HYBRID WEB- AND MOBILE-BASED

E-learning with Rich Media Support

Bertalan Forstner, Imre Kelényi, Gergely Csúcs, Dr. Hassan Charaf
{bertalan.forstner, imre.kelenyi, gergely.csucs, hassan.charaf}@aut.bme.hu

ABSTRACT

With the rapid evolution of mobile devices a new medium for e-learning technologies arises. These modern devices are capable of displaying an impressive amount of text with color pictures and good quality video. They can communicate with the e-learning servers over a high-speed third generation mobile network, which enables distance learning with rather low downloading time. The operating systems on these devices support a wide range of input widgets that are adequate for online questionnaire tests. With the selection of appropriate frameworks and architecture a flexible hybrid e-learning system can be built up.

INTRODUCTION

The rapid evolution in information technology and many other scientific fields makes the design and operation of a corporate knowledge base and e-learning system inevitable. The spread of the so-called smartphones can be a handy extension for these systems for the following reasons. First, the mobility can imply the quick access and mapping of the knowledge stored on central servers. Second, an ergonomic thick client can be installed on these devices. The mobility, that is, the m-learning can be used to harness the time spent on traveling, waiting, and so on.

The system should make it possible for the user to select whether to use the web-based client or the mobile client, depending on his or her context. In order to design such a system, we decided against the following minimum capabilities for the hybrid e-learning framework.

The framework is capable to organize different courses for different schools on the same server through a web application. The course organizers should assign teachers and students to course classes, where the teachers can supervise the progress of the participants. The course material should contain different kinds of multimedia elements, such as formatted text, pictures, videos, attached documents and so on, most of them should be converted automatically into a format that can be consumed by mobile clients. The progress of the students should be monitored by questionnaires. The questions will be selected from a question pool and all questionnaires will have their time limitation. The results will be displayed in diagrams from different aspects. The rest of this paper is organized as follows. The next chapter describes the related work on this area. A short description of the modern smartphones is followed by the introduction of the frameworks that are utilized in the hybrid system. After the presentation of the designed architecture and modules a short conclusion summarizes our results.

RELATED WORK

The number of currently available e-learning systems is extensive [15]. These can be categorized as follows.

Content development tools include software applications responsible for converting the learning resources into an electronic form. These can be simple, text-based solutions with only basic formatting capabilities, or more advanced tools which enable the authors to create documents with rich media content. Offline course builders, e-book creators, form builders and simulation tools are some examples for the more specialized applications, but almost any kind of graphic designer software, web page builder or presentation creator tool can be used for content development. More sophisticated solutions like podcasting software, blog creators and RSS frameworks should also be noted.

The common in the above listed tools is that although they are not full-service e-learning solutions, they still play an important role in the planned system as the means of creation for the e-learning content. Our application framework will be capable of processing the output of these tools.

Communication and cooperation systems focus on the exchange of information between multiple participants. They differ mostly in connection characteristics (synchronous or asynchronous) and in the number of participants (two or more).

Instead of creating the content, these software applications help to transfer it between the e-learning provider and the students. Instant messaging and chat software, e-mail based tools and remote assistance applications can be put into this category, but we also examined web-based solutions such as forums, discussion boards, webcasting tools and wiki systems too.

The base concept of our framework is an asynchronous, one-to-many communication based system, which supports multiple data sources and is capable of two-way communication. We do not except the new system to support working in groups or one-to-one communication as there are proper solutions for these objectives.

Course and content management systems combine the elements of the previous two categories while focusing on a specific field. This group includes enterprise training software, academic applications and custom solutions.

Mobile learning software systems are simple, database- or text-based solutions where the content is pre-stored on the mobile device itself. These applications enable the user to study on the road or while being away from a desktop computer, but we should not disregard the drawbacks of the simplified user interface which is a general disadvantage of any mobile solution. Our goal, after thoroughly examining the currently available systems, is to develop a mobile application suite that can process the content stored on the central e-learning server and display it in a feasible way on the small screen.

Hardware-based solutions like whiteboards, projectors and video conference systems are mainly used in distance learning courses and are not addressed in this study.

As a result of briefly examining numerous e-learning functions and solutions we can summarize that our application framework will be capable of processing the output of different content development tools and forward it to both mobile- and web-based clients. It will rely on an asynchronous, one-to-many based communication model and the system will also support a mechanism for querying the participating students.

M-learning on smartphones

In the recent years the proliferation of Symbian-based smartphones can be observed. Symbian [14] is an operating system designed specially for intelligent mobile devices. There are several versions of Symbian OS in the market running on different smartphones. The Symbian v9 is the latest of all. Symbian provides a robust architecture and API to support development. The popularity of the operating system is owed to its wide functionality, the numerous devices built on it and the thousands of available applications.

As Symbian is such popular mobile operating system, we decided to develop the mobile thick client on that OS. The phones that are built on Symbian have different user interface layers, from which the most widespread is the Series60 (or S60) UI. It specifies a moderate sized screen which is convenient for displaying forms or small size videos. However, it has only limited input capabilities (ITU-T keypad) that should be taken into account when designing the questionnaires. As programming for Symbian OS is very complicated, we should consider using a class library.

Supporting frameworks

As described in the introduction, our hybrid solution will rely on a web application that can serve the requests of both web and mobile-based clients. The architecture will be detailed in the next section. As we want to implement flexible software we will rely on different frameworks. They can help us to exploit the capabilities of the used technologies as well as speed up the development process and guarantee a minimum quality. To give a good understanding of the system, in this section we will describe both the web application framework and the mobile Simplian class library that are thoroughly used to ease the implementation.

The cPEED web application framework

Nowadays using the Internet becomes an everyday activity for businesses and individuals. Even the medium-sized and the smaller businesses are running dynamic web pages or applications. They are often employing badly trained and inexperienced programmers using the easy-to-use script languages, in an ad-hoc manner, usually without any preceding design, often patching the web site with new elements. [4] This leads to impenetrable and hardly upgradeable code, and this way is very inefficient. As [5] showed us, if the project team totally ignores the software engineering part, they are really making a mistake and the result may haunt them or their replacements during the maintenance, code reuse and upgrading procedure. The solution to this problem could be the use of a framework over a popular script language that will give complete solution for the regularly emerging web-specific problems, speeds up and gives a concrete structure for the programming, and helps creating efficient applications. Based on this we can build a development tool, a code generator to make the developing process even faster. The prepared modules can then in one block transferred to new projects, without restrictions to the look, language, used database or any other aspect.

The used script language will be the PHP because of its popularity [2].

The web applications that are built on the framework have layers that are suited to the multi-tier architecture that is well known from the traditional software development [3]. The layers are defined according to the developer role separation and software logic distinction. If wide-spread and general web standards for the layers or for the interface between them were to be found, they were used in the framework. In the next paragraphs some examples for these standards will be mentioned.

As it is represented in the following figure, there is a presentation part of the web applications. This layer contains only HTML template files that are provided by creative designers. In our e-learning solution the Smarty Template Engine was used. These files do not contain any program code at all, so the designers will not be messed up, and it is easy to change the look of a web application.

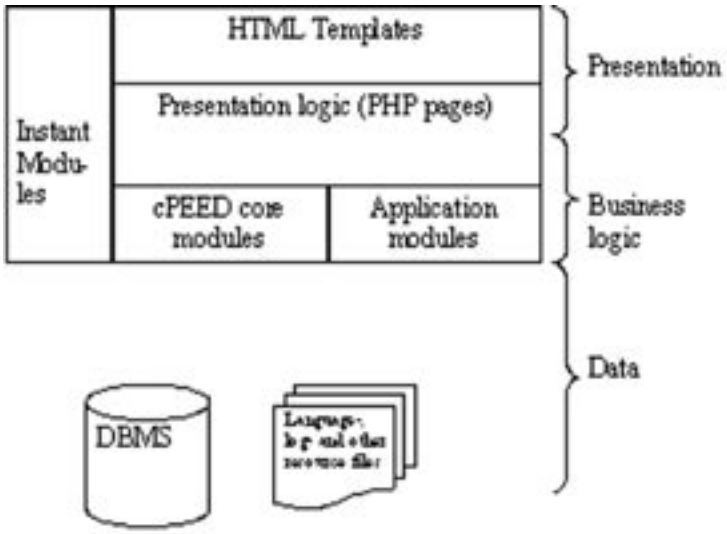

Figure 1
Web Application Architecture

The next layer is the presentation logic. Practically these are the script files that are requested by the browser or mobile client. In the framework these files are divided into sections, each of them containing program code

for a defined type of task. For example, there are sections for business class instantiation, input argument validation, form definition, validation and event handling, data binding and so on. This separation helps debugging and code generation.

Presentation logic utilizes the template engine to display the page based on the corresponding template. It does not contain business logic at all; it only uses the services of the next layer, the modules. The tasks in the presentation logic layer take the most time of the web application development, however, with a well-defined framework given, the most cases they could be generated with a code generator.

The layer of modules is divided into two parts. The core modules contain the services that are needed on almost every page. They cover the difficulties and peculiarities deriving from the web architecture.

An enumeration of them can be found in the next section.

The other part, application modules, contains the business logic of the application. These modules assure functionalities through well-defined interfaces for the presentation logic. Only this layer can access resource files and databases through a data abstraction layer.

The Simplian class library

Currently, Symbian is one of the most popular mobile platforms. However, Symbian-based software development is far more difficult and requires more specific skills than the development of desktop applications. This stems not only from the C++ characteristics, but also from the absence of easy-to-use integrated development tools.

We can mention, for example, the complicated memory management (for instance cleanup stack and two-phase object construction), the special exception-handling (leave-mechanism), string- and array handling, or the unique aspects of resource-management. Also, developers must strictly take care of these uncommon circumstances, since ignoring them could lead to bugs that are hard to reveal.

A class library can help the programmers by covering all the recurring tasks deriving from the mentioned facts. The Simplian Framework [12] provides a simple API for constructing the user interface of the application, and other simple means to bind data to the widgets, or send and receive them through different communication channels. Simplian also contains tools to generate the C++ code from a well-defined, platform-independent XML file, which can be constructed by the modeling tool.

The GUI widgets are usually divided into input and output elements. The classes for user input simplify the use of dialogs as described in Section 2.1. It is possible to create multipage dialogs with input widgets very simply. An elaborated new mechanism checks the user input against the required circumstances. As Series60 devices have a rather small screen, the different commands are assigned to menus, which can be activated with hardware softkeys by the user. The class library for the input widgets implements a state machine that is triggered by event handlers. However, the transition structure can also be reconfigured at runtime. The source code is easy to read and understand. Therefore, code maintenance is also simpler. The class library is prepared to work with user interfaces other than Series60, hence, the architecture is open to future development.

System architecture and modules

In this section we give an overview of our proposed system architecture. The general structure is as follows.

The core element of our hybrid e-learning framework is the main web application deployed on the central server. The web application is supported by a relational database which can be reached through a database-abstraction layer. Presentation logic for web-based clients communicates directly with the business logic. In case of mobile clients, client-server communication is mediated by a dispatcher layer, which enables the use of XML data over HTTP for the clients. The dispatcher can also accommodate other connections requiring XML over HTTP (e.g. web-service extensions for the server).

The following figure summarizes the proposed architecture.

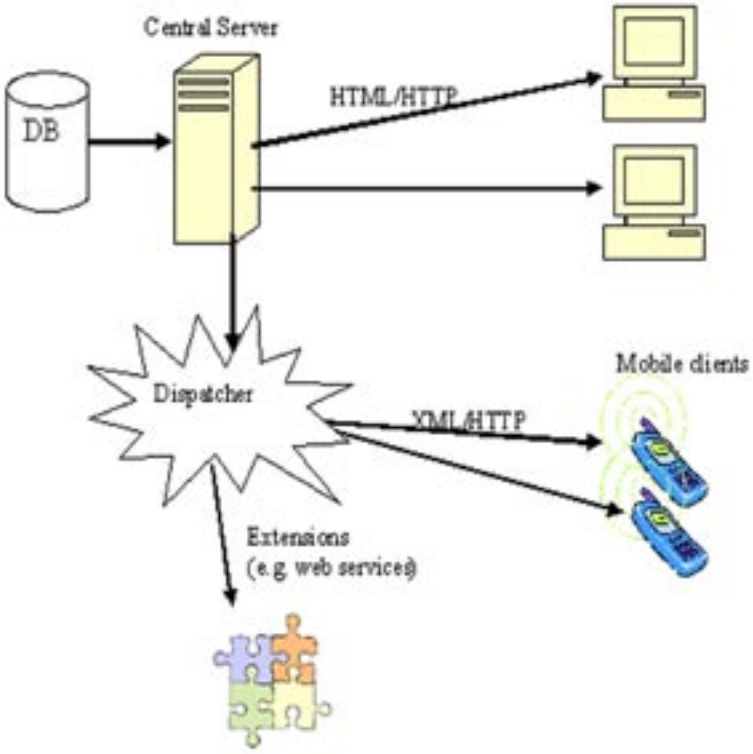

Figure 2. System architecture

Web-based clients connect to the e-learning system using ordinary browser applications. Capabilities of the various available browsers limit the range of deliverable content, the limitations should be considered on content generation for a pre-defined target audience.

Client application for smartphones employs online authentication and per-course authorization. The Simplian class library is capable of building the actual views dynamically, thus layout for presentations, forms are generated on-the-fly, based on XML messages, and the screens should look pretty on all devices supported by Simplian. Mobile users are also enabled to store course materials locally, although this feature is limited by the available storage capacity on the device.

Multimedia

By design, the e-learning system should support delivering most types of content.

Textual information is stored in formatted manner. The chosen format is HTML, since it naturally fits for browsers on desktop PCs, and it is also supported by the targeted mobile devices. The administrative web-client supplied for course-owners features a visual editor supporting editing, formatting of HTML documents, including the insertion of arbitrary number of pictures for visualization purposes.

Due to popular 3rd party applications, and its numerous built-in export converters, documents made with Microsoft Office can be tailored to course material fairly easy. In addition, the javascript application supplied for creating and storing materials preserves formatting of text copied from a Word-document. PowerPoint presentations can also be exported and uploaded as pictures, however these images fit badly for the screens of mobile devices when they actually get delivered.

Attaching of arbitrary number of supplementary materials is supported for any given course page.

Viewer applications for the most widespread file formats (Microsoft Office, Adobe Portable Document Format) are factory-installed on the most modern smartphones.

Attaching of multimedia materials are to be supported via several methods. Since browsers do not have necessarily native support for playing video files, a web-standard with higher availability should be found for maintaining automatic handling of video materials. According to our current understanding, Macromedia scripting-tools (Flash or Shockwave) could be employed for developing a video-player. Other viable approach is finding an application for automatic conversion of uploaded video materials to 3GP format, and serving them to the devices.

According the flexible design of the e-learning framework, the system is open for extending with support for new content-formats required by specific customization tasks or general development.

Anyway, in the current version of our e-learning system, it is a priority task for course administrators to identify the client platform for materials (web, smartphone), and to optimize the content for efficient delivery. It is feasible to survey or estimate the average equipment (display sizes, resolutions) to be expected to view the given course material on. When no prior information is available, compatibility with the widest possible range of devices should be aimed.

CONCLUSION

Fulfilling our prior expectations, we successfully identified the specific functional and architectural requirements for both web-based and mobile distance-learning systems. Learning from previous existing e-learning systems, a framework architecture had been proposed. Functional plans were developed, and also a relational database for supporting them. After taking these steps, we also made prototype user interfaces for various device classes. Finally, summarizing the actual requirements, our need for supporting possible future extensions, and the feasibility issues for deployment, we selected the target platforms for implementing the web-based and the mobile client applications.

ACKNOWLEDGEMENTS

The fund of “MobEduNet project” has supported in part, the activities described in this paper.

REFERENCES

[1] Andrew McDonald, Ray Welland, Web Engineering in Practice (Proceedings of the Fourth WWW10 Workshop on Web Engineering, 1 May 2001.) 21-30.

[2] Rasmus Lerdorf, PHP pocket reference (OReilly and Associates, 2002).

[3] Tobias Ratschiller, Till Gerken, Web application development with PHP 4.0 (New Riders Publishing, 2000)

[4] Andrew McDonald, Ray Welland, Agile Web Engineering (AWE) Process: Multidisciplinary Stake holders and Team Communication (J. M. Cueva Lovelle et al. (Eds.): Third International Conference on Web Engineering, ICWE 2003, LNCS 2722, July 2003. ISBN: 3-540-40522-4.) 515-518.

[5] R. S. Pressman, Can Internet-Based Applications Be Engineered? (IEEE Software, Volume 15(5), September/October 1998,) 104-110.

[6] Bertalan Forstner, Hassan Charaf PhD, cPEED: A Rapid Webapplication Development Framework (IASTED International Conference on Parallel And Distributed Computing And Networks)

[7] Kristof Aczél, Hassan Charaf, Automatic User Interface Code Generation In Symbian. MicroCAD International Conference, 2005, Miskolc.

[8] Bence Kővári, Hassan Charaf: User Interface Modeling for Mobil Devices. MicroCAD International Conference, 2005, Miskolc.

[9] Róbert Kereskényi, Bertalan Forstner, Több protokollt támogató kommunikációs komponens megvalósítása Symbian Series 60 platformon. TDK Conference, Budapesti Műszaki és Gazdaságtu dományi Egyetem, 2005.

[10] G. Mezei, T. Levendovszky, L. Lengyel, H. Charaf, A Flexible Attribute Instantiation Technique for Visual Languages, IASTED on SE, February 15-17, 2005, Innsbruck, Austria, pp. 355-359.

[11] L. Lengyel, T. Levendovszky, G. Mezei, B. Forstner, H. Charaf, Metamodel-Based Model Transformation with Aspect-Oriented Constraints, International Workshop on Graph and Model Transformation (GraMoT), Tallinn, Estonia, September 28, 2005.

[12] B. Forstner, L. Lengyel, I. Kelényi, T. Levendovszky, H. Charaf, Supporting Rapid Application Development on Symbian Platform, EUROCON 2005 International Conference on "Computer as a tool", Proceedings of the IEEE, Belgrade, Serbia & Montenegro, November 21-24, 2005, pp. 72-75.

[13] L. Lengyel, T. Levendovszky, G. Mezei, B. Forstner, H. Charaf, Towards a Model-Based Unification of Mobile Platforms, 4th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-06), Dubai, Sharjah, UEA, March 8-11,

[14] Leigh Edwards et al: Developing Series 60 Applications : A Guide for Symbian OS C++ Developers. Addison-Wesley, 2004

[15] Wendy Brunner, John Heinstein, E-Learning As You Like It: From Print to Desktop to PDA Using One Set of Content, The eLearning Developers' Journal, 12/12/2005