Work in Progress - A Framework for Building Interactive Learning Modules

Danilo L. Dalmon, Anarosa A. F. Brandão, Seiji Isotani and Leônidas O. Brandão
University of São Paulo, ddalmon@ime.usp.br, isotani@acm.org, anarosa.brandao@poli.usp.br, leo@ime.usp.br

Abstract - The development of good quality educational software is expensive, time-consuming and faces some underlying issues. In order to deal with such issues, many attempts were made, mainly on code reuse. Following a similar approach, the development of an application framework for implementing a family of interactivity-intense educational software called Interactive Learning Modules – iLM is presented. The framework main goal is to reduce development efforts while being part of a Software Product Line – SPL, an innovative technique regarding interactivity-intense educational software. The framework specification is presented by analyzing the common features of existing iLM. Therefore, the resulting component architecture is outlined. Currently, the design phase is finished and the implementation ongoing. Also, its instantiation is planned for an existing iLM, as a proof of concept.

Index Terms – Authoring Systems, Electronic Learning, Software Architecture, Software Reusability.

INTRODUCTION

The development of good quality educational software is expensive and time-consuming. Besides the usual challenges, developers face interdisciplinary and specific issues in personnel organization [1], instructional design [2], project and implementation [3], use and reuse by teachers [4], code maintenance [5], among others.

Several efforts focused on code reuse were made to meet some of these challenges, including the ones related to component architectures [3], software product lines for digital products [4], aspect-oriented development [6], refactoring [5], and standard specifications and repositories [7]. These related work can be divided into methods and techniques for developing interactivity-intensive systems [1]-[3], which provide assignments to students, or information-intensive systems [4]-[6], which mainly provide access to information.

This paper describes an application framework that specifies Interactive Learning Modules – iLM [8], a family of interactivity-intensive systems. This framework implements a Software Product Line – SPL [9]. To the best of our knowledge, there is no use of similar techniques during interactivity-intensive systems development, but for developing information-intensive systems.

The text is organized as follows: the next section describes the iLM main functionalities, used for specifying the framework core. The framework itself is described in the section after that. And the last section presents the current state of our research and its next steps.

INTERACTIVE LEARNING MODULES – iLM

Families of educational systems have some common attributes to address practical problems. iLM are basically designed to add interactive assignments embedded in a Learning Management System – LMS to support teaching and learning of specific content, such as geometry [8].

In order to develop an application framework tailored to build iLM, we conducted a systematic analysis of existing iLM [10] and found out their specific characteristics, which could be well established by their main, or code, functional requirements. The ones used to specify the iLM framework are pointed out as follows:

• run in browsers: To be used in Web-based courses or simply to be available on the Internet, iLM must be able to work on browsers. This is essential to raise the number of iLM users.

• provide communication with LMS: To improve its integration with LMS, iLM must have a protocol to exchange assignments data, e.g. assessment results, with the LMS it is embedded in. An example using Moodle is shown in [8].

• provide authoring tools for teachers: In order to avoid the adoption of an underlying pedagogical approach, iLM must be designed as authoring tools, leaving to authors (usually teachers) the task of authoring assignments.

• provide manipulate and co-construct types of interactivity [11]: These two types of interactivity allow students to explore, to manipulate and to create domain objects or learning elements during assignments, which may improve the learning process and possibly pedagogical approaches.

• provide feedback to students: Feedback is essential to keep students motivated and to know their current knowledge state [12]. With assessment data, teachers can follow their students’ progress more easily [8].

A FRAMEWORK FOR BUILDING iLM

To reduce the amount of effort needed, to raise the code quality and to establish a systematic development process, a SPL approach is being used with a modified Product Line UML-based Software Engineering Method – PLUS [9].
To implement this SPL, an application framework was chosen. Because of the iLM structure, instead of having many hot spots, such as regular frameworks, it has basically one hot spot without considering Graphical User Interface – GUI variations. This hot spot consists of domain-specific features. The framework core accounts for modeling generic actions (such as saving files) and assignments, leaving to the framework hot spot the handling of domain actions (such as creating a line using two points in geometry). Optional features can be added as plug-ins.

This framework is being developed using a component approach. Figure 1 depicts its architecture: white components belong to the core, light gray GUI components have hot spots to set appearance and dark gray components are the main hot spot that define domain specific features.

- **SystemControl** controls the main flows and the system structure. Other components interact with it by the command design pattern (ICommand).
- **Communication** accounts for data exchange with external systems. IComm interface provides its methods to other components. It uses IAssignCommand when asked for assignment data by LMS.
- **AssignmentControl** controls the flows and features regarding assignments, such as authoring tool and feedback. Other components interact with it by the command design pattern (IAssignCommand). It also provides the assignment model for domain model and GUI.
- **AssignmentGUI** and **BaseGUI** are generic domain independent GUI. They are a kind of hot spot, where the GUI configuration can be changed. **BaseGUI** is a singleton on which the assignments are created. There is an **AssignmentGUI** for each assignment in the iLM, organized in tabs.
- **DomainModel** and **DomainGUI** are the main hot spots of the framework. **DomainModel** must implement IAssignmentModel, and **DomainGUI** must be able to show them to the user, providing interactivity of manipulate and co-construct types.
- **Configuration** stores configuration parameters that are used by every component using the visitor design pattern. In Figure 1, the use of IConfig is omitted.

**CURRENT STATE AND NEXT STEPS OF RESEARCH**

Currently, the framework requirements and features were defined, the framework analysis and design is completed, as well as the detailed component architecture. Implementation has already started.

In order to achieve our goals, after finishing the framework implementation, a cookbook for guiding the framework instantiation and use will be developed. Furthermore, the framework will be instantiated to create new iLM and/or to rebuild existing ones. At this point, we will have a proof of concept for the proposed approach.

**ACKNOWLEDGMENT**

Danilo L. Dalmon is supported by FAPESP grant 2010/06805-2. Leônidas O. Brandão is partially supported by FAPESP.

**REFERENCES**