https://doi.org/10.48009/2_iis_2013_262-268

Issues in Information Systems

Volume 14, Issue 2, pp.262-268, 2013

RIA COMPONENTS AND DESIGN PATTERNS FOR THE DEVELOPMENT OF EDUCATIONAL MATERIALS UNDER THE PARADIGM OF WEB BASED EDUCATION

Rubén Peredo, Escuela Superior de Cómputo (ESCOM)-Instituto Politécnico Nacional (IPN), rperedo@ipn.mx
Iván Peredo, Universidad Politécnica de Querétaro (UPQ), ivan.peredo@upq.mx

ABSTRACT

Component Based Software Development is a growing discipline, developing software based component assembly. The proposal uses to improve the architecture, and a library of Rich Internet Applications components that call IRLCOO, for the development of educational materials under the paradigm of Web Based Education. The fundamental technologies used in our proposal are: components, software design patterns, RIA, and agents. Software components were fundamental to the design and implementation of the proposal due to three advantages: conquering complexity, managing change, and maximizing reusability. Software design patterns provide a robust architecture that allows us to maintain the code over the life of the project, and maximize the reusability of parts of the project. Integrating RIA technologies allow us: integrated multimedia players, multimedia handling and interactivity and an asynchronous bidirectional communication channel. Intelligent software agents are exceptional software of information society tools that independently perform various tasks autonomously, allowing us a high degree of tailoring at runtime for students.

Keywords: Components, Software Design Patterns, Rich Internet Applications (RIA), Education and WBE

INTRODUCTION

The impact of Information and Communications Technology (ICT) effects many areas. Educators can improve and redefine the teaching and learning processes using the ICT. The cheapening of Personal Computers (PC) has resulted in a more extensive use in all academic disciplines [16]. Educators are incorporating tools in the classroom to stimulate curiosity, and provide more support to students in their understanding. Students at all levels have responded to the computer simulations, which make the concepts more attractive and less abstract [15]. Students using technology to get more enhanced reasoning, have better understanding and skills [23]. Computers improve student attitudes and interests, *software design patterns* through personalized learning in a virtual environment more interactive and enjoyable [24].

Macromedia in 2002 with the presentation of Macromedia Flash Multimedia eXperience (MX) [7] first mentioned the Rich Internet Application (RIA). A RIA is a Web application that is designed to provide functionality comparable to a desktop application. RIAs have three common characteristics: rich multimedia and interactivity, integrated multimedia players, and asynchronous communication channel type Asynchronous JavaScript And XML (AJAX).

There are different platforms to develop RIA, being the main the following: Microsoft Silverlight, JavaFX, and Flash Platform. Silverlight is a powerful RIA development tool for creating interactive user experiences for Web and mobile applications. Silverlight is a free plug-in, supporting the .NET framework and compatible with multiple browsers, devices and operating systems, supporting a new level of interactivity [17]. JavaFX is the next step in the evolution of Java as a RIA client platform. It is designed to provide a lightweight, hardware-accelerated Java User Interface (UI) platform for enterprise business applications. JavaFX allows developers preserve existing investments by reusing Java libraries in their applications. They can even access native system capabilities, or seamlessly connect to server-based middleware applications [12]. The Adobe Flash runtimes are ubiquitous client runtimes that let you deliver compelling and expressive applications, RIA content, and video to the world. The Adobe Flash runtimes are mainly two: Adobe Flash Player [9] and Adobe AIR [6], built on top of a set of core RIA technologies to deliver a consistent platform for developing and deploying RIA applications. Flash Platform technologies allow guarantee the success of projects by providing three crucial factors: reach, expressiveness, and consistency. The Flash runtimes are installed on over 98% of all Internet-connected computers and over a billion devices. Industry-leading tools and technologies enable you to go beyond the limitations of HTML. The Flash technologies ensure the

integrity of your creative vision with consistent experiences across operating systems, browsers, and devices without having to write multiple versions of your code [5].

The Flash Platform is solid and a robust component framework to develop RIA, the framework is an open framework that allows customization. The aim of the framework is to facilitate the development of software and maintenance throughout its life cycle, to carry out the framework uses design patterns and libraries of components.

ActionScript is the programming language for the Flash Platform in Flash [7], Flex [8], and Flash Builder [4]; it has evolved to the latest version 3.0 implementing the paradigm Object Oriented Programming (OOP), enabling us to implement refined designs. Flex is an open source framework to build RIA Web applications [8]. Software Design Patterns (SDP) are proven solutions to common problems, enabling us to code with better architectures, improving maintenance and reuse of parts of our projects, allow us to manage change better. The SDP allow to handle large and complex projects with better design structures For the reasons mentioned above was chosen as the platform for our developments.

Component Based Software Development (CBSD) is an emergent discipline that promises to solve the so-called software crisis, relaunching to the software engineering into a new level, developing software based component assembly. The proposal uses software design patterns to improve the architecture, and a library of RIA components that call IRLCOO, for the development of educational materials under the paradigm of Web Based Education (WBE).

The fundamental technologies used in our proposal are: components, software design patterns, RIA, and agents. The software components were fundamental to the design and implementation of the proposal due to three advantages: conquering complexity, managing change, and maximizing reusability. The construction and operation of the components is based on the Composition pattern, enabling us building complex components based on smaller components, and simplifying management through a common Application programming interface (API). Software design patterns provide robust architectures that allow us to maintain the code over the life of the project, and maximize the reusability of parts of the project. Integrating RIA technologies allow us: integrated multimedia players, multimedia handling and interactivity and an asynchronous bidirectional communication channel. Intelligent software agents are an exceptional generation of information society tools that independently perform various tasks on behalf of human user(s) or other software agents, allowing us a high degree of tailoring at runtime for students.

RESEARCH METHODOLOGY

Our research centered upon the following research question: Do educational institutions are taking advantage offered by Web in developing their educational materials for the Web?

The research hypothesis to be tested is the following:

H_1: Educational institutions are not taking advantage of bidirectional communication channel that offers the Web due to its high complexity.

Currently developing educational materials is relatively simple, There are many Web publishing tools to support teachers with different characteristics, enabling that teachers can having Web presence, but the high technical complexity wasted the bidirectional communication channel of the Web, causing teachers in most cases develop educational materials called as eReading, wasting bidirectional communication channel of the Web, and using only the Web as a means of publishing educational materials. This approach has been used by various institutions who seek to have a presence on the Web; the main disadvantage of this approach is the limited adaptation at runtime of educational materials.

Another approach taken by numerous institutions to solve the high complexity to develop educational materials for the Web, is through technical support for teachers in developing their educational materials, the main problem with this approach is that often wasted the experience of teachers in their knowledge areas. The teachers often do not know exactly how to implement common activities in the classroom within Web environments.

Another fundamental problem common is the omission of a pedagogical model in many educational materials from various institutions.

In the technical aspects of virtual environments, there are two fundamental problems: code maintenance and reusability of parts of projects.

Our approach proposes the implementation of advanced tools for supporting teachers based on: components, software design patterns, RIA, agents, and including a pedagogical model. Considering the Advanced Distributed Learning (ADL)- Sharable Content Object Reference Mode (SCORM) and IEEE 1484 architecture – Learning Technology Systems Architecture (LTSA) - IEEE Learning (LTSA) models.

To test this hypothesis show our proposal to develop educational materials and the resulting materials will compare them with educational materials from other institutions.

RESULTS

IRLCOO components were developed by Peredo et al. [20]. The IRLCOO components are a special type of content and assessments for learning type RIA, supporting: reconfiguration, sequencing and dynamic feedback. The IRLCOO components have been improved over the years; the latest changes have focused on redesigning internal architecture of the IRLCOO components using SDP [21], and decoupling of the knowledge model from the application [22].

The changes along the development of a software project are inexorable. Changes in the requirements for the development of a software project are constant, requiring making changes that the original design did not consider. The best way to manage change in our software projects is through a robust design that allows managing changes and modifications.

The SDP make the software can be modified in a simple way, and allows us convenient code maintenance throughout the life of the project. This is because the SDP are proven and functional solutions that have improved over time to better architectures, managing change in a simple way. SDP encapsulate knowledge of experienced software developers enabling reuse this knowledge to similar problems.

The Figure 1 shows Architecture based design patterns on the client side. On the client side have a container IRLCOO component, which contains two subcomponents: content or evaluation IRLCOO component and navigation IRLCOO component, these components are configured using XML files for easy configuration and dynamic sequencing of educational materials in our proposal. Client side implements four main design patterns: Composite, Factory Method, Template Method and Decorator. IRLCOO components are assembled using the Composite pattern, using composite components and indivisible at Run-Time, assembling composite components content and assessment. The Factory Method pattern allows us to decouple the creating objects without specifying the exact class of object that will be created; it was used in our proposal to implement the logic of a test and its questions. The Template Method pattern is an sequence of operations to accomplish a goal, specifying the steps and blocking the order of operations, allows hook operations for extensions, it was used in our proposal to implement indivisible IRLCOO components: Video, Sound, and Animation, because indivisible IRLCOO components have common features only changing the type of multimedia. The Decorator pattern adds unique responsibilities to an object without adding these responsibilities to the complete class; it was used in our proposal to implement decorators to images, enabling us composition of dynamic images at Run-Time.

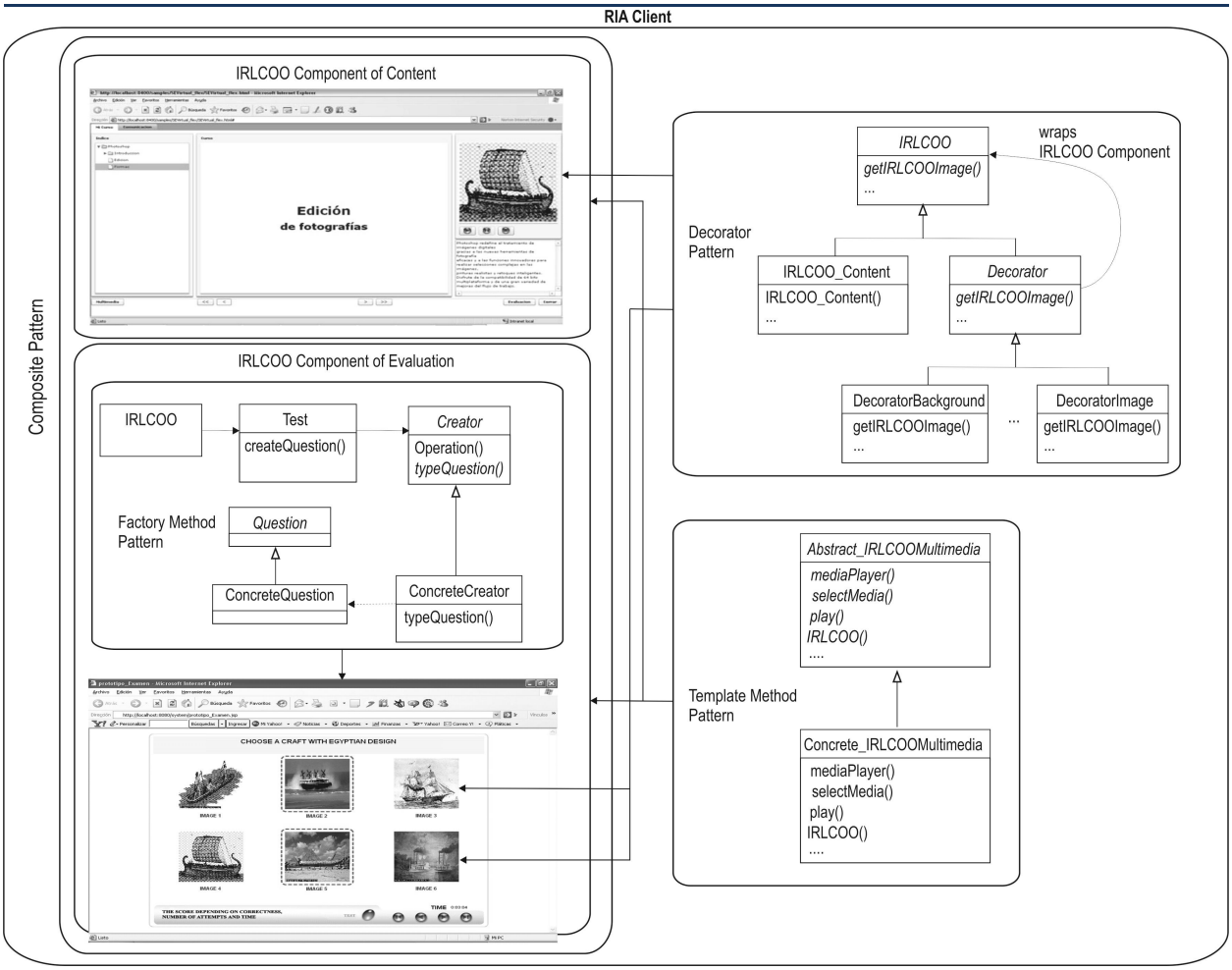

Figure 1. Architecture based design patterns on the client side.

The Figure 2 shows Architecture based design patterns on the server side. Server side implements three main design patterns: Model-View-Controller (MVC), Observer and Singleton. The MVC is a pattern composed of multiple patterns working together, in our proposal: Observer, Singleton and Composite (explained above), to create complex applications. Figure 2 shows the architecture improved of the MVC pattern using interceptors, improving services architecture of our proposal. The Observer pattern enabling us to keep updated the View and the Model in the MVC pattern. The Singleton pattern focuses on ensuring that only a single instance of a class in memory, and has only a single global access to the object, with the goal of optimizing the use of resources; in our proposal the Singleton pattern optimizes access to the Model in the MVC pattern.

Our proposal has several additional modules: ADL-SCORM structured, middleware, Web Applications, MAS, Data Base, and Knowledge Base. The ADL-SCORM structured module builds imsmanifest.xml file using the JDOM API [13], and metadata layer to maintain compatibility with ADL. The middleware module provides services to other modules for different functionalities. The Web Applications module was previously developed [21], highlighting the following: Java Virtual Laboratory, BlazeDS, VoiceXML, Virtual Desktop, Content Authoring Tool, Evaluation Authoring Tool, Problem Based Learning (PBL). The MAS module is the architecture of MAS based on the IEEE 1484 architecture – LTSA. The IEEE 1484 LTSA standard is a high level architecture for Learning Management Systems (LMS) supported education, learning and training that describes the high-level system design and the components of these systems. In general terms, the IEEE 1484 LTSA standard is acknowledged into the Web learning community [1-3]. In our proposal agents replaced the processes of the IEEE 1484 LTSA standard. The implementation of the system was made using the frameworks: Java Agent DEvelopment Framework (JADE) [10] and JADE eXtension (JADEX) [11], implementing the Belief-Desire-Intention (BDI) architecture. Other important framework was Webbridge [11] used for bridging the gap between the View in the MVC pattern and the MAS

platform. The Figure 2 shows the actual MAS platform based on IEEE 1484 LTSA standard. The latest improvement to our proposal introduced a Knowledge Base [22], decoupling the Knowledge Model from our application, enabling us sharing and reuse between users and applications.

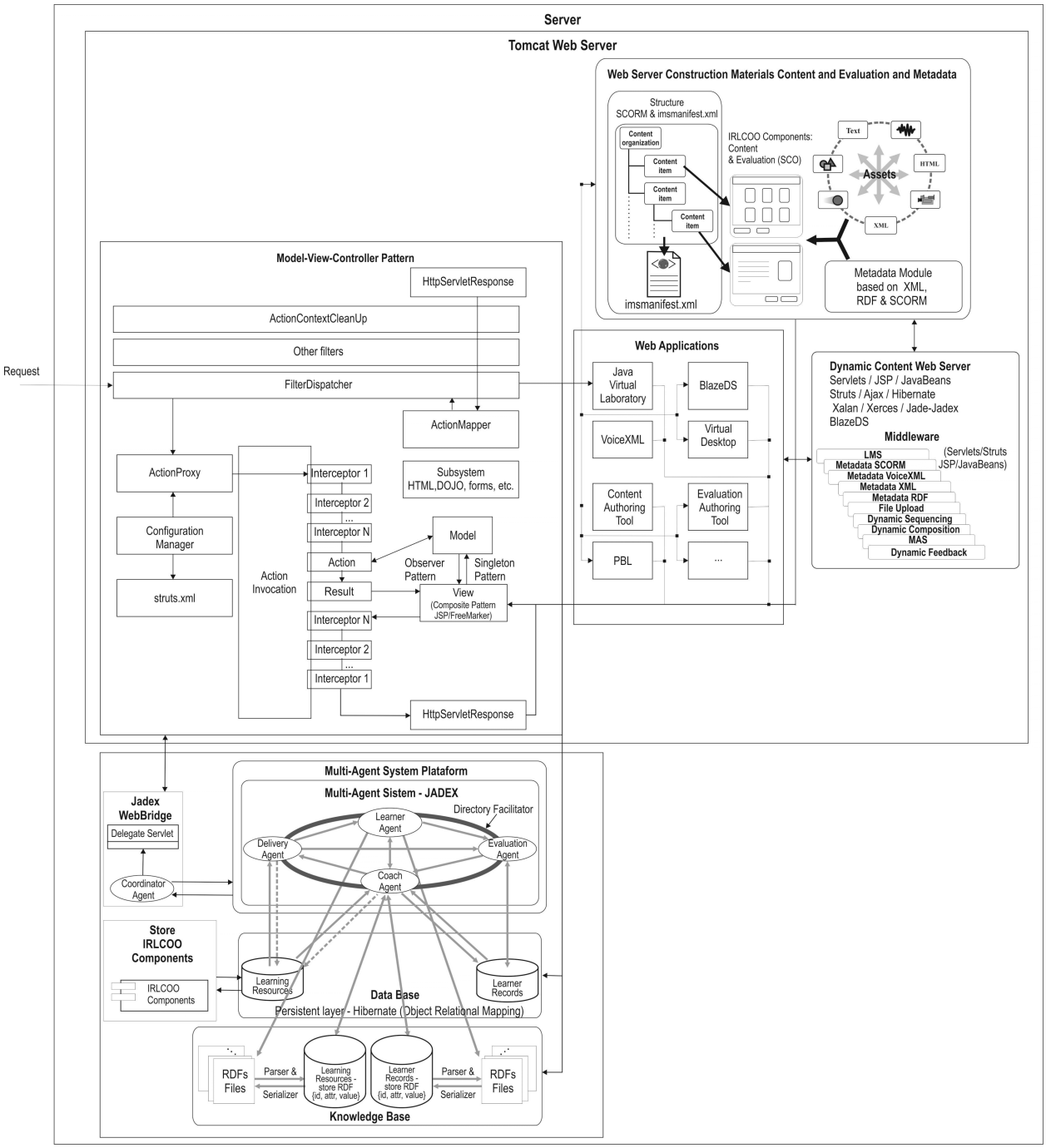

Figure 2. Architecture based design patterns on the server side.

The Figure 3 shows on the top IRLCOO components respectively of content and evaluation of our proposal, and on the bottom components respectively of content and evaluation for Moodle [19]. Moodle was used for comparison because it is one of the most popular tools to develop educational materials. The first hypothesis was tested with the following results:

1. The first significant difference when comparing our proposal is that educational materials are of type RIA.
2. The second significant difference is that our proposal is that educational materials are based on SDP.
3. The third significant difference is that our proposal is that educational materials have support for reconfiguration, sequencing and dynamic feedback.
4. The fourth significant difference of our proposal is that educational materials have support for ADL-SCORM.
5. The fifth significant difference of our proposal is that educational materials have support of a MAS for reconfiguration, sequencing and dynamic feedback.
6. The sixth significant difference of our proposal is that educational materials have decoupled Knowledge Model from our application.

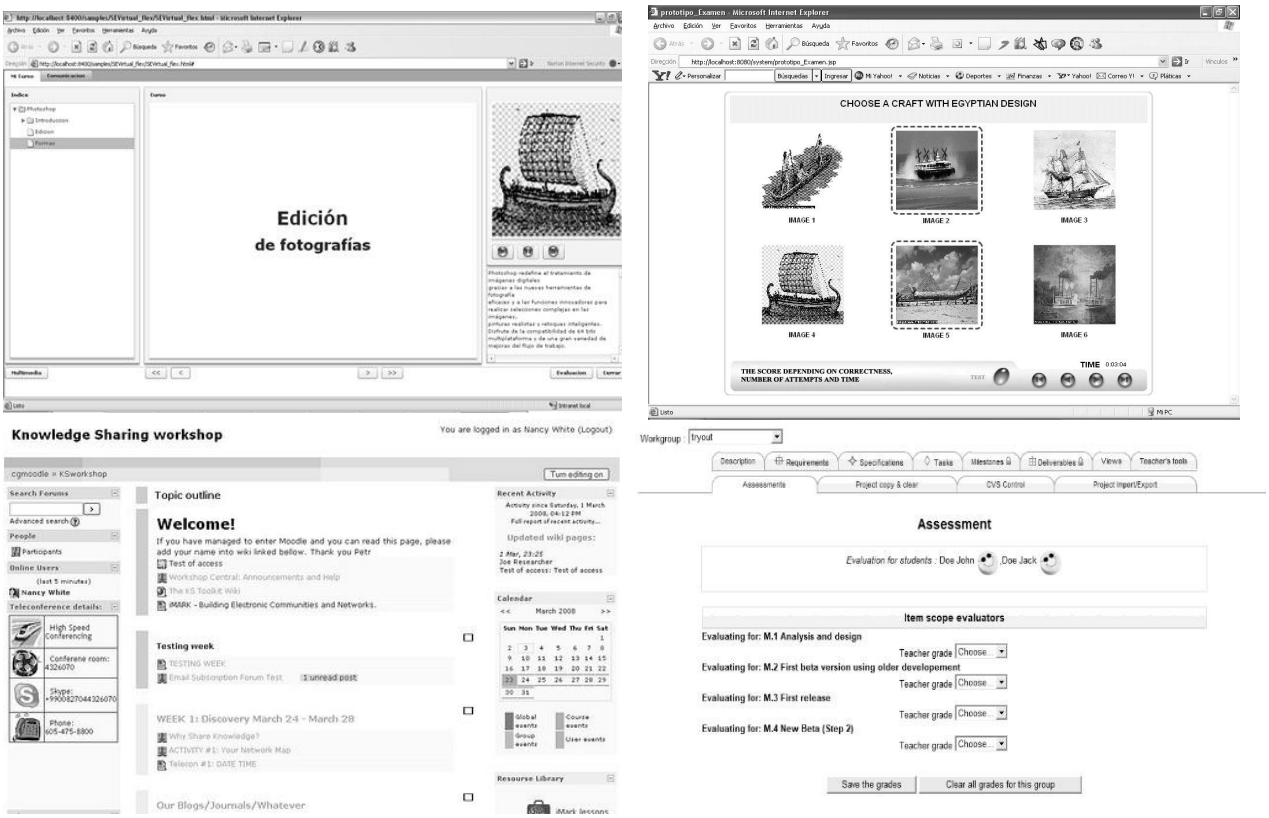

Figure 3. Comparison between our proposed educational materials and Moodle [14, 18].

CONCLUSIONS

Our proposal of educational materials has support for: rich multimedia and interactivity, integrated multimedia players, and asynchronous communication channel type Asynchronous JavaScript And XML (AJAX).

The SDP allowed us to manage large and complex projects in a better way, providing solutions for common challenges, focusing on maintenance and change, providing us with a better architecture of our proposal.

Educational materials have a high degree of customization for: reconfiguration, sequencing and dynamic feedback, and support of a MAS.

Our proposal introduced a Knowledge Base, decoupling the Knowledge Model from our application, enabling us sharing and reuse between users and applications.

Based upon the results of our study, we concluded that is challenging the development of educational materials that exploit the bidirectionality of the Web. Most educational institutions are looking to have a Web presence with educational materials type eReading, wasting the benefits of personalization offered by Web.

Our proposal offers an innovative vision to address the challenges of developing advanced educational materials.

ACKNOWLEDGEMENTS

Authors of this paper would like to thank the Instituto Politécnico Nacional (IPN) and the Escuela Superior de Cómputo (ESCOM) for the support for this work within the project SIP-IPN: 20131560. Thanking all students who participated in the development of systems.

REFERENCES

1. IEEE 1484.11.2 Standard for Learning Technology – ECMAScript Application Programming Interface for Content to Runtime Services Communication (http://ltsc.ieee.org/)
2. IEEE 1484.12.1-2002 Learning Object Metadata Standard (http://www.ieee.org/)
3. IEEE Learning Technology Standards Committee (http://ltsc.ieee.org/)
4. Adobe Flash Builder 4.7 Premium (http://www.adobe.com/mx/products/flash-builder.html)
5. Adobe Flash runtimes | Adobe.com (http://www.adobe.com/products/flashruntimes.html)
6. Adobe AIR | Adobe AIR 3 | Deploy applications (http://www.adobe.com/mx/products/air.html
7. Animation software, multimedia programs | Adobe Flash Professional CC (http://www.adobe.com/mx/products/flash.html)
8. Apache Flex® (http://flex.apache.org/)
9. Flash Player | Adobe Flash Player 11 | Overview (http://www.adobe.com/mx/products/flashplayer.html)
10. JADE Agent Platform (http://jade.tilab.com/)
11. JADEX Agent Platform (http://jadex-agents.informatik.uni-hamburg.de/xwiki/bin/view/About/Overview)
12. JavaFX Developer Home (http://www.oracle.com/technetwork/es/java/javafx/overview/index.html)
13. JDOM (http://www.jdom.org/)
14. Knowledge Sharing Tools and Methods Toolkit (http://www.kstoolkit.org/Moodle)
15. Manduca, C.A. and Mogk, D.W. (2002). Using Data in Undergraduate Science Classrooms. Carleton College.
16. Marlino, M.R., Sumner, T.R. and Wright M.J. (2004). Geoscience Education and Cyberinfrastructure. *Report of a workshop sponsored by the National Science Foundation*. Boulder, CO, University Corporation for Atmospheric Research: 43.
17. Microsoft Silverlight (http://www.microsoft.com/silverlight/)
18.mod/techproject/view/screens/assessment-MoodleDocs (http://docs.moodle.org/24/en/mod/techproject/view/screens/assessment)
19. Moodle.org: open-source community-based tools for learning, MOODLE (http://moodle.org/)
20. Peredo, R., Balladares, L., & Sheremetov L. (2005). Development of Intelligent Reusable Learning Objects for Web-Based Education systems. *Expert Systems with Applications*, 28(2), 273-283.
21. Peredo, R., Canales A., Menchaca, M., & Peredo, I. (2011). Intelligent Web-based education system for adaptive learning. *Expert Systems with Applications*, 38(12), 14690–14702.
22. Peredo, R., Canales, A., Ramírez, A.N., & Caravantes, J.C. (2013). Personalized knowledge management in environments of web-based education. *International Journal of Information Technology & Decision Making*, 12(2), 277-307.
23. Roschelle, J. M., Pea, R. D., Hoadley, C. M., Gordon, D. N., & Means, B. M. (2000). Changing how and what children learn in school with computer-based technologies. Children and Computer Technology, 10(2), Fall/Winter, 76-101. Available at http://www.futureofchildren.org. Downloaded January 14, 2005.
24. Valdez , G. , M , M. , Foertsch , M. , Anderson , M. , Hawkes , M. and Raack , L., Computer-Based Technology and Learning: Evolving Uses and Expectations (North Central Regional Laboratory 2000).