Supporting Acquisition of Knowledge to Personalize Interactive Learning Environments through a Meta-Model

Marie Lefevre, Stéphanie Jean-Daubias, Nathalie Guin

KONG, S.C., et al. (Eds.), ICCE2009; ©2009 Asia-Pacific Society for Computers in Education.

Supporting Acquisition of Knowledge to Personalize Interactive Learning Environments through a Meta-Model

Marie LEFEVRE, Stéphanie JEAN-DAUBIAS, Nathalie GUIN
Université de Lyon, CNRS
Université Lyon 1, LIRIS, UMR5205, F-69622, France
{Marie.Lefevre, Stephanie.Jean-Daubias, Nathalie.Guin}@liris.univ-lyon1.fr

Abstract: Outsourced personalization of Interactive Learning Environments (ILE) requires having knowledge on these ILE: their teaching description (type of activities, parameters for exercises generation...) and their technical description (files location, content of configuration files...). To make possible this personalization, we propose a meta-model to acquire, through an expert, relevant knowledge necessary to personalize an ILE. We combine this meta-model with two processes: the first allows using the meta-model to create a specific model to an ILE, and the second allows using this model to personalize the ILE.

Keywords: Meta-model of knowledge, models of knowledge, Interactive Learning Environments (ILE), personalization, externalized configuration

Introduction

Interactive Learning Environments (ILE) Interactive Learning Environments (ILE) designed in research laboratories are rarely widely used by teachers. This is partly due to the fact that ILEs are generally not adaptable to specific needs of teachers. The aim of the work presented here is to provide tools that allow teachers to adapt the content and the environment of a lambda ILE to his teaching habits and to the individual needs of learners. Our approach consists in providing teachers with a sole generic tool that enables them to define their educational choices and, from the learner's profiles, personalize the ILE they want learners to use. By personalization of ILE, we mean personalization of activities, sequences of activities, accessible functionalities, feedback provided to the learner and interface. This involves providing teachers with a tool that helps them in this personalization; making an outsourced setting of ILEs; and knowing, for each ILE, the parameters to set.

Implementing personalization requires fulfilling certain specific needs. In section 1, we study some related work and we show what is lacking in several system to fulfill these needs. In section 2, we present our AKEPI meta-model for define, for customizable ILEs, knowledge necessary for their personalization and. Finally, in section 3, we present the future work raised by the work presented in this paper.

1. Related work

1.1 Helping the teacher to personalize learning

In order to assist teachers to personalize learner's activities, two main approaches have been proposed. The first one makes use of *pedagogical scenarios*. A scenario describes educational goals and learning situations by identifying how the educational resources will be implemented within a context of learning [4]. In our approach, we want to provide sequences of activities suited to the skills identified in the learner's profile at some point, while letting the teacher decide later the context of use: work at home, extra teaching, assessment… So the same sequence of activities can be used in different contexts, unlike the pedagogical scenarios.

The second approach devotes a part of ILE to its setting by the teacher, as do [2, 5]. Thus, teachers may choose activities or define the parameters of the activities generation suited to their learners. But this personalization must be made each time by a teacher, and only some systems allow such a personalization. In our approach, we want to allow teachers to personalize softwares, either if they have a part specific to the setting by the teacher or not.

1.2 Piloting a system outside of it

In order to control software from the outside, the first approach is to design an epiphyte system, i.e. a system grafted on any application to spy on and to think about the user's actions. The epiphyte systems and their hosts are independent both from the conceptual point of view (their architectures are independent) and from the software point of view (hosts are not designed according to the epiphyte system) [7]. So this approach consists in developping, for each host, a special system. For similar hosts, epiphyte systems may have the same architecture, but their contents differ for each host to observe and control it.

The A second approach consists in considering each ILE as a component and in providing a device allowing combining these components. Recommendations for this component approach have been proposed in [9]. According to the authors, among others things, components should be *scrutinizable* to allow the observation of some of their mechanisms, states, objects...; be *scriptable* to allow some assumption; and be *indexable*. The device managing these components must, among others things, be able to establish a components language command; to manage data formats; and to manage a components database for indexing interesting functionalities. Hence the component approach requires indexing ILE with metadata that must focus on the technical aspects and on the educational use of ILE and requires allowing a communication of data and models between each component.

In the following, we retain from these approaches the need of information on the system to pilot. For satisfying this requirement, the epiphyte system approach offers a real time control system while the component approach combines many softwares. By contrast, our approach provides a system enabling ILE software setting before they are used in autonomy by learners.

1.3 Describing an ILE using metadata

For describing an ILE, one approach uses oriented content description standard. Among these, the Dublin Core [1] has a set of general metadata describing any type of resources while others standards are more specific to areas or to professions. With regard to educational resources, we can cite LOM [6], which allows us to describe the educational tools, notably e-learning softwares, or SCORM [10] which allows us to create interoperable and reusable structured teaching objects. These standards are not suitable for the description of components constituting ILEs because they do not have enough metadata to describe the software appearance of these components, such as technical needs (hardware and software) to run it, their characteristics (services, properties, methods) and let-alone rules and dependencies required to assemble it by computer specialist developing ILEs.

In response to this observation, the software component description pattern LSCM proposes a set of metadata devoted to software components [8]. LSCM has two sections: a section common to all classes of components describing software engineering, and a section specific to the category of educational software components describing the pedagogical and didactic appearance of software component. Hence, LSCM allows us to describe components for reuse, but not for personalization as we want.

Thus, the various standards proposed in current work do not offer solutions to describe systems for personalization: description of their educational content allowing personalization and description of technical knowledge allowing stepping in the system to implement this outsourced personalization.

Thus, no system allows outsourced personalization of several ILEs to be performed by adapting sequences of activities offered to learners, from their profile, while letting the teacher interfere in the choice of personalization. Offering such a system requires identifying the properties characterizing activities and environment of an ILE and identifying technical information allowing acting on this ILE. In the next section, we describe the AKEPI meta-model as a response to this need.

2. AKEPI, a meta-model to acquire knowledge necessary for outsourced personalization of ILEs

The outsourced personalization of an ILE requires obtaining some knowledge on the ILE. First, the properties characterizing the activities and the environment, allow for each ILE, to know the type of proposed activities and how it is possible to choose or generate activities by this ILE. These properties also permit to know if the ILE proposes customizable sequences of activities, customizable functionalities… To personalize the environment and activities offered to learners, it is necessary to have a description of all the parameters relating to activities, sequences of activities, functionalities and / or interface of the ILE. Then, teaching competences associated with these properties allow teachers to be assisted when making educational choices when setting the ILE. Finally, to be able to act on an ILE, it is necessary to be able to modify files allowing personalization. It is therefore necessary to have technical information on the ILE, as the presence of an exercise generator, the place and content of configuration files, etc.

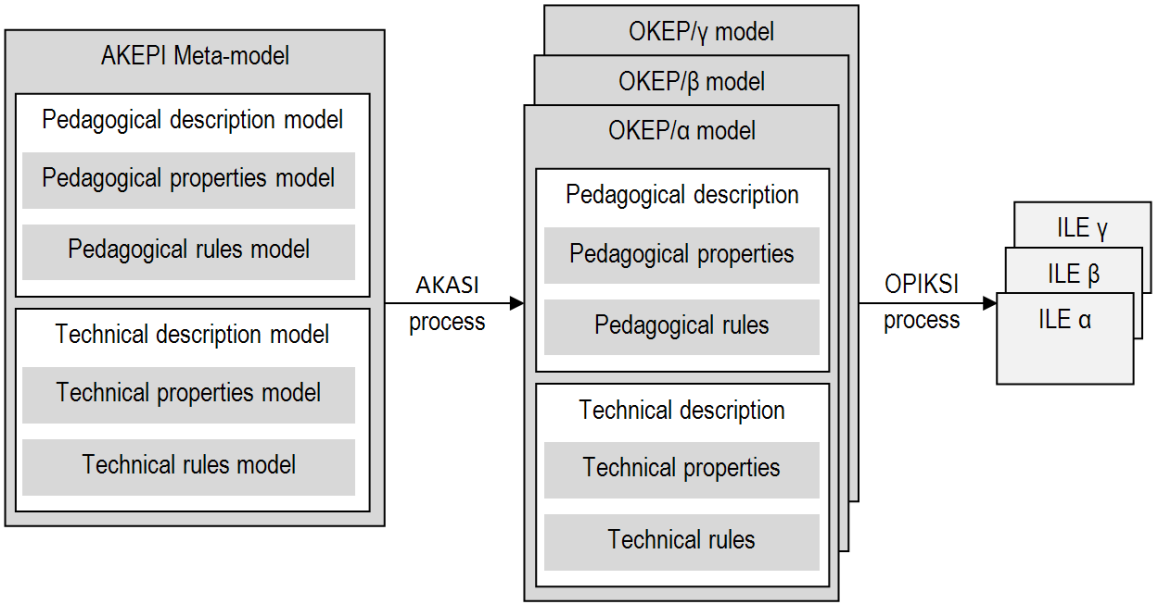

Fig. 1. Overview of the AKEPI meta-model and associated processes.

The AKEPI meta-model (Acquisition of Knowledge Enabling Personalization of ILEs) is designed to facilitate the expert's complex task of identifying knowledge needed to the outsource personalization of an ILE. This meta-model defines the type of information needed to personalize an ILE and is made up of two parts: the pedagogical description of an ILE and the technical description (see Fig. 1). We combine two processes with this meta-model: the first one instantiates the AKEPI meta-model with specific knowledge to an ILE α to get the OKEP/α model (Operational Knowledge Enabling Personalization of the ILE α), and the second one uses the OKEP/α model to personalize the ILE α.

2.1 Knowledge of the meta-model

The AKEPI meta-model defines the type of information an expert must provide to create, for an ILE α, the OKEP/α model allowing its outsourced personalization. This information should cover the pedagogical description and the technical description of ILEs (see Fig. 1). The pedagogical description aggregates the customizable properties of a system, with the associated competences, and the rules to manage these properties. The technical description aggregates all the necessary information to act concretely on the system: localization of the system, folder of the exercises generator or exercises database, place and content of configuration files and rules for fill in these files. In its implementation, the AKEPI meta-model is defined using the XML Schema formalism [11]. In the remainder of this section, we describe the ILE pedagogical description model of the AKEPI meta-model.

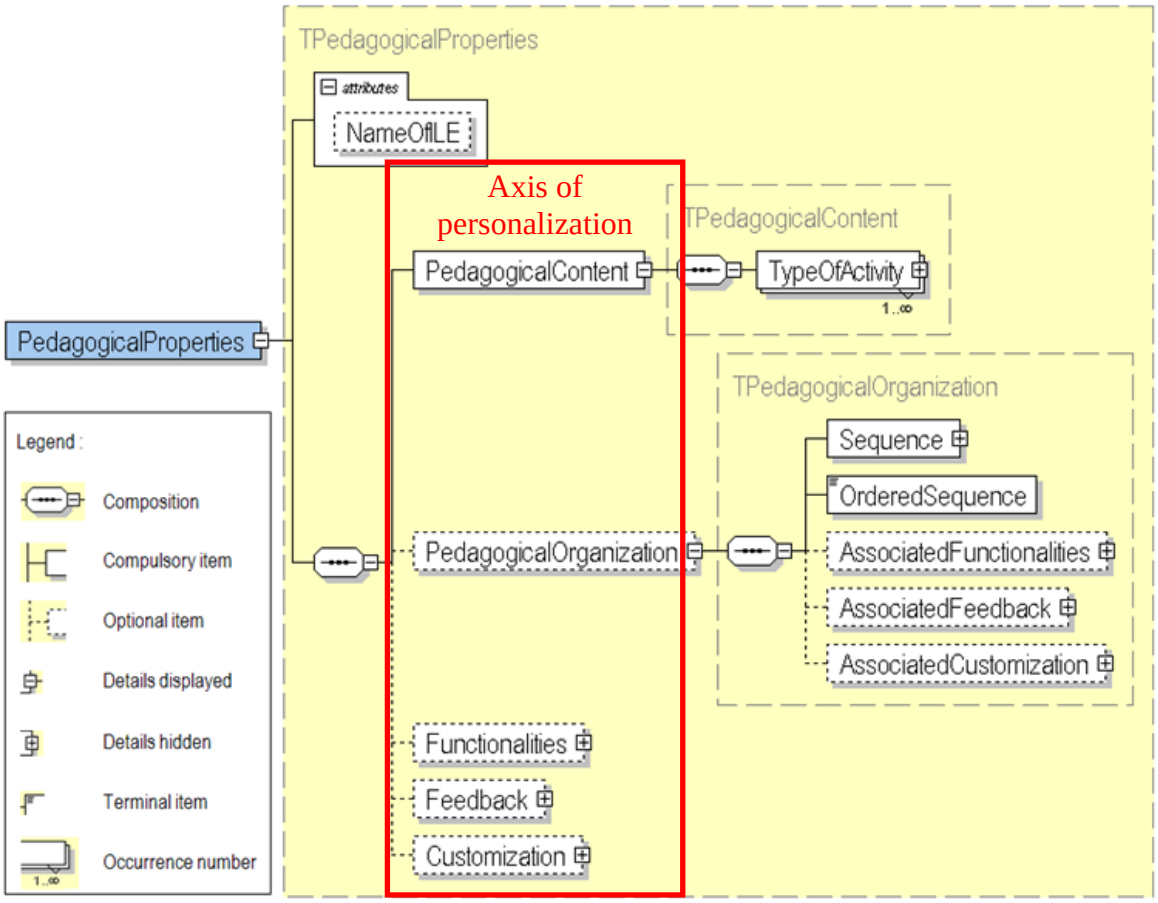

Fig. 2. Overview of the AKEPI meta-model and associated processes.

The pedagogical description model aggregates the customizable properties of an ILE and the rules managing these properties. By studying about thirty systems (freewares, sharewares, commercial softwares or from research), we have identified the types of parameters stepping in the ILEs personalization. We have formalized these parameters to define the AKEPI meta-model. The figure 2 presents an overview of the part of the AKEPI meta-model defining different types of pedagogical properties. We can see that these parameters are organized into five axis according to the ILE's part on which they act: pedagogical content, pedagogical organization, functionalities, feedback and customization. The *pedagogical organization* indicates how sequences of activities are made. The elements *functionalities*, *feedback* and *customization* contain a description of the elements allowing setting it (access to diagnosis, interface language...).

As for the *pedagogical content*, it allows describing how different types of activities are be chosen or generated. A type of activity is defined, as we can see in Fig. 3, by a *name* ①, e.g. "problem" or "arithmetic exercise"; by the list of *parameters* ② to choose an activity of this type in a database, or to generate it with the generator in the ILE; and possibly by parameters of *associated functionalities*, *associated feedback* and *associated customization* ⑦ that may be associated with this type of activity, e.g. "access to assistance" or "access to diagnosis". Each one of the *parameters* ② is defined by a *name* ③; by the *competences* ④ that can be associated; by a *scale* ⑤ respecting one of the six types of scales proposed covering the different cases detected in the systems studied; and eventually by the *category* ⑥, or even the *sub-category* to which it belongs. This precision allows organizing parameters in order to aggregate the parameters that have links between themselves. For example, the generation of an additive word problem brings into play parameters

concerning numbers, but also parameters concerning the language to provide statements more or less grammatically complex.

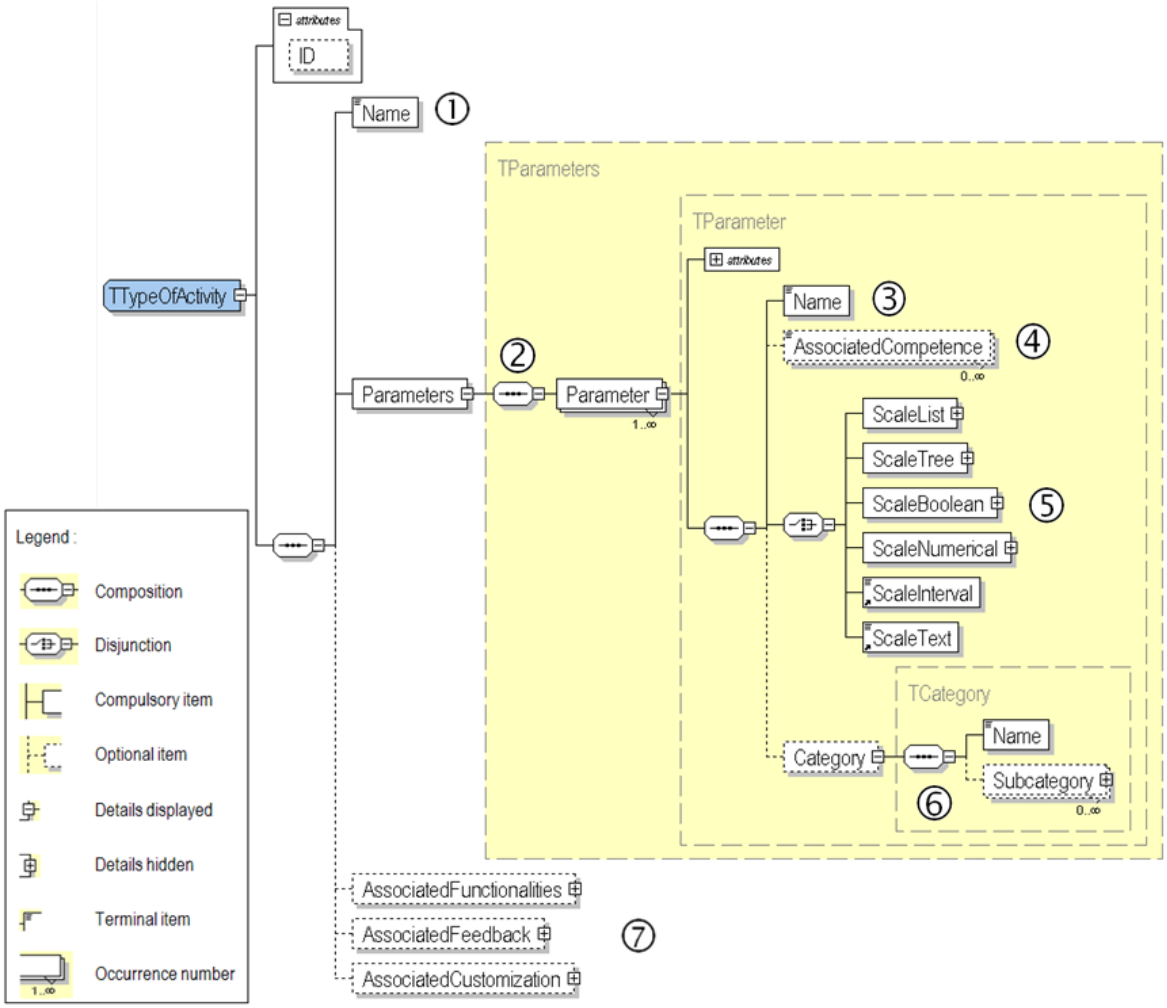

Fig. 3. Detail of an activity in the AKEPI meta-model.

The pedagogical description thus contains the customizable properties of an ILE. The AKEPI meta-model combine these properties with pattern of rules allowing the expert to formalize the connections and the constraints that could link the properties he has identified. Hence, the expert can define constraints on the properties using rules.

On the same principle, the technical description of ILEs in the AKEPI meta-model contains a set of properties associated with a set of rules.

2.2 Processes associated with AKEPI meta-model

We have presented the principles of AKEPI meta-model, as well as an overview of knowledge it can acquire. We will now indicate how this meta-model can be used by presenting two processes, AKASI and OPIKSI, that we associate with it (see Fig. 1).

The AKASI process, allowing defining ILE's specific knowledge in order to create an OKEP/x model, is decomposed into three steps (see Fig. 4). In the first step, the expert specifies, using an interface created dynamically from the AKEPI meta-model, the pedagogical properties of an ILE. In the second step, the expert defines the rules for the management of pedagogical properties respecting the format of pedagogical rules defined in

the AKEPI meta-model. In the final step, the expert specifies the technical properties and technical rules for the ILE. By implementing these rules, the XSL file converts the constraints made by teachers on an ILE to produce a configuration file.

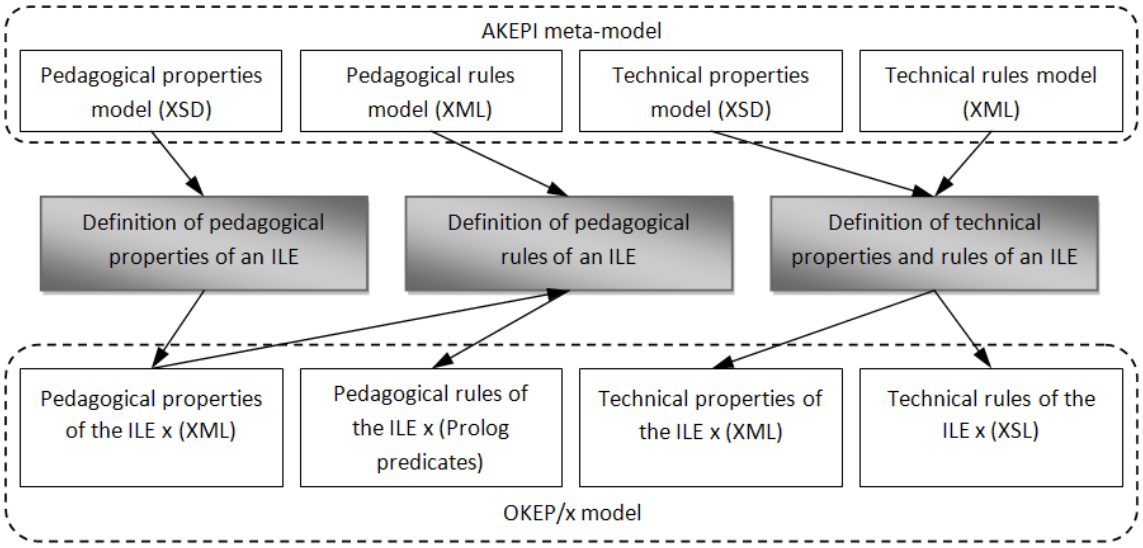

Fig. 4. AKASI process.

The OPIKSI process, allowing the use of the OKEP/x model to personalize an ILE, is decomposed into four steps (see Fig. 5). In the first step, an interface is automatically generated from the pedagogical properties of OKEP/x model. In the second step, the teacher uses this interface to define constraints on the generation or on the selection of activities already generated, and defines constraints on the choice of parameters acting on the functionality and the interface of the ILE x. The pedagogical rules, in the form of Prolog predicates, associated with an inference engine implemented in Prolog, manage dynamically the interface offered to the teacher. In the third step, from the constraints specified by the teacher and the pedagogical properties of OKEP/x model, the system generates sequences of activities suitable to learners and defines parameters to customize the environment. In the last step, the system uses the technical description of the OKEP/x model to create the configuration files specific to the ILE x.

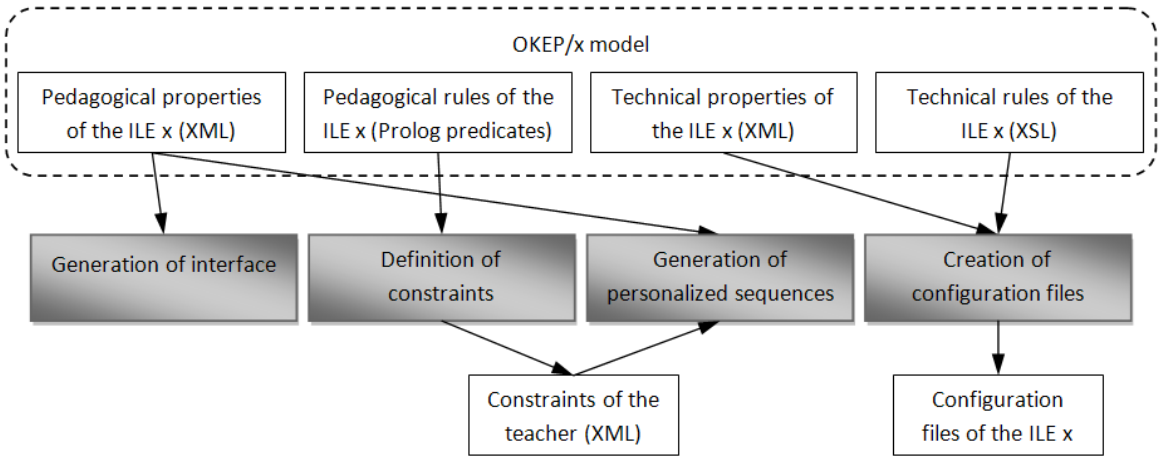

Fig. 5. OPIKSI process.

3. Discussion and future work

In this paper, we have presented the AKEPI meta-model to acquire the necessary knowledge to personalize an ILE from outside itself. This meta-model is used to help the expert to define a pedagogical and technical description of an ILE. The instantiation of the AKEPI meta-model with knowledge specific to an ILE x can create the OKEP/x model of the ILE x, used to personalize it.

In order to define the AKEPI meta-model, we have studied about thirty educational softwares including intelligent tutoring systems, microworlds and simulators. The meta-model presented allow defining knowledge for each one of these types of systems. We now have to validate it by comparing to new systems, not included in the initial study and so define its limit of use. One limit may be the validity of our proposals for feedback personalization. Indeed, we have not encountered any system that allows the customization of feedback. We must confront the AKEPI meta-model to such systems to check if our proposals remain valid.

Furthermore, we must test the AKEPI meta-model and the associated processes with the target public. The AKEPI meta-model and its processes have been implemented in the Adapte module belonging to the EPROFILEA environment. This module allows personalizing the ILE suited to learners' profiles while leaving the teacher interfere in the choice of personalization [3]. We must therefore make use the integration phase of the Adapte module, corresponding to the AKEPI meta-model and AKASI process, by experts from various ILEs.

References

[1] DublinCore. *Dublin Core Metadata Initiative*. [cited April 2009]; Available from: http://dublincore.org/documents/dcmi-terms/.

[2] Duclosson, N., Jean-Daubias, S. & Riot, S. (2005). AMBRE-enseignant : un module partenaire de l'enseignant pour créer des problèmes. *Proceedings of Environnements Informatiques pour l'Apprentissage Humain (EIAH'2005)* (pp 353-358). Montpellier, France.

[3] Lefevre, M., Cordier, A., Jean-Daubias, S. & Guin, N. (2009). A Teacher-dedicated Tool Supporting Personalization of Activities. *Proceedings of ED-MEDIA 2009 - World Conference on Educational Multimedia, Hypermedia & Telecommunications* (to appear). Honolulu, Hawaii.

[4] Lejeune, A. & Pernin, J.-P. (2004). A taxonomy for scenario-based engineering. *Proceedings of Cognition and Exploratory Learning in Digital Age (CELDA 2004)* (pp 249-256). Portugal.

[5] Leroux, P. (2002) *Machines partenaires des apprenants et des enseignants - Étude dans le cadre d'environnements supports de projets pédagogiques*. Habilitation à Diriger des Recherches en Informatique, Université du Maine, Le Mans.

[6] LOM. *LOM v1.0. Final Draft Standard for Learning Object Metadata. IEEE Standards Department. IEEE P1484.12.1-2002*. 2002 [cited 17 novembre 2008]; Available from: http://ltsc.ieee.org/wg12/.

[7] Paquette, G., Pachet, F., Giroux, S. & Girard, J. (1996). EpiTalk, a generic tool for the development of advisor systems. *International Journal of Artificial Intelligence in Education*, *7*: 349-370.

[8] Rebaï, I., de la Passardière, B. & Labat, J.-M. (2008). To Store and Retrieve Software Components for Interactive Learning Environments: the ECR Repository. *International Journal of Advanced Media and Communication*, *2*(1): 73–95.

[9] Rosselle, M. & Grandbastien, M. (2004). Towards interoperable ILEs: results from a case study in the geometry domain. *Proceedings of International Conference on Computer Aided Learning In Engineering education (CALIE'2004)* (pp 185-190). Grenoble, France.

[10] SCORM. *Advanced Distributed Learning. SCORM Metadata set*. 2001 [cited November 2008]; Available from: http://www.adlnet.gov/scorm/index.aspx.

[11] XSD. *XSD*. [cited November 2008]; Available from: http://www.w3.org/XML/Schema.