by experts in the given field. However, in traditional VLE design processes for training in such systems, all of this expertise needs to be rewritten every time, which is costly and can lead to errors. Furthermore, system design is combined with the conception of the training course during the development phase when creating a VLE. The system model cannot therefore be reused for other learning situations. The design process for a VLE such as that put forward in the Siforas project (SImulation FOR training and ASsistance) means that system specifications can be differentiated from pedagogical specifications. Equally, the system specifications can be obtained directly from the specialists' expertise; that is to say directly from the *Product Lifecycle Management* (PLM) tools [2].

To do this, the system model needs to be considered as a piece of VLE data. With this in mind, we propose a meta-model which can be used to represent these system models: MASCARET. In order to ensure that the meta-model is capable of describing, representing and simulating such systems, MASCARET is based on a standard defined by OMG: SysML. SysML is a UML extension specifically devised for designing industrial systems. The advantage of basing our work on this standard is that it is already used in industry and it is therefore compatible with many modellers (both free and commercialised). Furthermore, an exchange format in Xml has also been standardised. This is known as Xmi format.
To design a VLE from PLM specifications, we need 4 kinds of information [3]:

- the 3D geometry (Collada files),

[1] http://www.sysml.org

- the system structure (Blocks, properties, compositions in SysML),
- the behaviour of the entities within the system (StateMachines, Event, Signals in SysML),
- the system's usage or maintenance procedures (Activities, partitions, actions in SysML).

The first problem to be resolved is the fact that each company has its own unique PLM system. The aim of the SIFORAS project is to work on an exchange standard between PLM tools: PLCS. This standard can be used to express a number of different points of view of the system. The PLCS system representation is supplemented by other standards. Existing work has shown how the concepts of PLM translate to SysML. Figure 1 illustrates our proposal for a new VLE design flow. Both the geometry and certain structural elements of the system are generated by CAD tools such as CATIA. The behavioural aspects are generated by systems experts using specific tools. This information is then translated into pivot languages: SysML for the system model and COLLADA for the geometry. These files are the input for the pedagogical workshop which enables the instructor to design the learning situation.

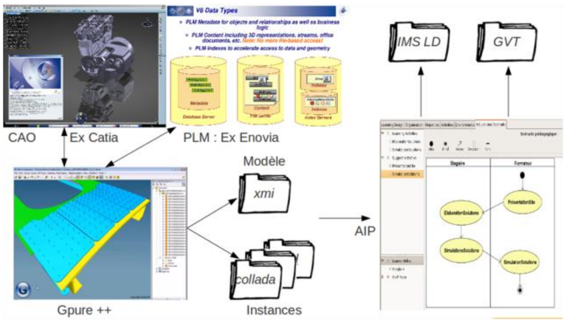

Figure 1: Workflow

The second problem is to provide operational semantics for each of the meta-model's concepts. This will first of make it possible to automatically simulate the system in the virtual environment along with the reactions to the user's actions. In addition, it will act as a knowledge base for the agents' reasoning with the aim of problem solving or pedagogy. In section 2 of this article we will identify a subset of SysML which can be used to model the system, but above all a subset for which we provide the operational semantics. Using this subset it is possible to automatically generate the simulation in VR and provide input for the agents' knowledge base. In section 3, we will show how these knowledge bases can be manipulated by agents' behaviour with a pedagogical objective.

2. System Modelling

3.1 Structural Modelling

The structural aspect of the system can be described by a set of **blocks**. These blocks are connected by **composition** relationships which expresses the break-down into sub-systems. A block is defined by its **properties**, which in turn are defined by a **name** and a **type**. This type can be expressed in a **unit** and in a field of validity (a **dimension**). The block carries out **operations** stated in the structural description of the system. The realisation of these operations is defined during the description of the system's behavioural aspects.

3.2 Behavioural Modelling

There are many different ways of describing the block's fundamental autonomous behaviour. This behaviour can be governed by both its state and its evolution, and we therefore suggest describing it using state machines. The description of behaviour using state machines hinges on two concepts: states and transitions between states. Transitions are partly defined using events. Figure 3 presents an example of a state machine within the framework of the electronic programmer.

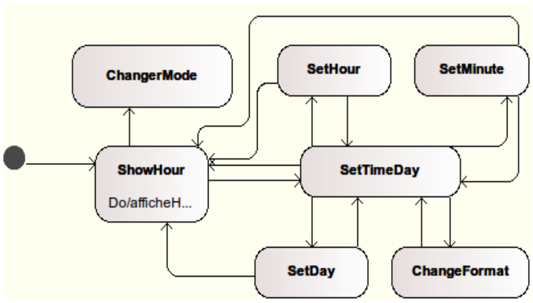

Figure 3: Describing the behaviour of a block using a state machine.

The procedures which we are focusing on are usage, maintenance or malfunction diagnostics procedures. These procedures are defined by sequencing actions on the system. This sequencing of actions can be considered as an activity. The activity can be carried out by multiple roles and directs resources. Figure 4 presents an example of a procedure within the framework of the electronic programmer.

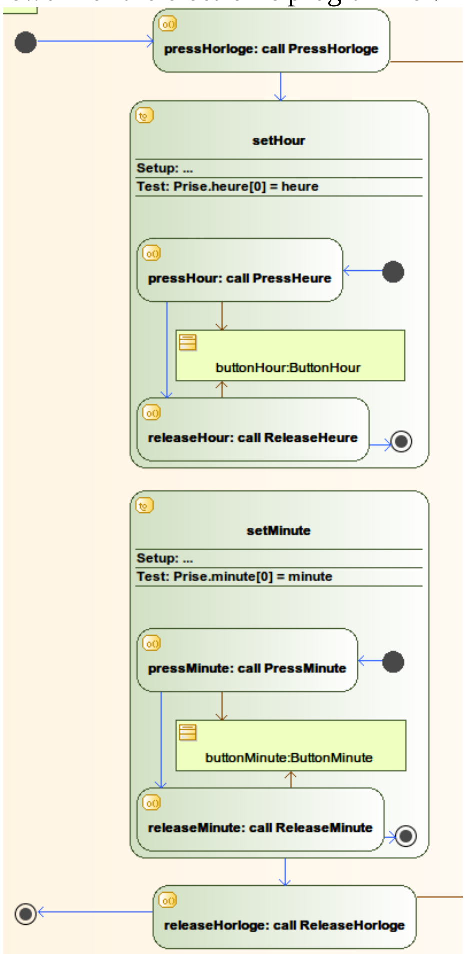

Figure 4: Describing a procedure with an activity diagram.

4. Mascaret for VLE

Mascaret was created in order to be used in designing VLE. Within this context we developed two types of VLE tools. The first, Poseidon, is used to define the sequencing of the pedagogical activities. The second, Pegase, is an intelligent tutor who assists the instructor in his or her pedagogical role during the exercise.

4.1 Training Course Design

Poseidon [4] is a generic training course model for VLE. It is used to specify the configuration of the virtual environment which hosts the exercise, roles and pedagogical activities. This scenario model is inspired by the pedagogical scenario traditionally used in VLE (pedagogical objectives and pre-requisites, environments, roles and activities), which makes it compatible with the norms defined in this field, such as IMS-LD. Nonetheless, unlike IMS LD, Poseidon can be used to explicitly define the execution of activities and more particularly, activities in virtual environments. In order to define the environment, sequencing and activities, Poseidon draws on Mascaret, enabling it to be used independently of expert models. For the instructor, defining the configuration of the environment consists of instantiating the domain model. The initial states of the entities therefore need to be instantiated, positioned and fixed. The instructor is able to identify particular states of the environment within the scenario. For each element in the model or the environment, the instructor can give a link to external resources (video or other documentations). In Mascaret, pedagogical organisation is described as a configuration of agents and expresses two specific roles: the instructor and the learner. When the scenario is instantiated, these roles will be assigned to a human, a reactive agent (whose behaviour is suggested in Poseidon), or an ITS such as Pegase. As is the case in Mascaret or Pegase, the pedagogical scenario is a knowledge base used by the agents, or the humans.

The pedagogical scenario is in itself a sequence of pedagogical activities. In order to define this sequence, we draw on activities from Uml 2.1. The description of these pedagogical activities is based on procedures defined in the business model. The instructor can provide further specific information (common error situations, pedagogical actions to be carried out, etc.). The instructor also has access to a range of pedagogical actions (explaining restrictive situations, drawing attention to an object, etc.).

4.2 Intelligent Tutor

Pegase [5] is a generic and adaptable intelligent tutoring system. Its objective is to provide pedagogical assistance to the learner, and to help the instructor. Pegase offers a representation of those models which are traditionally required in ITS, the difference being that the starting points of Pegase are business and pedagogical models. The overall functioning of Poseidon can be seen in figure 6. The business model makes a direct link with Mascaret. Thus, Pegase can reason about the environment, both on a structural and on a behavioural level. It also has access to business sequences and procedures. It can also reason at the class level (structural properties or a given type of device, for example), and the instance level (the action of the procedure completed by the learner). Thanks to these qualities, Pegase can detect learner errors and identify the context of those errors (previous actions, actions relating to pre or post conditions, elements on which the actions are based, roles that the actions must carry out, etc.).

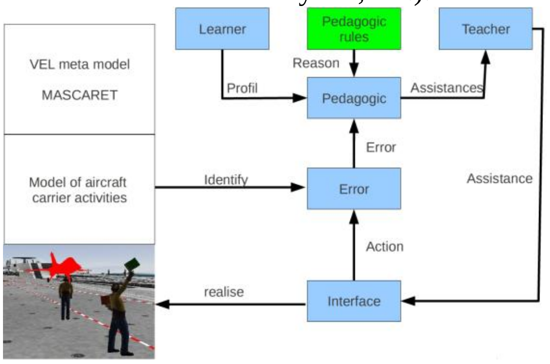

Figure 6: Describing the behaviour of the ITS using the business model.

The pedagogical model is a set of condition/action rules structured on three levels of abstraction. On the most concrete level, the actions are the execution of pedagogical assistance in the virtual environment (highlighting in red the target object of the concrete action, etc.). The most abstract level brings together different traditional pedagogical procedures (active, expositive, interrogative, etc.). The actions are therefore the activations of a set of rules from the lower level specifying pedagogical attitudes (disruption, suggestion, etc.). On every level of abstraction, the condition element of all of these rules is based on the pedagogical situation. This brings together information about the learning (level, course, etc.), the error, the context of the action or even an error model. The choice of rules is weighed up, and the ITS suggests a sequenced list of pedagogical assistance to the instructor. The instructor then choses the most appropriate response.

5 Conclusion

In this article we presented an approach and a model for designing learning situations in virtual environments. These situations are the direct results of PLM tools used in industry. The suggested meta-modelling approach brings an intermediary language by which the system (structures, behaviours and procedures) can be described and simulated automatically in a virtual environment. This also provides ontology, representing the knowledge base of the agents which use it both for problem-solving and with pedagogical objectives.

The suggested meta-model is a specialised version of SysML for virtual reality. This guarantees that we will be able to express the different aspects dealt with in modelling the system. SysML also includes two new concepts which we are going to add to our future work to complete our meta-model. Firstly, SysML will enable us to tackle the physical aspects (mechanics, electrics, fluids, etc.) of the system using parametric diagrams. Secondly, the notion of requirements and validation of requirements will enable us to use the simulation in virtual reality in order to validate the system and/or establish pedagogical objectives.

[1] Rickel J. and Johnson L. (1999) Animated agents for procedural training in virtual reality : Perception, cognition and motor control. In Applied Artificial Intelligence 13(4-5):343--382

[2] Morad M. Monticolo D. Gomes S. Sagot J.C. (2010) A collaborative design for usability approach supported by virtual reality and a multi-agent system embedded in a PLM environment. In computer-aided Design 42:402--413

[3] Oren T., Yilmaz L. (2012) Synergies of simulation, agents and system engineering. In Expert system with application. 39(1):81--88

[4] Marion N., Querrec R. et Chevaillier P. (2009) Integrating knowledge from virtual reality environments to learning scenario models. A meta-modeling approach. Dans International conference of Computer Supported Education, pages 254--259, Lisbonne, Portugal.

[5] Buche C. and Querrec R. (2011) An expert system manipulating knowledge to help learners into virtual environment. Expert systems with applications. 38(7) :8446--8457.