A pattern-oriented and web-based architecture to support mobile learning software development

Manouchehr ZadahmadJafarlou\textsuperscript{a*}, Bahman Arasteh\textsuperscript{b}, Parisa YousefzadehFard\textsuperscript{c}

\textsuperscript{a}Department of Computer, Ilkhchi Branch, Islamic Azad University, Ilkhchi, Iran
\textsuperscript{b}Department of Computer, Tabriz Branch, Islamic Azad University, Tabriz, Iran
\textsuperscript{c}Young Researchers Club, Department of Computer, Ilkhchi Branch, Islamic Azad University, Ilkhchi, Iran

Abstract

With the rapid development Wireless/Handheld (W/H) technologies, mobile learning has become a new trend in education. In spite of abundant tools and literature about mobile educational software, the lack of a web-based mobile architecture is felt. This study uses design patterns to present platform independent architecture to support development of m-learning systems for greater flexibility, agile and accurate web based software development. The paper also includes an Ajax and Decorator pattern-based method to develop robust and more interactive presentation layer. Qualitative evaluations of proposed architecture illustrate the optimization on aesthetic, content, navigation and other presentation-related metrics.

\textsuperscript{a*} Manouchehr ZadahmadJafarlou. Tel.: +984123326064; fax: +984123326063. E-mail address: zadahmad@iauil.ac.ir.

Keywords: Design Pattern; m-learning; e-Learning; mobile networks; Service Oriented Architecture (SOA); Architecture Style; Model Driven Architecture (MDA); Platform Independent Model (PIM)

1. Introduction

The freedom to learn what, where, when and how one likes, availability, adapt learning content to the context the learner finds him- or herself in, and taking place both in the classroom and outdoor across formal and informal settings are the major advantages of Mobile learning (m-learning) [Fransen (2008)]. M-learning concerns the acquisition of knowledge through a mobile device and focuses on how society and its institutions can support an increasingly mobile population (macro-level), how mobile learning technology can be coupled with other forms of learning taking place in organizations and schools (meso-level), and aims for a clarification of the conditions necessary for m-learning to be successful for an individual or a group of learners (micro-level) [Sandberga, Maris & de Geusa (2011)]. The efficiency of a learning environment is determined by the adapted learning strategy used in the environment [Khalifa & Lam (2002)].

Much attention has been focused on new learning strategies with appropriate software tools and environments ([Chun, Hwangb, Tsaib & Tsenga (2010)], [Chen, Hsieh & Kinshuk (2008)], [Yeh, Chen, Hung & Hwang (2010)],...
These studies show that a sophisticated and well-designed presentation layer must be designed to increase efficiency of m-learning in zillion different Wireless/Hand held (W/H) devices by different screens sizes, different technologies and different computation power and resources.

In spite of W/H device popularity with students, not much has been done to extend e-learning to these devices [Motiwalla (2007)]. This study presents a web-based architecture using thin-client architecture and device independence of Wireless Access Protocols (WAP) and WML (Wireless Markup Language). The thin-client architecture allows applications to run on the server and the thin page is the only part of the application that is transferred to W/H devices, thereby removing the need for sophisticated client device.

The rest of this paper is organized as follows. Section 2 explains our classification of W/H devices from supporting educational services viewpoint. Section 3 illustrates the proposed pattern-oriented architecture to web-based mobile learning software. Section 4 contains qualitative evaluation. Paper is concluded with a conclusion.

### 2. Classifying of W/H devices from supporting educational services viewpoint

Mobile Learning software (ml-Software) has so interactive, network intensiveness, data driven, content sensitive, continuous evolution and aesthetics nature that require input and output components with formatted texts, high quality graphics, audio, video, and animations. But the main problem is inequality of potential of various W/H devices with internet access and web browsers in supporting ml-Software.

Classification of W/H devices related to computational resource of them. Any category of software that runs on these devices involved with resource management subjects. In [ZadahmadJafarlou, YoushefzadehFard, Arasteh (2011)] a pattern-oriented PIM for resource management of learning services in mobile networks are presented. In general there are four classifications of W/H devices -from supporting learning services input and output components and resource viewpoints- that have internet access capability and web browsers (Table 1).

<table>
<thead>
<tr>
<th>Classification of W/H devices by resource</th>
<th>Text Support</th>
<th>Image S.</th>
<th>Audio S.</th>
<th>Video/Animation</th>
</tr>
</thead>
<tbody>
<tr>
<td>minimum resources</td>
<td>Limited</td>
<td>Narrow and nearly zero</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>less than intermediate resources</td>
<td>Medium</td>
<td>Medium</td>
<td>limited</td>
<td>No</td>
</tr>
<tr>
<td>Intermediate resources</td>
<td>Good</td>
<td>More than Medium</td>
<td>Medium</td>
<td>Medium</td>
</tr>
<tr>
<td>High resources</td>
<td>High</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

When we compare this classification with PCs we infer that even the weakest PCs can support Intermediate class of this classification. Therefore view requirement of ml-software are completely different as it must adapt view layer with capabilities of various W/H devices and must utilize the maximum capacity of resources of these devices.

### 3. The proposed pattern-oriented architecture to web based mobile learning software

Web based ml-Software are complex enterprise applications that are composed of a large number of components across multiple levels of abstraction. The presentation layer of these applications is in relation with different W/H devices by different presentation capabilities, technologies, computation power and resources.

A pattern describes a recurring problem that occurs in a given context and recommends a solution. The solution is usually a collaboration between two or more classes, objects, services, processes, threads, components, or nodes that work together to resolve the problem identified in the pattern. The patterns are becoming a commodity tool in our treasure chest of software development techniques. Two patterns that provide a fundamental structural organization for interactive software systems are Model-View-Controller (MVC) and Presentation-Abstraction-Control (PAC) ([Buschmann, Henny, & Schmidt (2007)] and [Trowbridge, Mancini, Quick, Hohpe, Newkirk, Lavigne (2003)]). MVC includes three components: The model that contains of the core functionality and data, the view that manages the display of information and the controller that interprets the mouse and keyboard inputs from the user [Trowbridge and et al. (2003)]. PAC consists of three components: presentation, abstraction, and control.
PAC organizes the application as a hierarchy of decoupled agents. The downside of arrangement of the PAC is necessity of coordination between multiple agents. The objective of MVC or PAC is accomplishment of the interaction between presentation layer and functionalities of the application. MVC pattern is not as same as the Three-Layered Architectural style (TLA) while the presentation layer of the TLA contains of functionality responsible for managing user interaction with the system. Hence the presentation layer of the TLA is equal to the view and the controller parts of the MVC and model part of the MVC equal to business and data layers of the TLA.

3.1. Proposed Architecture

Our proposed architecture is a variation of the service-oriented TLA where main focus is on strengthening of the presentation layer of the application to undertake different display requirements of various W/H devices belong to different classifications. The user interface code tends to be more device-dependent than business logic and main responsibility of this layer in theses software is to publish educational materials to students in high quality manner. Hence the powerful user interface is a key element in success of ml-software. Decoupling of User Interface (UI) and the application’s core functionality allow independent creation of PIM of the UI and modification and transforming it to any PSM depends on chosen platform. Figure 1 shows the proposed architecture. The main focus of our research is on presentation layer of ml-software and components, and on patterns and relation between them to create more flexible, robust and education related mobile interaction environments.

Table 2 briefly describes the patterns and components that used in proposed architecture.

3.2. Using AJAX and Decorator based collections of template views and transform views in creating presentation layer based on classifications of W/H devices

Classification of W/H devices lets ml-software to detect capabilities of various connecting devices, and optimized resource and capability usage. In the proposed approach each educational Pages considered as a collection of educational contents. Each educational content consists of input and output components compound of text, image, audio, video, animation and others. When designing every educational content, input and output components are chosen from class four, then component named Automatic content transformer (ACT), transforms it to any other three classes. Each of these contents, using Decorator pattern, developed as a service and makes a template view. All classes of educational contents preserve main content, but W/H devices that belong to third and fourth class uses high quality input and output as well as Ajax foundational technology patterns, Ajax Programming Patterns and Ajax Functionality and Usability Patterns [Michael Mahemoff 2006] to create more interactive template views.
Table 2. Summary of used patterns and components

<table>
<thead>
<tr>
<th>Pattern/Component</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Educational_UI_Components</td>
<td>The presentation layer provides the ml-Software UI. Typically, this involves the use of WML for thin or fat client browser-based interaction based on Classification of W/H devices.</td>
</tr>
<tr>
<td>Educational_UI_Process_Components</td>
<td>Educational software is so interactive Complex UI, it often require many highly complex forms. We create a separate UI process component to encapsulate dependencies, validation, and navigation between components, pages and the logic associated with navigating between them. These UIP components are typically custom components that are based on design.</td>
</tr>
<tr>
<td>MVC [Fowler (2002)]</td>
<td>Solves coupling between layers by modularizes the mobile education UI functionality so that developers can easily modify the individual parts.</td>
</tr>
<tr>
<td>Page Controller [Fowler (2002)]</td>
<td>Best structures the controller for ml-Software so that developers can achieve reuse and flexibility while avoiding code duplication.</td>
</tr>
<tr>
<td>Front Controller [Fowler (2002)]</td>
<td>Centralizes common logic replicated in different views to reduce the amount of code duplication so that code duplication is avoided.</td>
</tr>
<tr>
<td>Application Controller [Fowler (2002)]</td>
<td>Encapsulates leading logic of a series of pages or following a specific workflow within a separate application controller. UI Process components use it to determine correct workflow and views. Separates the request for a service from its execution, treat with it as separate objects, and schedules their execution, and provides additional services such as logging and storing request objects for later undo/redo.</td>
</tr>
<tr>
<td>Command Processor [Buschmann, Meunier, Rohnert, Sommerlad &amp; Stal (1996)]</td>
<td>Observer [GoF95]</td>
</tr>
<tr>
<td>Transform View [Fowler (2002)]</td>
<td>Introduces a dedicated transform view component that converts data received from the application in response to specific user requests into concrete views onto the data.</td>
</tr>
<tr>
<td>Template View [Fowler (2002)]</td>
<td>Introduces a template view component for each view that renders application data or other information into a predefined view format using a specific user interface technology.</td>
</tr>
<tr>
<td>Helps protect an application from external attacks by introducing a proxy that inspects the payload of service requests to identify and remove suspicious content.</td>
<td></td>
</tr>
</tbody>
</table>

Template views of all classes have been assessed by ACT from aesthetic viewpoint (such as the word count, header count, font style, text positioning count, link count, page size, graphic percentage of page, graphic count, color count, font count), content viewpoint (such as page wait, page complexity, graphic complexity, audio complexity, video complexity, animation complexity) and navigation viewpoint (such as number of links per page (Page-linking complexity), Number of links (Connectivity), average of links per page (Connectivity density)). Afterwards ACT restructure template views to expose best quality for different kind of W/H devices (for example by dividing one template view of forth classes to two or more linked template views of lower classes).

In addition to educational content, template views also includes UI components like page borders, scrolls, background and others developed with service-oriented decorator. Template views developed for forth classification transformed automatically to other three classes as service-oriented template views. When a W/H device connects to ml-software, its classification was detected and related template views or transform views was used. Implementation of template views follow pattern-based MDA process ([Zadahmad Jafarlou, Moeini, & Yousefzadeh Fard (2010)] and [Zadahmad Jafarlou & Yousefzadeh Fard (2010)]) that use patterns combination to modeling system and transformation between different models of MDA. Quantitative and qualitative evaluations of it illustrate the ease of reuse, accurate automation, and large granular transformation among modeling levels of MDA.

4. Qualitative evaluation of the presentation layer of the proposed architecture

A useful set of UI metrics can assess the presentation layer of proposed architecture and provide insight into the usability, aesthetics appropriateness, content design, and efficiency of navigation and component design of the UI. [Ivory, Sinha & Hearst (2001)] propose a useful set of measures for estimating the impact of aesthetic design. Table 3 illustrates that using proposed architecture will increase aesthetic of mobile educational web pages. All suggested metrics will be on their perfect and optimized quantity by applying ACT, patterns and classification.

Metrics for content [Mendes, Mosley & Counsell (2001)] measure content complexity and cluster of content objects on pages. Classification of W/H devices, ACT and patterns help developers to design pages that adapted to their classification and generate optimized and best values for content metrics like page wait (page wait optimized
because of creating class-specified pages, page complexity, graphic complexity, audio complexity, video complexity and animation complexity.

Table 4. Estimation the impact of aesthetic design

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Best adapted to</th>
<th>Graphic count</th>
<th>Best conform to</th>
<th>Color count</th>
<th>Best conform to</th>
<th>Font count</th>
<th>Best conform to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Word count</td>
<td>class of W/H</td>
<td>devices</td>
<td>content &amp;</td>
<td>classification</td>
<td>content</td>
<td>classification</td>
<td>content</td>
</tr>
<tr>
<td>Text positioning</td>
<td>count</td>
<td>Best determined by the classification</td>
<td>Best conform to</td>
<td>Font style</td>
<td>Best conform to</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Header count</td>
<td>Best determined by</td>
<td>the classification</td>
<td>Font style</td>
<td>Best conform to</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[Mendes (2001)] also presented metrics for navigational required to use the system effectively. Proposed architecture use efficient components and patterns such as MVC, Page Controller, Front Controller and Application Controller to enhance and optimize metrics such as page-linking complexity, connectivity and connectivity density.

5. Conclusion

Popularity of mobile communication devices, such as cell phones, cannot directly accommodate traditional content due to the major limitation of display size and computational resource limitations. These problems have motivated the authors of this study to design a web-based architecture using design patterns to support learning system. Qualitative evaluation of the proposed architecture illustrate enhances in flexibility, agility, accuracy of software development and aesthetic, content and navigation of ml-software.

References


Fowler, M. (2002). Patterns of Enterprise Application Architecture, Addison-Wesley


