ABSTRACT

This paper presents PAMELA, an authoring tool dedicated to teachers who want to add new pedagogical activities for the foreign language students. We detail here the methodology used for developing this environment in collaboration with teachers and the technical solutions retained to allow the widest possible use of the authoring tool through the network. We are particularly interested in the contribution of the XML technologies in the structuration and in the distant exploitation of Interactive Learning Objects. We ask the question of the necessary normalisation of their architecture.

KEYWORDS: Authoring environments, Software Architecture, Interactive Learning Objects, XML technologies, Standardisation.

INTRODUCTION

The ARCAD team of the CLIPS-IMAG laboratory is interested by the methodology of production of interactive pedagogical applications, such as simulations. We want to rationalize this production and decrease the costs of development by:

- Offering each actor (domain specialist, modeler, pedagogical expert, interface specialist, project leader, etc.) formalisms, methods and tools specially adapted to express their own specifications;
- Insuring a coherent integration of the results obtained by the different actors.
- Satisfying quality criteria (such as modularity, evolutivity) for the resulting simulation.

The principal theoretical result of our previous work is a conceptual framework, MARS (Model – Associations – Representation – Scenario) which structures the architecture of a pedagogical simulation. MARS distinguishes three independent components:

- the Model where the system behavior is described
- the Scenario which regroups abstract pedagogical aspects (list of exercises, pedagogical controls of the learner’s activities)
- the Representation which deals with the simulation interface and pedagogical interactions

A last component, the Associations, then links the three previous components.

Based on this generic framework, we proposed to develop two kinds of authoring tools [6]. The first generic approach provides the author with a set of tools which allow him to express a wide variety of simulations. This approach can be used in contexts where the development is done by a software company, according to specifications given by a customer. In this case, a different specialist is in charge of each activity: project leader, model expert, pedagogical expert, HCI expert, etc.

The second specific approach provides the author with a simpler tool dedicated to a specific class of problems. It is adapted to situations where trainers themselves are in charge of the pedagogical material development. In this case, we must provide environments that are more integrated and more sophisticated, in order to process automatically a number of tasks and to offer a coherent assistance to authors.

We have used these generic and specific approaches to develop several authoring tools in different contexts: the Hewlett-Packard Technical Planning Education Center and the ARIADNE Project [4,5]. In previous papers, we have compared these two approaches [6,7,8]. One of the main result was that the specific approach was preferred in contexts where the author had to develop a complete application without assistance of specialists, because it allowed him to concentrate his activity on domain-expertise (the Model) and pedagogical features (the Scenario). The other components of the application (establishing links between model and scenario, creating application interface) can be automatically processed by the authoring environment from specific domain-related rules and templates.

Our first development specific authoring tools concerned technical domains and used an implicit methodology. In this paper, we formalize this methodology and we show how we reused it in a very different context, the PAMELA project.

PAMELA : AN EXAMPLE OF SPECIFIC AUTHORING ENVIRONMENT

The aim of the PAMELA project (in french : Production d’Activités Multimédia pour l’Enseignement des Langues et l’Apprentissage, 1998-2000) was to provide a specific authoring tool dedicated to teachers who want to develop new pedagogical activities for students in foreign languages. The goal of these activities was to complete
traditional activities proposed by traditional or CD-Rom methods (often based on written competencies) by interactive situations centered on oral competencies.

**The development process**
In order to insure an effective matching between the authoring environment and the common needs of this community, we used the following process:

- **Step 1: Needs Analysis.** In this step, we had to animate a working group of ten teachers representing 6 different languages (Chinese, Japanese, English, German, Arab, and French). The main goal of this group was to define, develop and experiment a shareable set of new learning activities centered on oral competencies. This included the precise definition of a common vocabulary and the identification of the common characteristics and differences between different concrete case-studies.
- **Step 2 to 5: Incremental design of the authoring functionalities.** These steps were designed to (1) help teachers to express their problems with paper forms where are formally separated the content features and the control of learner activity (2) use the paper forms to develop a prototype of an authoring tool with RAD techniques (Rapid Application Design) and (3) validate the prototype by testing it with trainers and learners.
- **Step 6: Development of the final authoring tool, after the prototype validation by the working group.**

![INCREMENTAL DESIGN](image)

**Figure 1: The development process**

**The result**
During the incremental design phase, teachers ideas were concretised through successive prototypes. The final developed authoring tool runs from any computer connected to the local network of the Stendhal university. It allows (1) a teacher to quickly create exercises for the different considered languages and (2) a student to access very simply to the exercises. The types of proposed activities were for example:

- connecting and matching sounds, pictures or texts
- localizing a sound fragment
- reconstructing a message from sound fragments

**Software architecture**
As shown on Figure 2, the architecture of the system is based on three components:

![Software architecture](image)

**Figure 2: Software architecture**
**Component 1: Authoring module**

The authoring module, loaded from the server with a simple web browser, allows the teacher to create a formal description of an exercise with a set of visual forms, and to store this description into the database.

![An example of authoring screen](image)

**Figure 3: An example of authoring screen**

**Component 2: Centralised database**

The centralised database contains the formal description of the exercises. This description was expressed in a XML-like language (Cf. Fig. 4).

```xml
<PAMELA Version 0.1>
<SUBJECT> Anglais </SUBJECT>
<AUTHOR> J.Ph.Pernin </AUTHOR>
<EXERCISE_NB> 2 </EXERCISE_NB>
<EXERCISE>
  <NUMBER> 1 </NUMBER>
  <NAME> Exercice N°1   </NAME>
  <TYPE> Localisation   </TYPE>
  <INSTRUCTION>
    <Donner ici la consigne de l'exercice>
  </INSTRUCTION>
.../
```

**Figure 4: A fragment of a formal description of an exercise**

**Component 2: Player module**

The player module, used by the student (and by the teacher when testing the result of authoring), allows to generate automatically and dynamically the student interface from the formal description by using implicit presentation rules.

![The corresponding automatically-generated student interface](image)

**Figure 5: The corresponding automatically-generated student interface**

The first version of the authoring tool and the player have been developed in 1998-99 with Toolbook and the Neuron plugin. Those technical solutions were chosen because of the following reasons:

- Toolbook was, at the beginning of the project, one of the best solutions to quickly develop prototypes of highly interactive multimedia applications with sounds. One of the main constraint during the incremental design was to regularly provide the teachers with a new version of the authoring tool in order to validate the different choices;
- The Neuron plugin allowed to execute the authoring tool or the resulting exercises from any computer of the local network, by using a simple web browser. This satisfied another important constraint: the different users had to be able to retrieve their own working environment without worrying about the place of their workstation.

With this first version, we had obtained a set of very interesting results. The ten teachers implied in the project could use an environment they had themselves defined, and quickly created pedagogical resources adapted to their audience.

The adopted technical solution allowed to reach the main goals we initially fixed. However, we met some difficulties:

- The system, based on a local network, was not available from the user's homes. The teachers particularly needed to modify or create exercises from any place;
- The system was not available on a variety of operating systems (Mac OS, Linux);
- The runtime environment was based on the version 5 of the Neuron plugin of Asymetrix. Frequent updates of this software required regular modifications of the source codes of the authoring and player modules;
• The installation on each workstation of the relevant version of the plugin was impossible in several cases, for example when a more recent version was already installed for another software.

For these reasons we decided to develop a new version, based on a architecture using only internet standards in order to insure a better portability and perenniality.

A NEW ARCHITECTURE DEVELOPED ON INTERNET STANDARDS
In order to solve the problems listed above, we developed a new architecture based on XML and XSLT. The following figure shows how data entered by a teacher is automatically processed to generate the student application.

In a first step, the teacher uses a web-based authoring tool to generate an XML file describing the exercise data. This file must be conform to the Pamela DTD (Document Type Definition) which defines the formal structure of the different types of exercise. This structure, inspired from the QTI specification (Question and Test Interoperability [1]) proposed by the IMS consortium, is the following:

- the “Content” describes data (for example, questions, text or media references)
- the “Resprocessing” part describes the behavior expected from the student (for example, the right answers to a MCQ)
- the “Feedback” describes the information delivered to the student when he manipulates the exercise (for example, an audio message when the answer is wrong)

In a second phase, the XML file is transformed into a student Web application by a XSL processor from the specific PAMELA XSLT stylesheet. The stylesheet rules define how a well-formed Pamela XML document must be transformed in HTML code for the static part of the application and, in Javascript or Java code for the dynamics.

In comparison with the previous one, the new architecture presents real advantages:

- the formal definition of the Pamela DTD insures a greater perenniality to the developed exercises. Even if the technical solutions change in a few years, we will still be able access their content and semantic;
- the decomposition of the DTD in three separate components allows to adapt existing exercises very simply, for example by modifying just one of them, the feedback component for instance; it also anticipates solutions for exploiting those resources in a distant-learning context;
- using only web-based techniques allows to exploit authoring or playing environments in a very wide range of situations; the only constraint being the availability of a recent web browser.

A prototype of this new architecture has been developed with a French company specialized in e-learning, and the authoring tool should be industrialized soon.

Figure 6: New software architecture
NORMALISING ARCHITECTURE OF LEARNING OBJECTS ?

The main results of the PAMELA project are the following:

- the implication of pedagogical experts is necessary to define new high-quality learning objects or activities;
- a formal description is necessary in order to isolate the different features of a learning object; this decomposition improves design and exploitation;
- the most recent techniques based on internet standards (XML, XSL, etc.) allow to implement effective solutions in a very wide panel of situations.

Thus, the question of software architecture becomes central. If we want to develop high-quality pedagogical interactive resources that could be durably integrated within the e-learning platforms, we must propose a common set of shared rules.

Since 1998, a set of important works are related to the normalisation of educational technologies. Today, main efforts concern the description of the pedagogical resources in order to retrieve, share and reuse them. Particularly, the LOM (Learning Object Metadata) [2] standard is currently in the course of approval at the international level.

If these efforts are necessary, it is also essential to carry out a basic reflection on the standardisation of architecture of learning objects. If not, we are likely to find us in a few years with immense libraries of objects, correctly described and indexed, but of which only a small part could be usable or reusable in real contexts of formation.

REFERENCES