Improving coursework for Web Engineering based on MVC pattern

Seokchan Yun[1] , Sunghun Kim[2] ,Hong-Gee Kim[1]

[1]Biomedical Knowledge Engineering Laboratory, Seoul
National University, Seoul, Korea
{channy, hgkim}@snu.ac.kr

[2]Department of Computer Science and Engineering,
Hong Kong University of Science and Technology
sungkim@ cse.ust.hk

ABSTRACT

Recently, there have been tremendous changes in web engineering with adoption of Web 2.0 technologies such as front-end applications; rapid application development based on lightweight framework and service platform models by open APIs. However, college course works for web engineering do not cover these changes, while most of them tend to be focused on HTML and form processing with database or learning specific languages (e.g. PHP, Java). These lead to that many students lose their interests and they do not recognize a novel method for web development, although there exist remarkable importance of web engineering. To solve these limitations, we develop web engineering coursework 1) applying MVC (model, view and controller) framework to both server-side and front-side web technologies, 2) using lightweight framework for rapid application development (e.g. CakePHP, Ruby on Rails), and 3) using database-less programming approach with open APIs. A proposed coursework was experimented for junior students, majored in computer engineering at both Yabian University of Science and Technology, China in 2007 and Jeju National University, Korea in 2008. As a result, most of students could understand conventional MVC patterns can be applied in whole processes of web applications both server and client side. They could approach to more methodological web programming in comparison of conventional ones and understood the importance of emerging front-end technologies and are initial motivated studying web engineering furthermore. We found out this approach increase motivation and satisfaction of students significantly. In future, we will extend this approach to develop an independent course-ware of web engineering for students, making up of modeling, implementation and qualifying web applications as well as this kind of entrance coursework motivating them.

Categories and Subject Descriptors

K.3.2 [**Computer Science Education**]: University Curriculum for Computer Science – W*eb engineering, MVC pattern, Web APIs.*

General Terms

Management, Education

Keywords

Web Engineering, MVC Patterns, Web Standards, Curriculum

In: Proceedings of the WebSci'09: Society On-Line, 18-20 March 2009, Athens, Greece

1. INTRODUCTION

Recently World-wide web has been given a great deal of weight on the internet business. Especially Web technology was rapidly grown up and expanded effects for all IT industry. It reflects paradigm shift from software platform to web as a platform cause d by high computing power and broadband penetration for general users.

Web 2.0 innovators such as Google, Amazon and Ebay have offered Web APIs based on cloud computing and web developers was good at rapid development based on light-weight platform such as Ruby on Rails and rich user-interface technology as like Ajax. Varieties of web technology have been expanded, whereas college curriculum is still classical and didn't be updated. Most of them teach basic definitions of HTML, CGI and HTTP and programming skill to process HTTP transactions. It means not to be matured for educators to teach them to students systematically in terms of software engineering. It means the technology trend has been rapidly changed since adoption of Web 2.0

But, traditional college coursework for web engineering is not updated yet. Hendler J et.al [1] also indicated this problem seriously *"Similarly, if you look at CS curricula in most universities worldwide you will find "Web design" is taught as a service course, along with, perhaps, a course on Web scripting languages."*

This research improves a syllabus that offers a systematic viewpoint based on a MVC pattern in both server and client, database-less programming to develop mash-up applications with open APIs for easy understanding web services. We applied this approach to two classes and found out this approach increase motivation and satisfaction of students significantly.

2. PROBLEMS AND IDEAS

Most of web programming classes of universities have various syllabuses. But, most of them focused only on HTML markup, CGI programming and handling database. Of course some college adopted parsing XML and usages of Web APIs offered by Google, Yahoo and Microsoft.[2]

As a result of surveying 43 web programming courses of US universities, most of all focused on skill up for HTML and form processing with CGI interface, specific languages and database connection. One or two courses were just given in most CS/CE departments although there are many web based business and enterprises.

We surveyed the students that majored in computer engineering (CE) by polling methods of 135 CE students of Yabian University of Science and Technology, Yanji, China and Jeju National University, Jeju, Korea.

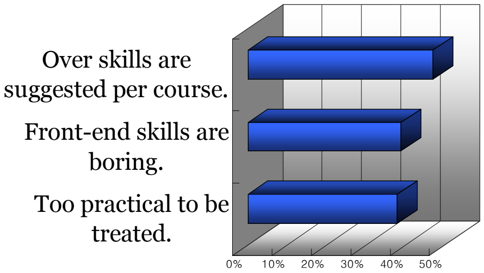

Figure 1 Why do not students attend to courses for web programming?

As a result (Figure 1.), most of them do not have interests in terms of web engineering lectures. They do not have systemic skills to develop web sites and do not understand why they have to learn HTML and CSS in CE. It needs to reduce burden for students to make interested proto-type with small skills.

So students lose interests by offering many non-systematic skills to make web sites. They thought front-end skills are not part of engineering area. HTML, CSS is boring and regarded Web is not academic but service part by mixing of idea and techniques.

Nowadays client based technologies has been strong in proportion to the computing power that markup and script language are used in rendering user-interface and controlling behavior. (i.e. Microsoft® Sliveright's XAML, Adobe® Flex's MXML and ActionScript etc.)

Also it has been diffused authoring skill that separates data structure, style and behavior with X/HTML, CSS and DOM scripting based on W3C's standards since 2000 with web standards evangelists [3] as a central figure.

It is very similar with a multitier architecture pattern so called MVC (model, view and controller). The web standards based skill became important paradigm shift with division and collaboration of planning, design and programming as well server side. It means all areas of web engineering have an elaborate process and architectural pattern by itself.

Especially various MVC pattern frameworks occurred that supports the rapid and lightweight development with PHP, Python and Ruby such as CakePHP, Django and Ruby on Rails[4]. So it's possible for web developers to make rapidly own web applications with advanced software engineering skills.

Now Web APIs often referred to as Open APIs is widely supported by web based platform vendors such as Google, Amazon, Ebay and most of web 2.0 application companies. It describes sets of technologies that enable website to interact with each other applications by using XML via REST, SOAP and any other web technology. It offers mixing outer data model to own applications so called "Mash-up". So students can divide data model in existing MVC pattern based project.

So we redesigns web class syllabus focused on as following things:

- MVC pattern in front-end technologies : Applying web standards techniques such as HTML as a structure, CSS as presentation and JavaScript as behavior. Enable to a server-less prototype using Ajax techniques mapping Model (HTML) – View (CSS) – Controller (JavaScript) pattern.
- MVC enabled light-weight framework for rapid development: Recommendation of a development framework to make (construct) a rapid prototype such as PHP (CakePHP), Ruby (Ruby on Rails) and Python (Django).
- Database-less programming to make mash-up application : Students can realize their idea by mashing up various Open APIs using XML/REST protocol.

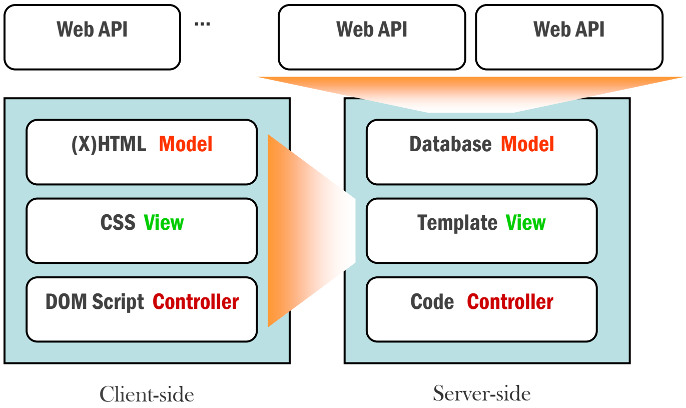

Figure 2. Suggested systematic learning elements

Figure 2 shows the basic idea to design this course work based on MVC patterns both server and client-side as well data modeling. Student can understand the basic structure of web applications and how to approach methodology to develop them.

3. CLASS DESIGN

Our class is deigned for students to adopt MVC patterns both client and server side, mixing outer data source via Web APIs and complete a creative project in short time with perfect development environment.

3.1 Brief of Coursework

Our idea of new redesigned coursework was experimented in each two different colleges in countries. Students had already the basic background of programming language and network programming.

One is the CE031 Internet Engineering: 35 Junior students majored in CE of summer semester in Yabian University of Science and Technology, Yanji, China in 2007 (Table 1. is 5 weeks intensive coursework with 2 hours in everyday.)

The other is the CE312 Advanced Web Programming: 43 Junior student majored in CE of spring semester in Jeju National University, Jeju, Korea in 2008 (Table 2 is completing 16 week based syllabus base on summer coursework.)

Table 1. Web Engineering Syllabus for 5 weeks

Weeks	Topics	Contents
1 week	Introduction	MVC patterns of server and client side and web 2.0 platform and Web APIs.

2 week	Web Documents	HTML, XHTML and XML, Styling with CSS, Client-side programming with DOM and JavaScript.
3 week	Web Programming	The basic of PHP and MySQL, MVC framework (CakePHP).
4 week	Web APIs	Web APIs and mash-up, Examples of Web APIs.
5 week	Project	Making creative web applications with overall concepts.

Table 2. Web Engineering Syllabus for 16 weeks

Weeks	Topics & Contents
1 week	A History and Introduction of Web, Front-end technology: Structure, Presentation and Behavior
2 week	Server-side technology - LAMP, Web Application Frameworks, Web 2.0 and Web APIs
3 week	Front-end: HTML, XHTML, XML
4 week	Front-end: CSS : Styling, Layout, Selector
5 week	Front-end: Document Object Model and JavaScript?
6 week	Client-Programming: Web APIs with JavaScript? (Google Ajax API…)
7 week	Server-Programming: MVC Model and Lightweight Framework
8 week	Server-Programming: Ruby
9 week	Server-Programming: Ryby on Rails
10 week	Project: Web APIs – REST, XML, JSON, RSS Parsing
11 week	Future of Web: Microsoft Silverlight
12 week	Future of Web: Adobe Flex
13 week	Future of Web: Firefox Extensions

In the latter course, we substituted CakePHP to Ruby on Rails and added special lectures about Microsoft Silverlight and Adobe Flex calling attention for front-end technology.

Both class adopted open source project hosting to operate course. In detail, you can see all of class material in our class homepage. [5] and [6.

3.2 Teaching Methods

Each lecturer gave a team teaching with oral and exercise in every day, so that the class was divided by eight project team. Sudents practiced a code excise in every time except CSS styling. In case of CSS, they used design templates of free CSS sites after understanding structure markup with XHTML. They can make simple web applications via Ajax and Web APIs without server-side code.

In teaching server-based MVC pattern, it's very useful various screen-cast such as CakePHP and Ruby on Rails for making simple blog in 10 minutes. Each project teams make their own ideas and give a concrete from code excises of Web APIs.

3.3 Term Projects

In this basic background, term projects could make their own idea to web applications to use the latest web technologies including structural markup, CSS styling and Ajax skills mixed with Google Maps and Korean search APIs. Students are interested new concepts for web programming, but hard to understand MVC pattern and process model without the basic knowledge of software engineering. But, during term project all of members divided each rolls such as data modeling, markup, designing and programming and could accomplish the collaboration work together.

4. RESULTS

We surveyed polls for students after course and found significant increasing of satisfaction. Respectively, 85% and 88% students had satisfied this coursework and wanted to study web programming further. Respectively, 83% and 92% students understood importance of front-end technologies in the modern web services. Respectively, 73% and 83% students wanted to divide this course into three or more intensive courses.

Also we could make induction of interests that students could make simple web applications via Ajax and Open APIs without server-side programming. They were interested in screen-cast such as "Making blog in 10 minutes" and could make their own ideas by using concrete code excises offering by Google Ajax Search and Yahoo! Maps API

5. CONCLUSION

Most of students could understand conventional MVC patterns can be applied in whole processes of web applications both server and client side. They could approach to more methodological web programming in comparison of conventional ones. They understood the importance of emerging front-end technologies and are initial motivated studying web engineering furthermore. We found out this approach increase motivation and satisfaction of students significantly. We have to make the independent courseware of web engineering for students of computer engineering including modeling, implementation and qualifying web applications as well as redesigning entrance coursework motivating them.

Most of students could understand traditional MVC patterns and excises whole processes of web applications both server and client side in short time. They could approach to web programming with architectural methodology compared with the past.

6. REFERENCES

[1] Hendler J., Shadbolt N., Hall W., Berners-Lee T., Weitzner D. (2008). Web Science: An Interdisciplinary Approach to Understanding the Web. Communications of the ACM, 51(7), pp. 60-69.

[2] Jim Whitehead, Hypermedia and the Web, CS 183,University of California, Santa Cruz http://www.soe.ucsc.edu/classes/cmps183/Spring06/

[3] Homepage of Web Standards Project. http://www.webstandards.org.

[4] Kinds of Web Application Framework http://en.wikipedia.org/wiki/Web_application_framework

[5] CE031 Internet Engineering, http://yust.wikispaces.com

[6] CE312 Advanced Web Programming, http://code.google.com/p/web-engineering-class/