Web-Oriented Software Development

Seminar OHJ-1860

Tampere University of Technology

Spring 2006

Prof. Tommi Mikkonen & Dr. Antero Taivalsaari

Background and Motivation

- The widespread adoption of the World Wide Web has dramatically altered the landscape of software development.
- Today, most new software applications will have to be designed with the Web in mind
  - … either to be used via a web browser or some other client device, such as a mobile phone.

Background and Motivation

- The programming paradigms and languages that we use today are still primarily *web-agnostic*.
- Most programming languages and systems today are not well-suited for software development for the Web.
- They were designed during an era when ubiquitous networking and globally available data and resources were still an unrealized dream.

What is Missing?

- Support for networking as a built-in mechanism; instead, networking is included as an add-on feature.
- Support for global object references at the language level.
- Support for object, code and thread migration.
- Support for geographically dispersed applications with geographically distributed data and resources is limited.
- Support for continuous 24x7 operation and dynamic changes is limited; applications have to be stopped before they can be updated, upgraded or otherwise modified.
- No close integration with web servers or web browsers.
- …

What is the Problem?

- Today, software developers writing applications for the Web often find themselves in working *around* the tools and language features, rather than being helped by them.
- In the absence of web-oriented programming mechanisms and idioms, software developers are often repeating the same tasks over and over again.

Towards a New Disruptive Period

- Programming language evolution has been slow in the past ten years.
- Nothing really new has happened after Java came along in mid-1990s.
  - Even Java is mostly a combination of technologies from the 1970s and 1980s.
  - Aspects, frameworks, etc. haven’t offered much relief; they aren’t really focusing on the problems that the web app developers are facing.

A New Disruptive Period

- In the past two years or so, new “web-oriented” systems have started to emerge.
- These systems have become increasingly hot topics in the developer community.
- As is typical in the beginning of a disruptive period, most of these systems are “hybrid” systems; some common patterns are starting to emerge.

Some of the New Systems

Ajax	(http://www.ajaxinfo.com/, http://www.ajaxian.com/)
BeanShell	(http://www.beanshell.org/)
Dojo	(http://www.dojotoolkit.org/)
E4X	(a.k.a. JavaScript for XML or ECMAScript for XML) (http://www.w3schools.com/e4x/default.asp)
Flash	(http://www.macromedia.com/flash)
Groovy	(http://groovy.codehaus.org/)
Jython	(http://www.jython.org/)
Microsoft ASP	(http://www.asp.net/
Open Croquet	(http://www.opencroquet.org/)
PHP	(http://www.php.net/)
Ruby on Rails	(http://www.rubyonrails.org/, http://www.ruby-lang.org/)
Wheat	(http://www.wheatfarm.org/)
XJ	(http://www.alphaworks.ibm.com/tech/xj)

Highlights: Ajax

- Ajax is a combination of technologies to facilitate the development of highly interactive web apps.
- Four key technology components:
  - *JavaScript* – A general purpose scripting language.
  - *Cascading Style Sheets* (CSS) – for defining reusable visual styles for web page elements.
  - *Document Object Model* (DOM) – to present the structure of web pages as a set of objects that can be manipulated by JavaScript.
  - *XMLHttpRequest object* – to retrieve data from a web server as a background activity.

Highlights: Groovy

- A variant of Java with a number of web-friendly features:
  - closures, embeddable expressions
  - first-class primitive objects
  - native syntax for lists, maps and regular expressions
  - native syntax for markup languages (XML, HTML, SAX, …)
  - dynamic typing and polymorphic iteration support
  - simpler syntax for JavaBeans development
- To be standardized as a Java Community Process (JCP) effort JSR 241.

Highlights: Ruby on Rails

- An open-source web framework optimized for creating database-backed web applications according to the MVC (Model-View-Controller) pattern.
- Based on the Ruby programming language.
- Goals: Enable the development of web apps ten times faster than with a typical Java framework; avoid verbose XML configuration files.

Highlights: Wheat

- A language and environment for creating dynamic web sites likes blogs and portals.
- Features:
  - Every object is an URI
  - Namespaces are global (internet-wide)
  - Tree-structured object hierarchies; can be mounted on any media; XML used as the object interchange format
  - The Wheat virtual machine is also a web server
  - Development environment is integrated with Wiki

Highlights: Flash

- A language and authoring environment for creating interactive websites and web content.
- A commercial product by Macromedia.
- Includes an advanced development environment/toolset.
- Includes a powerful scripting language called ActionScript.
- Uses vector graphics extensively to speed up web content rendering.

Highlights: PHP, JavaScript, ...

- Various scripting languages are also evolving into more general-purpose web development systems.
- *PHP*: a server-side HTML embedded scripting language. Provides web developers with a suite of tools for building dynamic websites.
- *JavaScript*: A client-side scripting language embedded directly into HTML pages. Supported by pretty much every web browser.

Common Aspects in These Systems

- Challenge the status quo.
- Combine previously (mostly) unrelated technologies.
  - XML, CSS, scripting capabilities, etc.
- For now: Solve a practical problem rather than search for the most elegant solutions.
- Often: Tool-oriented or tool-assisted approach.
- Much more dynamic than conventional programming languages (interpreteration & dynamic typing preferred).
- At the implementation level: Close coupling with a web server or web browser.

Why This Seminar?

- We think that a new period in programming language evolution has started.
- We are potentially moving towards a new paradigm beyond POOOP (Plain Old OOP).
- Learn more about these emerging technologies in general.
- Investigate the trends and identify common patterns.

Practical Arrangements

- The seminar will be arranged on the following Fridays, 12-14 o’clock.
- Student presentations:
  - March 24, 31
  - April 7, 28 (14 is Easter, 21 we both travel)
  - May 5, 12, 19, 26
- Reports completed by Apr 30, peer review during May, and final report completed in June.

How to Get Credits

- Maximum number of credits: 5 op

- Attendance: 1 op
- Seminar presentation: 2 op
- Written report: 2 op (optional)
  - Gathered to a laboratory report at the end of the seminar; distributed to all the participants.

Choosing the Presentation Topic

- Please choose your presentation topic and the preferred presentation date as soon as possible.
  - tjm@cs.tut.fi. Updated list at http://www.cs.tut.fi/~taivalsa/kurssit/WOSD2006
- Topics allocated on a “first-come-first-serve” basis.
- Topics include (but are not limited to) the topics mentioned in the seminar overview.
  - Obviously, the focus in the presentation should be on software development for the Web.

Possible Presentation Topics

Ajax	(http://www.ajaxinfo.com/, http://www.ajaxian.com/)
BeanShell	(http://www.beanshell.org/)
Dojo	(http://www.dojotoolkit.org/)
E4X	(a.k.a. JavaScript for XML or ECMAScript for XML) (http://www.w3schools.com/e4x/default.asp)
Flash	(http://www.macromedia.com/flash)
Groovy	(http://groovy.codehaus.org/)
Jython	(http://www.jython.org/)
Microsoft ASP	(http://www.asp.net/
Open Croquet	(http://www.opencroquet.org/)
PHP	(http://www.php.net/)
Ruby on Rails	(http://www.rubyonrails.org/, http://www.ruby-lang.org/)
Wheat	(http://www.wheatfarm.org/)
XJ	(http://www.alphaworks.ibm.com/tech/xj)

Discussion