Creating Interactivity inside eLearniXML Educative Platform

Habib M. Fardoun[1], Iyad Katib[2] and Lorenzo Carretero Gonzaléz[1]

[1]Information Systems Department, Faculty of Computing and Information Technology,
King Abdulaziz University, Jeddah, Saudi Arabia
[2]Computer Science Department, Faculty of Computing and Information Technology,
King Abdulaziz University, Jeddah, Saudi Arabia

Abstract. Nowadays, the speed of response and interaction are very important factors for most applications. In addition, institutions, educational or not, have less Internet bandwidth due to the fact that they have to filter major information quantity. In this paper, we present the design and development of a module for the eLearniXML platform; this consists of its communication with the server, and the use of technologies such as HTML5 and JavaScript in order to improve the inter-action between the user and the application. Thus, an improved user experience is obtained with greater flexibility and efficient communication.

1 Introduction

The eLearniXML [1] platform is anchored in new methodological models associated with the activities conducted within a determined classroom environment. By using eLearniXML, students can access major resources repositories and instantly interact with them leading to improved teaching and learning experience. This is because in eLearning, teaching and learning heavily depends on underlaying technologies, and on the network connections available in the specific institution, which in some cases, are not the optimal ones [2].

Therefore, we have to take into consideration that the available Internet bandwidth in institutions, in many cases, cannot satisfy the demanded quality. Part of this is because there is a need for filtering information, by-passed from any type of data towards its destination inside that institution [3]. In some cases, this is because the technologies used for any kind of resident application consequently produce a delay when they have to provide the requested information to a user.

To alleviate such difficulty, we present a new module to include within the eLearniXML platform. This module improves user interaction (teacher or student) of the application to provide efficient management and communications within the system. The coherent system, in which the teaching and learning process is the users' focused point, prohibits side problems derived by bad usability or any bad synchronization and update overall system mechanisms.
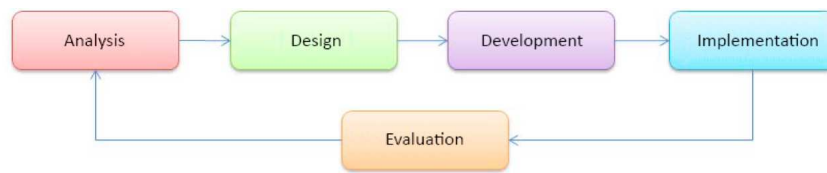
Fig. 1. ADDIE Module.

2 State of the Art

The actual needs are determined by the use of mobile devices and Internet resources. Consequently, and due to the great quantity of devices and different operation systems, people work over solutions that can adapt to any kind of devices and existent operative systems. With the inclusion of HTML5 technology [4], which is under the W3C (*World Wide Web Consortium*) supervision, a new wide variety of possibilities are opened as the creation of the multiplatform Web applications becomes easier to reach. In addition, it is anchored in HTML (*HyperText Markup Language*) standard language, the most commonly used in existing Web pages facilitating the developer's adaptation work. Moreover, technology provides students with the possibility to interact with the Web as a desktop application using the "canvas" tag, which supports increasingly great performance by the whole system, providing the capacity to interact with diverse elements such as graphics.

Therefore, it is of great interest to improve the user experience. It is also noted that the best potential is obtained with a flexible programming language like JavaScript [5]. This programming language is based on the instruction execution on the client side, which activates processing by the destination host and doesn't produce additional burden on the traffic in order to reach the server. Moreover, the obtained practical results are also of great quality and supports user's high experience due to increased speed. But as in any actual application functioning within a network, we need an exchange of information with the server. In order to avoid such exchange to be carried out with an indiscriminate increase of traffic on the net, the object XMLHttpRequest [6] is used to mediate the communication between the client side and the server side through HTTP protocol requests [7]. This communication can be synchronous or asynchronous, elevating the number of possibilities while designing a determined interface. In the next sections, we focus on the ways the new module is included inside the eLearniXML methodology and its main features.

3 ELearniXML Educative System Architecture

Based on the proposed structure by Fardoun, et al. in [1], we develop the new module according to the ADDIE methodological model used for eLearniXML platform as shown in Fig. 1. In this work, our intention is not to make changes in the initial analysis module. Rather, we concretely focus on the design part where the proposed module elements in this paper are established.

In this way, we take the eLearning system architecture as reference to present the
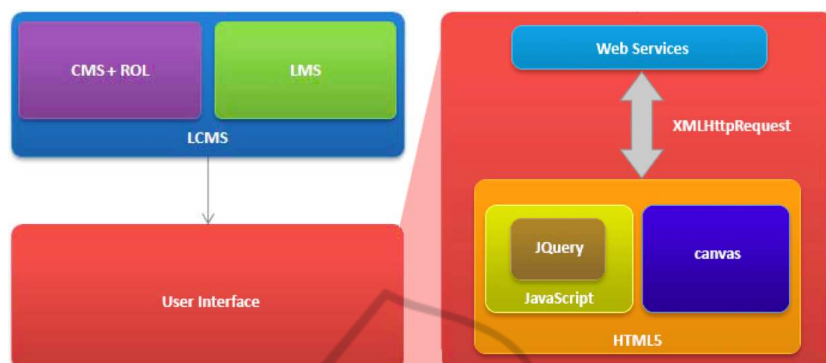
Fig. 2. Structure of the interactivity model within the eLearniXML platform.

part related to user interaction with the system. This functionality is on the user interface providing optimal performance in any type of environment. As in Fig. 2, the user interface is divided into various modules with concrete functionalities inside the platform. The HTML5 module is used for information visualization related to every activity inside the platform. The language features provide the system with the capacity to run under any actual device. With the "canvas" tag the contents are much more attractive and functional, also giving the application a better appearance and thus, improving user experience. The JavaScript code provides the system with the necessary interactivity flexibly without suffering lacks and gaps of time due to communication problems with the server; this is because by definition, this programming language runs at the client side, therefore improving user experience. On the top of that, JQuery library [8] facilitates the developer's work with the activity models, thus content creation and communication become easier.

The *Web Services component* provides the system with the functionality of content visualization to get the information associated to each user and their interactions with the activities. In addition, it provides great flexibility to make changes in the interface layer closer to the user. Thus, we can create different types of contents using diverse technologies. However, the functionalities of those presented here and the communication model determines the efficiency and final deployment. In this work, the communication between the platform Web Services and the HTML5 module is conducted by `XMLHttpRequest` requests to provide the possibility of a synchronous or asynchronous communication and, consequently, provided the good practices; it uses small quantities of Internet bandwidth. In Section 4 we present more details about its structure and how to use it.

4 Communication Model within ELearniXML

The `XMLHttpRequest` object request consists of an important part of the AJAX technology used to obtain server data without reloading the Web page. Therefore, usability, interactivity and speed are enhanced and increased. This object provides access to HTTP and HTTPS (Secured) requests through the client side code. Almost all current

browsers support this object either directly or indirectly. For instance, the following code is used for creating an `XMLHttpRequest` object and is accepted by Firefox 1+, Opera 8+, Safari 1.2+ and Internet Explorer 7+ browsers.

```
var xhr = new XMLHttpRequest();
```

However, for versions 5, 5.5 and 6 of Internet Explorer, the `XMLHttpRequest` object must be instanced by an ActiveXObject constructor as follows:

```
var xhr=new ActiveXObject("Microsoft.XMLHTTP");
```

Once we initiate this object on any browser, we will need the code related to the communication model to incorporate it into the eLearniXML platform. For that, we must establish a connection (open) to send the data (send). In the following method, used as an example of sending HTTP type of data: POST, GET, HEAD, 'url' is the address of the resource to establish the link, and 'async' is used to select an asynchronous communication (true) or a synchronous communication (false).

```
xhr.open(method,url,async[,username,password]);
xhr.send(data);
var r = xhr.responseText;
```

Once the connection is established and the data is sent, the variable 'r' will receive the requested information. This example would work correctly only if the communication is synchronous. If we request an asynchronous communication, we should also establish an event handler to control the returned information. In the following code, a function is established for the application to be ready instantly when the `XMLHttpRequest` object state changes. Therefore, the request is correctly establish through the HTTP protocol if (value 4) and its state is successful (200).

```
xhr.open(method,url,async[,username,password]);
xhr.send(data);
xhr.onreadystatechange= function(){handleResponse(xhr);};

function handleResponse(xhr)
{
    if (xhr.readyState==4 && xhr.status==200)
    {
        var r = xhr.responseText;
    }
}
```

The user only sees the various activities by information visualization and the possibility to save the information derived from conducted changes. For that, sending saved data is realized asynchronously which allows for example the teachers and students to continue with their teaching and learning process. Furthermore, a mechanism is established to ensure that the system does not resend the information if the user has not performed any changes. Hence, we avoid any unnecessary increment on the data traffic which leads to burdening the Internet bandwidth.
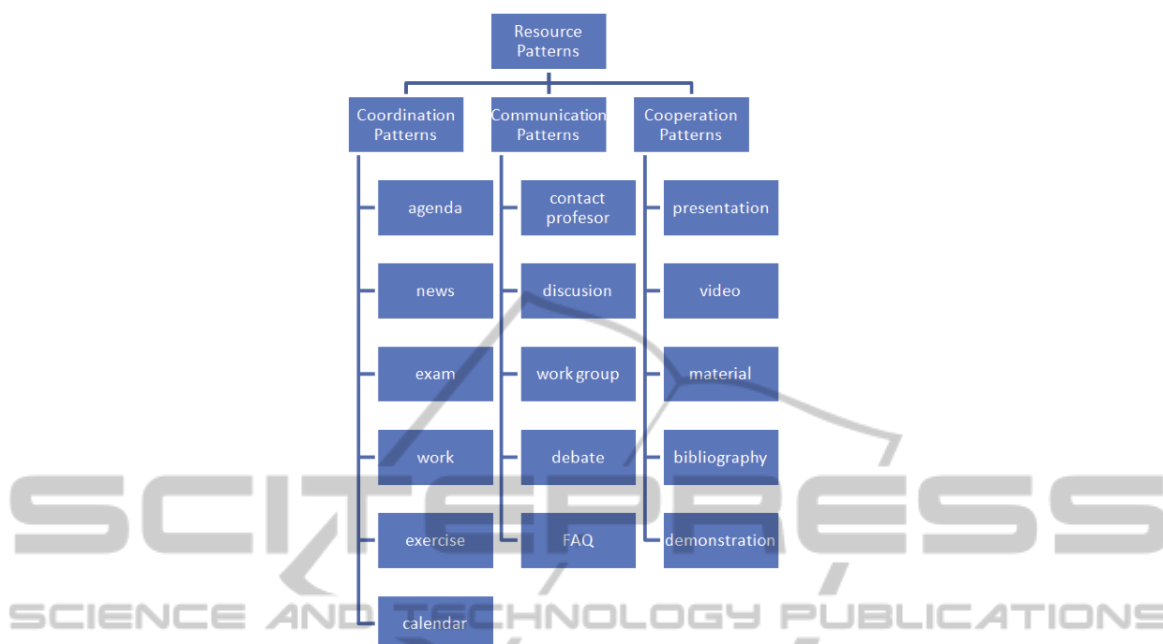
Fig. 3. Activity patterns inside eLearniXML platform.

5 How to Use the System

Given the resource patterns, shown in Fig. 3, and based on an existing plan developed by the teacher so in a determined moment in time the teacher can establish the learning activities and resources shown to his/her students during a teaching lesson as for example, video, bibliography and discussion.

At the same time, and within student's user interface, the proposed activities instantly appear. The student can interact with them as well as with the given environment so to adapt them to own needs. The diagram related to the model running in that period of time, is shown in Fig. 4.

Fig. 4 shows the ways each of the patterns comes determined by a set of elements inside the HTML5 document. These elements use JavaScript functions to increase the interactivity inside the platform. In addition, that code provides the needed communication with the rest of the system by the `XMLHttpRequest` object and Web services. As each of the patterns comes determined for a set of elements tangibles inside the platform, the content creation becomes much easier. The user who manages the activities only will have to drag the abstract component and automatically (in that moment) it is generated with the language that the browser can interpret. If a Web page is created, that Web page will be totally dynamic where the addition or subtraction of components will immediately modify its content, but this process will be transparent to the user who manages the contents or uses them. That carries that the modifications of each pattern become invisible to its creator (teacher) providing a more robust platform to use.
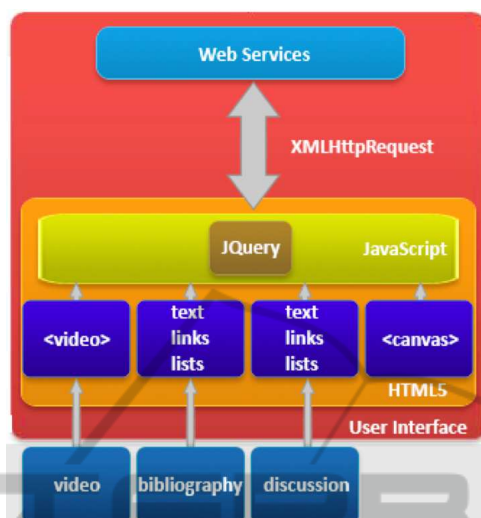
Fig. 4. Model for activities that use video, bibliography and discussion patterns.

6 Conclusions and Future Work

This paper is based on the eLearniXML platform and on the different models and patterns used to manage an eLearning platform. Advancing the research work realized for this project, a module for interactivity and communication with the student's user interface was created. In this module, we use current leading technologies related to usability, speed and multiplatform capacity, enhancing user experience that promotes the interaction with the system and facilitate the teaching and learning process. Currently, we are working on the proposed module improvement, perfecting details and adapting the platform so that the user has in hand what is needed to interact with the different eLearniXML patterns. We are also taking into account the following system features:

– Auto-save: Such system functionality to keep user's work safe with special attention needed to sustain the system when traffic increases.
– Familiarity: Including and adapting each pattern for the needed elements and modules for the users to interact by familiar ways (like activities of Facebook and Twitter).

References

1. Fardoun H. M.: eLearniXML: Towards a Model-Based Approach for the Development of E-Learning Systems. University of Castilla-La Mancha. (2011)
2. Cipres A. P., Fardoun H. M., Mashat A.: Cataloging Teaching Units: Resources, Evaluation and Collaboration. In: FedCSIS, Wroclaw, Poland (2012) 825–830
3. Recker, M. M., Walker, A., Lawless, K.: What do you recommend? Implementation and analyses of collaborative information filtering of web resources for education. In: Structional Science: Vol. 31(4-5),(2003) 299–316

4. R. Berjon, T. Leithead, E. Doyle Navara, E. O'Connor, S. Pfeiffer: HTML5, W3C, http://www.w3.org/TR/html5/

5. JavaScript: The Definitive Guide. David Flanagan 2011. O'Reilly Media. ISBN 978-0-596-80552-4.

6. The Complete Reference. Thomas A. Powell 2008. McGraw-Hill. Chapter 3.

7. HTTP: The Definitive Guide. David Gourley, Brian Totty, Marjorie Sayer,Anshu Aggarwal, Sailu Reddy. 2002. O'Reilly Media. ISBN 978-1-56592-509-0.

8. jQuery CookBook. Cody Lindley. 2009. O'Reilly Media. ISBN 978-0-596-15977-1.