Research Proposal on

Evaluation of future JEE technologies

CA1068A: Research Methods and Dissertation

Nicolas Lanquetin
0604918@abertay.ac.uk

15th December 2006

University of Abertay Dundee
School of Computing & Creative Technologies

Abstract

Most large web applications run on the J2EE standard. Many frameworks have been developed to ease developers' work and offer a standardised platform for teams to realise their projects on. Two very recent technologies, Struts Shale and EJB 3.0, will be used in a web application implementation in order to evaluate their new features, their degree of sophistication and their maturity. The results of this research will outline if it is advisable to already adopt these technologies in the development of new projects. The study will be carried out by analysing different performance tests and evaluating the results of the critical appraisal of various criteria.

Contents

1. Introduction

Since 1991 the world has started to actively use the World Wide Web. There are currently over 700 million people connected to the internet, making it one of the most important communication network. It has not taken long for corporations and individuals to find out that the internet is an ideal communication medium.

The most common form to communicate information over the internet is the classical website, which is still widely used today. Whereas the past technologies were still very simple and allowed mostly static content, web applications needed to meet new requirements such as the coping with great amount of data, dealing with session handling to provide customized content and ensure security, performance and reliability. It is needless to say that the technologies gained in complexity. There was a need of standardisation and code reusability in order to reduce production time and avoid redundant tasks.

Nowadays, it is therefore required to have experience in various areas to build an enterprise web application: Professional web development demands great knowledge of different technologies and a clear view of how they interact. Most large-scale web applications are build on the JEE[1] standard and make use of various prebuilt components, in most cases as a framework and some middleware. The last 6 years have brought a lot of revolutionary changes to the JEE web development, introducing many web application frameworks and middleware components in the persistence layer. For large-scale web applications these components are no more an option.

This research study will not only clarify the use of many of these technologies, but also critically analyse two very promising newcomers in the framework world. It is the aim of this study to find out if these two very recent technologies are worth being used in professional environments or if it is better to stay with already well established technologies. Only little resources exist at the moment, because of the recentness of these technologies. It is therefore all the more interesting to take a closer look at them.

[1] Java Platform, Enterprise Edition (formerly known as J2EE up to version 1.4)

2. Research Question

Within the research area of JEE web development, future JEE technologies will be evaluated. The research topic includes the evaluation of a specific web application frameworks (hereafter WAF), as well as specific middleware and build tools.

2.1. General Research Questions

The general questions of this research are as follows:

1. *How sophisticated are the capabilities of the WAF Apache Struts Shale[1]?*
2. *How well does the middleware component EJB[2] 3.0 do, compared to its older version?*

For a large-scale web application, the WAF (question 1) and one middleware component (question 2) for data persistence is an appropriate start. By answering the two general questions, this research will clarify if it's advisable building new web applications on these technologies.

2.2. Specific Research Questions

The research and especially the data-gathering will be guided by the following questions:

1. How well does the WAF Struts Shale do on the following criteria:
   a) **Complexity**
   How complex is the technology?
   b) **Performance**
   How fast are the request speeds in different test situations?
   c) **Scalability**
   Can the framework be extended easily? Can other frameworks and middleware components be easily integrated?

[1] Apache Struts Shale is produced by the ASF (Apache Software Foundation). In this document the codename *Struts Shale* will be used.
[2] Enterprise Java Bean

d) **Server- and client side validation**
How is form validation implemented and what kind of different possibilities exists?

e) **Internationalisation (i18n)**
How can different languages and nation-specific standards be used in the WAF?

f) **Speed of Development**
How fast can the technology be implemented (short- and long run)?

g) **Testability**
How sophisticated is the testing framework? Is it possible to unit test and test HTTP requests?

h) **Bookmarkable URLs**[3]
Is it possible to generate URLs which can be bookmarked?

i) **Duplicate post-problem**
How is the duplicate post-problem solved in the WAF?

j) **Tool Support**
Which development environments tools support the WAF?

2. How well does the middleware component EJB 3.0 do on the following criteria:

a) **Complexity**
How complex is the technology?

b) **Performance**
How fast are the request and virtual machine speeds in different test situations?

c) **Speed of Development**
How fast can the technology be implemented (short- and long run)?

d) **Testability**
How sophisticated is the testing framework?

e) **Tool Support**
Which development environments tools support EJB 3.0, especially concerning annotations[4]?

[3] Uniform Resource Locator (a HTTP address for instance)
[4] Annotations are used in attribute-oriented programming and considerably ease the work of developers.

3. Literature Review

For this research project, there are two main bodies of literature which are of interest. Firstly reviews or opinion based articles, and secondly development guides and any other support documents. While it is possible to find many opinions about the technologies used in this study, it is much more difficult to find support on some of the technologies. This lack of information is due to the fact that the evaluated technologies are still very recent and not widely used yet.

For this reason, only a few books can be found at the moment. Nevertheless, most information can be acquired from articles or short guides of specialised on-line magazines. The most famous on-line magazines for JEE development are *JavaWorld.com*, *TheServerSide.com* and *OnJava.com*. Beside providing excellent articles, these websites also offer a good ground to discover the current hypes and opinions about specific issues or technologies through the community comments and discussions. Most development related information can also be found on the projects' official websites.

Furthermore the literature can be split up into two further categories: one dealing with EJB 3.0, and another concerning the Struts Shale WAF and the build tools Maven 2 and XDoclet 2. There exists many resources on the first category, since the EJB 3.0 specification is included as standard in JEE. The second category however has only very little information and support, since the first Struts Shale alpha version has been released in August 2006. Not only the release of Struts Shale, but also the development with Maven 2 and XDoclet 2 is very recent. On Maven 2 there is therefore very little support, except one very detailed book from Massol et al. (2006). As of XDoclet 2, needed for the Struts-specific descriptor[1] generation, it is integrated in Maven 2 as a plugin and already covered in most Maven 2 tutorials.

3.1. Struts Shale

Even though the literature resources are relatively poor for Struts Shale, the evaluation of that particular WAF is of great interest. First of all, a WAF is in most cases necessary for large-scale web applications to avoid starting an application from scratch. It has of course many other advantages as explained by Shan and Hua (2006, chapter 2.3): The most important ones would be the 'reduction of the amount of time, effort, and resources

[1] Descriptors are meta information needed principally for the deployment and running of the application.

required to develop and maintain web applications. Furthermore, the framework integrates many commonly accepted standards and greatly improves the teamwork, since developers acquainted with the framework can rapidly start the development.

There exists hundreds JEE frameworks, but only a few made it to the top. The most popular frameworks include *Struts*, *Spring MVC*[2], *JSF*[3], *Turbine* and *Tapestry* (Corcho, López-Cima and Gómez-Pérez, 2006, p. 145). The ones relevant to our study are Struts 1.x and WebWork[4]. Both communities decided in November 2005 to merge these two frameworks into one (see figure 3.1). In fact the new Struts version will build on the second version of the WebWork framework. The resulting WAF has the codename *Shale* and is now the successor of Struts 1.x (IndicThreads, 2006*a*).

Figure 3.1.: WebWork and Struts merge to Struts 2

Lightbody, project lead for WebWork and committer at Apache Struts, seems very confident about the benefit of that merge. According to him 'WebWork is a great *technology*, and Struts is a great *community*' (Lightbody, 2005; IndicThreads, 2006*b*). Each community will greatly benefit from each other's strength: Struts will benefit from the many advantages of the WebWork framework, and WebWork will be able to grow due to Struts' larger community.

There are several reasons why Struts Shale was chosen as WAF in this research project. Raible, experienced consultant for J2EE developers, has done a comparison of different frameworks. The outcome of his research shows that Struts is by far the most popular one: The latter has the most job opportunities, the largest community and also the greatest support in term of books (Raible, 2006, p. 107–111). However, it must be kept in mind, that these results regard Struts version 1.x. The reason why it is mentioned here is that developers using this WAF will tend to stay with something they are already familiar with. They will probably change for a newer version of their favorite framework, instead of moving on to something completely new. Even though Struts Shale is in many ways different to Struts 1.x, both versions share the action[5]-based approach and, according to an interview with Lightbody, the compatibility between both Struts versions is a high priority (IndicThreads, 2006*b*, p. 1).

[2]Model View Controller
[3]Java Server Faces
[4]WebWork is a Java web application development framework.
See http://www.opensymphony.com/webwork/
[5]Actions are the main components of the Struts WAF. They invoke the business logic and data access logic.

One might argue that it is too soon to already use Struts Shale. There are of course many reasons for this: The framework is too recent, therefore possibly neither stable nor reliable, and it has only limited support because the documentation is not very complete at this moment. Nevertheless, Apache (2006*c*) claims that Struts Shale is very stable, because 'it is a direct descendant of WebWork 2'. While Struts Shale is not recommended for the migration of old Struts projects, is it however ideal for new projects. In addition, the community provides some working example applications or blank applications to start with.

As mentioned in chapter 2.1, one of this research project's aims is to find out if the framework is worth being used in productive environments. As a result of the study, it will be clear if the WAF has matured to a state where it is reasonable to start new projects with it. As of the second aim of this study, the evaluation of EJB 3.0, it tries to find out how useful and high-performance the middleware really is.

3.2. EJB 3.0

The primary use of EJB is the object-relational mapping[6] (hereafter O/R mapping) and the transactional integrity[7]. It is widely known that EJB, till version 2.3, suffers from a bad reputation (Yuan, 2005*a*; Sharma, 2004). This is due to the complexity of the technology and the bad performance till version 2.1[8]. Many businesses continued to adopt EJB, yet knowing that EJB had many problems. At the same time a growing number of developers started using some more lightweight solutions such as *Hibernate* and the *Spring Framework*.

Why choose EJB as middleware for the O/R mapping over one of the lightweight and far less complicated solutions? The argument in favor of EJB 3.0, is that the mistakes of the past were recognised, and the specification completely revised. Crume et al. (2006, p. 2) explains that the new version of EJB (version 3.0) is developed by an expert group[9] and differs completely from its predecessor: The EJB specification was adapted to be more lightweight and is now based on the Hibernate O/R mapping framework.

[6] Object-relational mapping is the mapping of objects used in an object oriented language to data in a relational database. It serves the purpose of making objects persistent without having to deal with types.

[7] EJB's real power lies in the very good support for transactions.

[8] In fact, EJB was using remote method invocation for each called method, even if the application was running locally. This can be extremely slow, especially in multi-user application.

[9] An expert group is a development community formed by experienced members from important organisations, such as JBOSS, Oracle or Google.

Furthermore EJB 3.0 makes use of annotations[10] introduced with JEE, the fifth version of the Java Platform. Using annotations considerably shorten and simplify the code, as demonstrated by Yuan (2005*b*,*c*). The code is now 'extremely simple' and functionality from older EJB versions became 'more manageable and succinct' (Rubio, 2006, p. 3). Regarding the performance, it appears that EJB 3.0 has made huge progresses: According to an interview with Kodali (2006), product manager and evangelist in the Oracle Application Server Group, some performance tests have been run at Oracle with very surprising and satisfactory results. In some cases the performance even doubled, compared to the predecessor version. In addition, migration to EJB 3.0 is fairly uncomplicated, as claimed by the software architect Tiwari (2006). This could be very interesting for projects still using the old version.

However, some developers, such as the J2EE senior engineer Sharma, are anxious about some decisions made in the EJB 3.0 specification; for instance about the use of a SQL-like syntax in EJB annotations, or the growing number of defined annotations, which certainly makes EJB 3.0 *easier*, but still not *easy* for non-specialised developers (Sharma, 2004, p. 3).

For all this great changes and implicated features or problems, EJB 3.0 is without doubt very interesting to look into. Because the technology is not widely used yet, only little resources exist. An own evaluation of EJB 3.0 is therefore more than appropriate. To complete the image, the build tools needed by the WAF and for the deployment will be discussed.

3.3. Build Tools

Maven is used to manage project builds, as well as reporting and documentation from a central place and in a standard manner (Apache, 2006*b*). It is a result of best practices based on years of experience. The number of Maven users grows each day, as developers discover the benefits of this tool. Struts Shale and its proposed example applications also make use of Maven 2 to manage their projects. Many professional developers such as Massol (2005), also author of the most complete Maven book (Massol et al., 2006), Redmond (2006) and Smart (2005, 2006) agree that Maven 2 is the best solution to simplify the build process. In their work, they demonstrate the ease of use with various examples. For all these reasons, Maven 2 will therefore also be used in the research project.

To summarize, many different technologies are necessary to build a large-scale web application. By comparing the different technologies, it was possible to find the ones that are worth the most being analysed, tested and evaluated. A great amount of work

[10] More about EJB annotations can be read in the EJB 3.0 specification (Sun, 2006)

has already been done in the new J2EE technologies discussed in this proposal. Some technologies are however very recent and still not labeled as *stable*. It is nevertheless all the more a reason to evaluate those technologies. The results might very well have an impact on decisions regarding migrations or new projects.

4. Methods and Approaches

4.1. General Design

Each technology of the study will carefully be analysed and evaluated on different criteria mentioned in chapter 2.2. A mixture of quantitative and qualitative research will be carried out for all performance tests, since the evaluation will principally be based on the gathering of load test data. As for the other research areas, a qualitative approach will be chosen. It is with the results of the tests and the critical appraisal of the different criteria, that the research question will be answered.

Because the study tries to answers questions from different areas, it makes sense to look at each research method separately.

4.2. Methods for Struts Shale

Scalability

Strategy and Framework For a WAF it is important that other standardised technologies can easily be integrated. Furthermore the application should be extendible in functionality, without increasing the degree of complexity.

Data Collection The data to be collected are all the possible standards which make sense to be integrated into the WAF. It must also be researched to what extent and how easy this integration process can be implemented.

Server- & Client Side Validation

Strategy and Framework One of the most important functionalities in WAFs is the handling of form validation. They are many solutions proposed by the framework, such as the server-side validation, or client-side validation with either JavaScript or AJAX[1].

[1] AJAX (Asynchronous JavaScript and XML) is used to refresh specific parts of a website instead of reloading the entire page.

Data Collection Each solution should be implemented, analysed and compared. The data to be collected is if the functionality works efficiently and faultless. Especially for the client-side validation, it must be evaluated if the results are conform with standards (W3C[2] for instance).

Other WAF features

Strategy and Framework Other features, such as internationalisation, duplicate post-problem or bookmarkable URLs play an important role in a WAF.

Data Collection Each of these features, once implemented, should be tested and evaluated on specific criteria.

4.3. Common Methods for Struts Shale and EJB 3.0

Performance

Strategy and Framework The technologies must be evaluated on their capability to handle multiple requests simultaneously, especially when session data must be hold.

Data Collection For the gathering of these test data, a load test application must be used. In the proposed research, the free desktop application *Apache JMeter* is the best choice. It is designed to load test functional behaviour and measure performance. Over a network of distributed JMeter clients, the application will be tested using HTTP requests. It is possible to vary the test cases by changing the test parameters. Possible variations could for instance depend on the definition of numbers of requests in a particular time, which pages are requested or how the runtime environment is tuned. For the recording of internal application speed, the very popular logging tool *Apache Log4J* will be used.

Complexity and Development Speed

Strategy and Framework For the development it is important to produce effort estimations. To do this accurately, it is necessary to know the complexity of the used technologies. It is also important to know how fast one could get acquainted to new technologies and at which pace the developer can implement projects.

[2]The W3C is the the World Wide Web Consortium. They define the specification and guidelines for internet standards.

Data Collection The data which needs to be collected is principally the time which is necessary to start the project and the time required for the maintenance and addition of future requirements. The data could be represented by learning curves diagrams.

Testability

Strategy and Framework Each framework component should be testable in order to produce reliable quality software. In this research project, *JUnit* will be used to unit test the application.

Data Collection It must be evaluated if it is possible to test the application directly without going through a HTTP request, and also if all parts of the application can be tested efficiently.

Tool Support

Strategy and Framework Tool support is very important when it comes to save time on redundant tasks; the developer should ideally only focus on the business logic.

Data Collection The data to collect is principally which tools exist for which development environments. It is furthermore crucial to analyse how beneficial they are, and how easy they are to use.

5. Significance of study

The research on future JEE technologies is without doubt very beneficial for JEE developers. In practice it is rare that developers get a chance to work on new technologies, because businesses prefer to stay with working and stable technologies. It is however crucial for developers to stay up to date in order to stand a chance against other competitors. On that account, this research is greatly enriching for developers which do not have the time to evaluate the technologies, but want to know if it is worth to adopt them and how to use them.

The research on the new Struts version is of great interest. Struts Shale will certainly be a promising revival of the old Struts version. Chances are good that this WAF will be used in future projects, since the Struts community is very large and the framework very powerful. The same is also true for the new EJB version. Especially in JEE development, there is a need for highly specialised developers with perfect knowledge of the various technologies and how they interaction. By having early experience in that field and especially with new technologies, the prospects for a qualified job might be much brighter. Furthermore it could also be imaginable to get a lead position for projects build on these new technologies.

On a personal base, this research study will also increase the ability to self-evaluate oneself. In the practice, it will be required to always stay up to date with new and emerging technologies. The evolution in IT environments is much faster than in most of the other branches; developers have to constantly acquire new development skills. To have a feeling of the required time to get acquainted with new technologies is therefore to the best advantage.

6. Limitations and Limits

In order to evaluate and critically appraise the technologies, most of the research results will be set in relation to other frameworks/middlewares or older version of the used technologies. However, because of the time restriction, it will in some cases not be possible to compare the technologies; answers to the research questions will then be based on personal practical experience and known customer expectations.

Furthermore, the topic and specification of the web application is not yet defined. The reason for this is that the application purpose will depend on what there is to test and evaluate. Analogue to professional web development, the topic and ideas for the web application will be given by the client. The developer in return focuses on his core skills: the *implementation* of the application. Therefore, it should at this point not matter what the web application will be about in the research project.

A. Research Project Aims

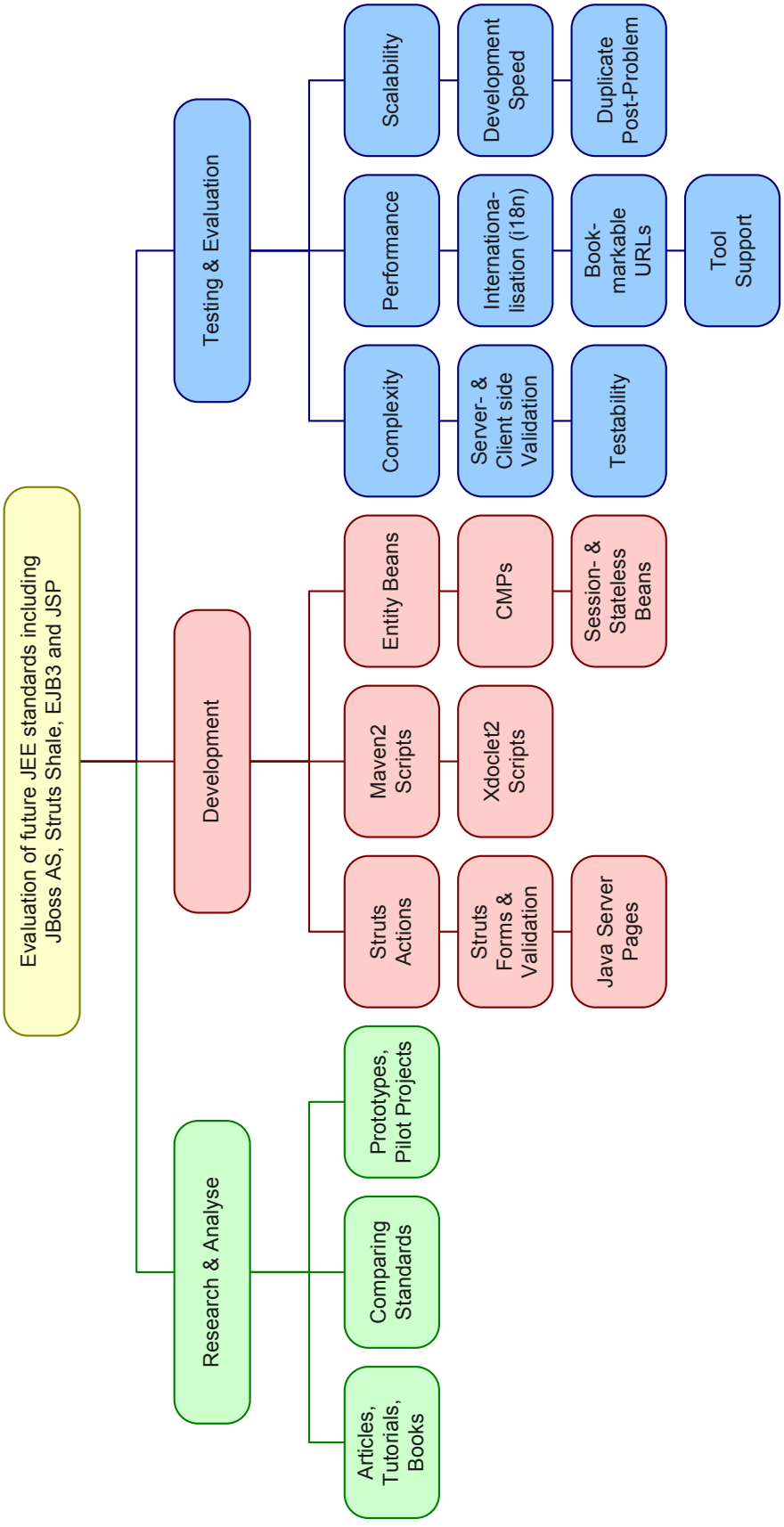

Figure A.1.: Research Project Aims

B. Research Project Gantt Chart

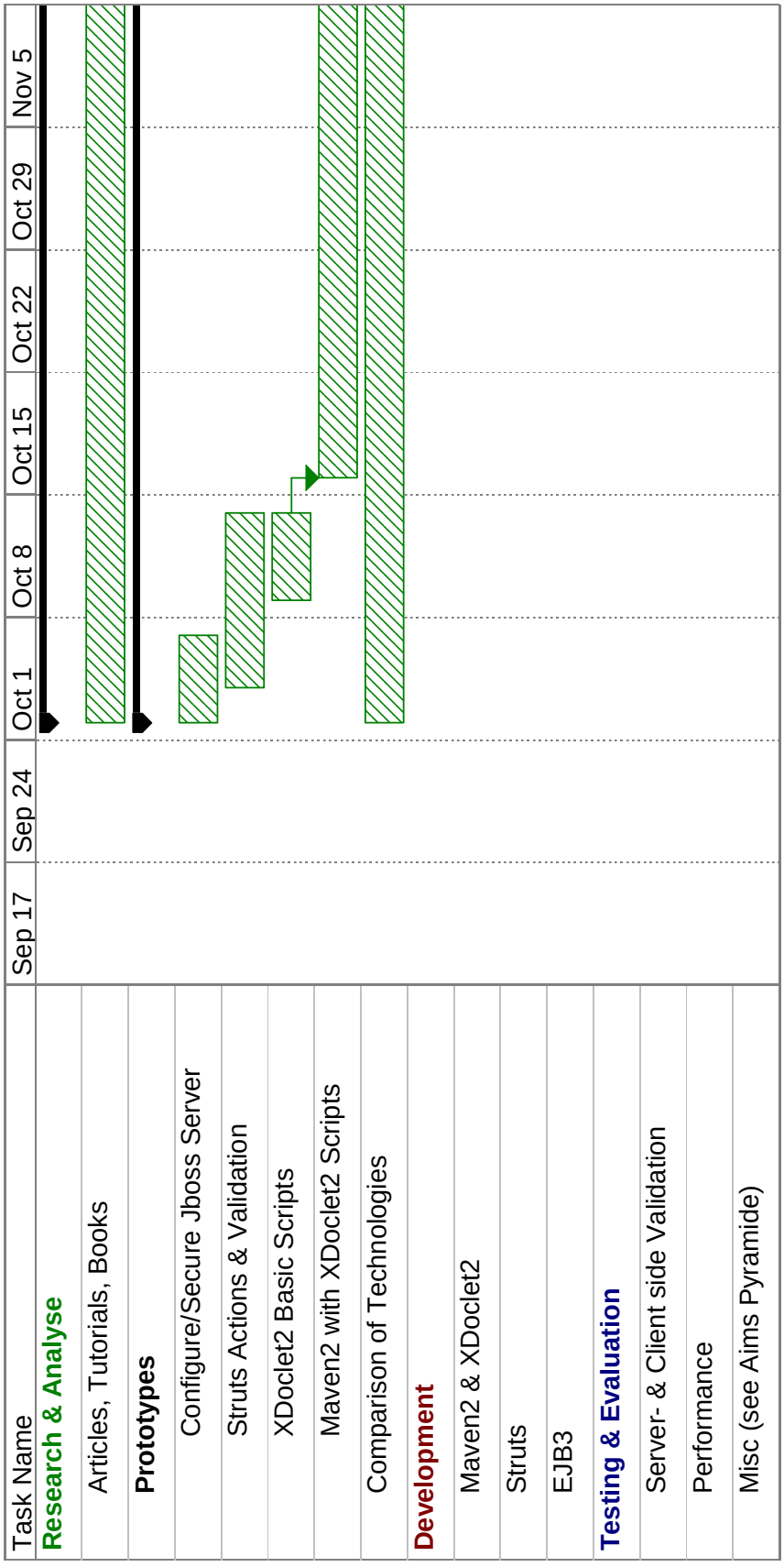

Figure B.1.: Research Project Gantt Chart: September '06 – November '06

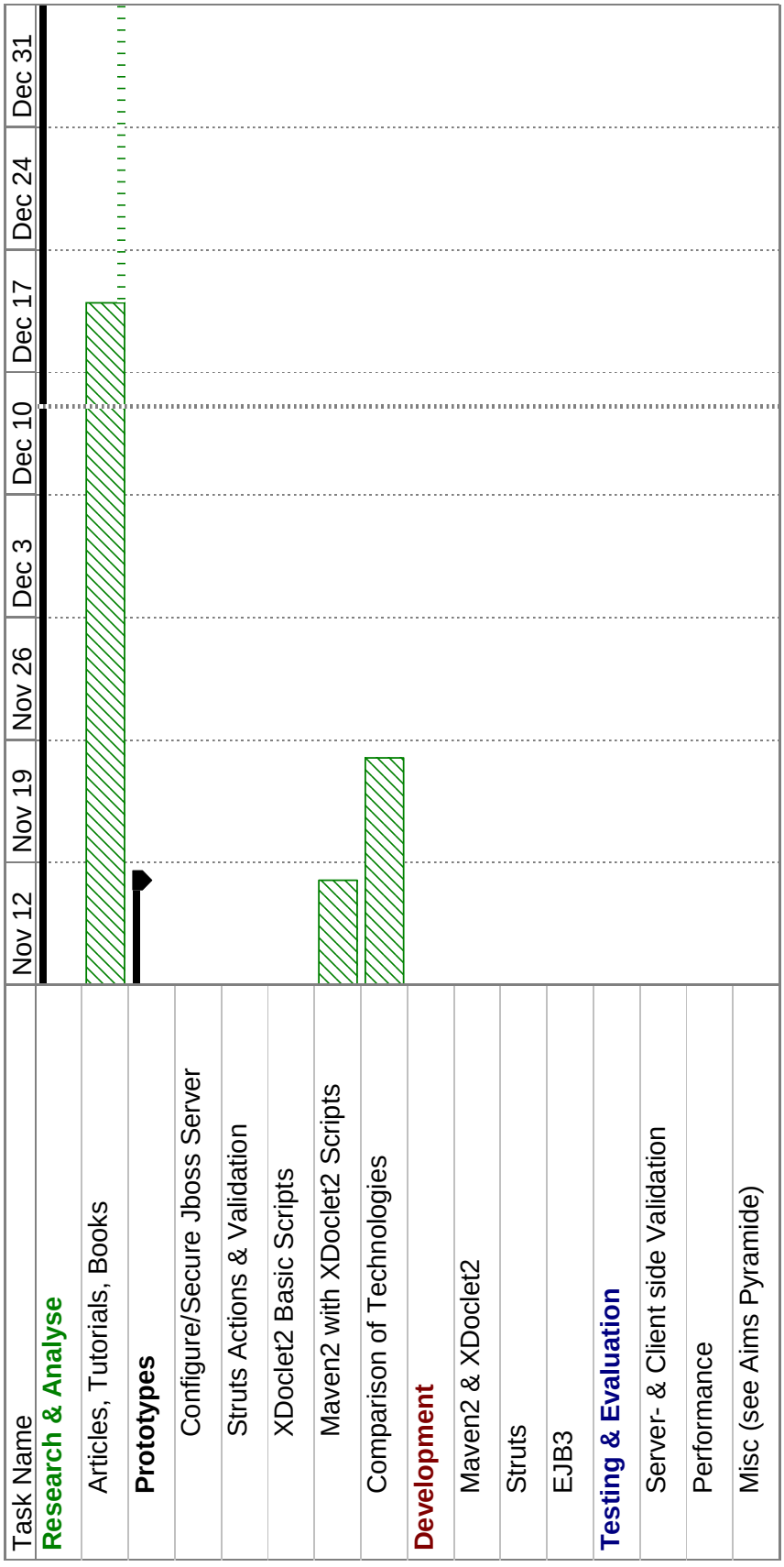

Figure B.2.: Research Project Gantt Chart: November '06 – December '06

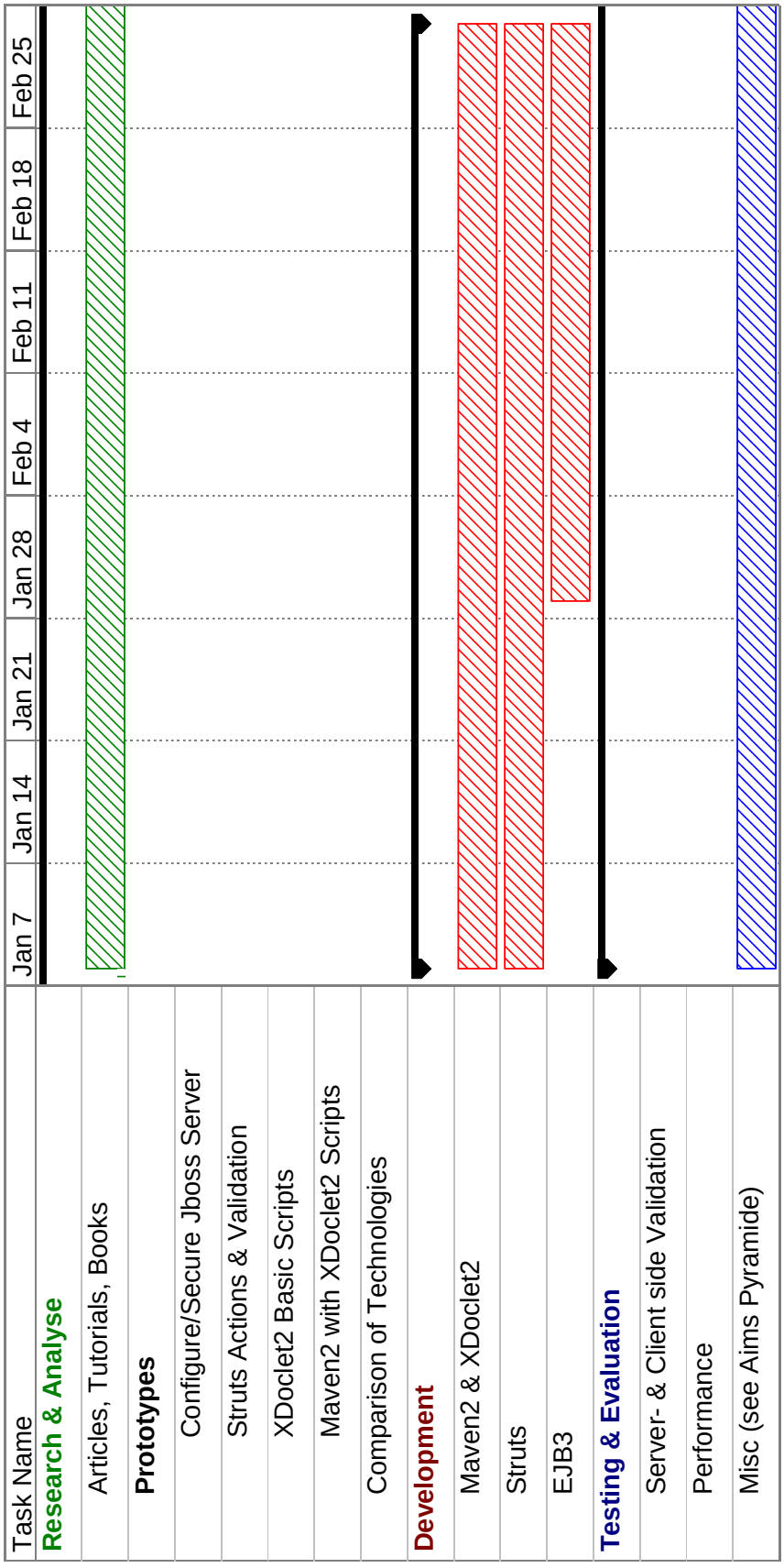

Figure B.3.: Research Project Gantt Chart: January '07 – February '07

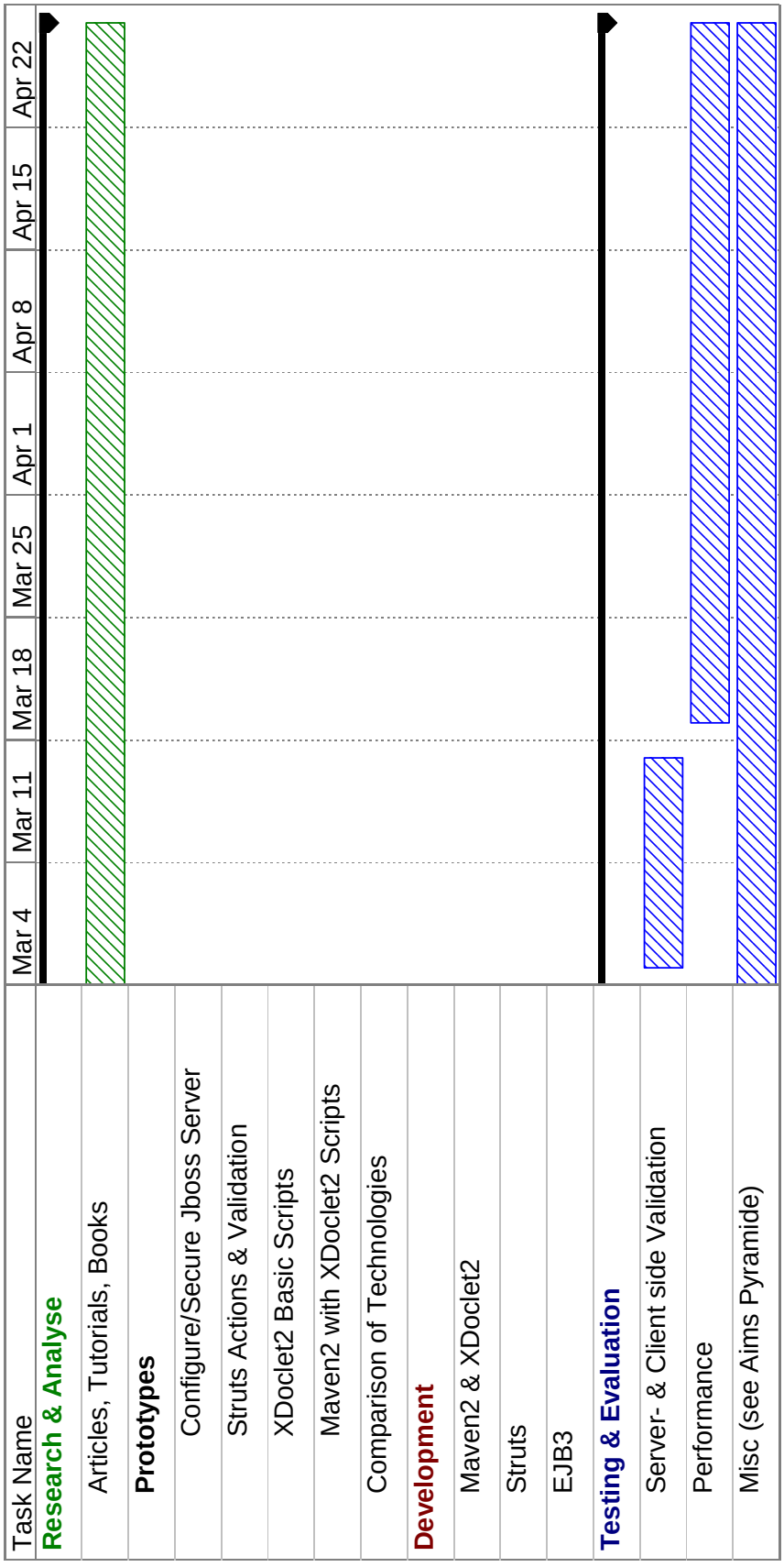

Figure B.4.: Research Project Gantt Chart: March '07 – April '07

List of Figures

Glossary

Notation	Description
AJAX	**A**synchronous **J**avaScript **a**nd **X**ML: *AJAX is used to only refresh specific parts of a website instead of reloading the entire page.*
ASF	**A**pache **S**oftware **F**oundation: *The Apache Software Foundation is a non-profit corporation, operating many different web related projects and a wide community of members.*
EJB	**E**nterprise **J**ava **B**eans: *EJB is a component architecture for the development and deployment of object-oriented, distributed, enterprise-level applications.*
HTTP	**H**ypter **T**ext **T**ransfer **P**rotocol: *HTTP is amongst others the protocol used to browse the World Wide Web.*
J2EE	**J**ava **2** Platform, **E**nterprise **E**dition: *J2EE is a version of Java for developing and deploying enterprise applications.*
JEE	**J**ava Platform, **E**nterprise **E**dition: *Formerly known as J2EE up to version 1.4, the term JEE is now used for web application running on version 5 of the Java Platform.*
JSF	**J**ava **S**erver **F**aces: *JSP is a java-based web application framework.*
MVC	**M**odel **V**iew **C**ontroller: *A paradigm, stating that the data (model) should be seperated from the user interface (view) and the processing (controller).*

Notation	Description
URL	**U**niform **R**esource **L**ocator: *A URL is the unique address for a file that is accessible on the Internet.*
W3C	The **W**orld **W**ide **W**eb **C**onsortium: *The W3C defines the specifications and guidelines for internet standards.*
WAF	**W**eb **A**pplication **F**ramework: *A WAF provides various functionalities required to build a web application.*

Bibliography

Apache. 2006*a*. *Apache Struts Shale Documentation*. The Apache Software Foundation. Available at: http://shale.apache.org/documentation.html.

Apache. 2006*b*. *Maven Documentation*. The Apache Software Foundation. Available at: http://maven.apache.org/guides/index.html.

Apache. 2006*c*. "Struts Roadmap FAQ.". Available at: http://struts.apache.org/roadmap.html#stable.

Cavaness, Chuck and Brian Keeton. 2003. *Jakarta Struts Pocket Reference*. O'Reilly.

Cecchet, Emmanuel, Julie Marguerite and Willy Zwaenepoel. 2002. Performance and scalability of EJB applications. In *OOPSLA '02: Proceedings of the 17th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications*. New York, NY, USA: ACM Press pp. 246–261.

Corcho, Oscar, Angel López-Cima and Asunción Gómez-Pérez. 2006. A platform for the development of semantic web portals. In *ICWE '06: Proceedings of the 6th international conference on Web engineering*. New York, NY, USA: ACM Press pp. 145–152.

Crume, Jim, Kevin Mukhar, James L. Weaver and Chris Zelenak. 2006. "EJB fundamentals and session beans." *JavaWorld Magazine* . Available at: http://www.javaworld.com/javaworld/jw-02-2006/jw-0213-ejb.html.

Hightower, Richard. 2004. *Jakarta Struts Live*. SourceBeat.

IndicThreads. 2006*a*. "The big MVC Framework merger (Struts + WebWork)." *IndicThreads Enterprise Java Station* . Available at: http://www.indicthreads.com/news/345/struts_webwork_merger.html.

IndicThreads. 2006*b*. "Struts Action Framework 2.0 should be released by August 2006." *IndicThreads Enterprise Java Station* . Available at: http://www.indicthreads.com/interviews/415/struts_action_framework_2_webwork.html.

Kodali, Raghu. 2006. "Interview of Raghu Kodali on EJB 3.0." Tech Talk Video of the TheServerSide.com. Available at: http://www.theserverside.com/tt/talks/library.tss#kodali.

Kurniawan, Budi. 2002. *Java for the Web with Servlets, JSP, and EJB*. New Riders.

Lerner, Reuven M. 2001. "At the forge: Enterprise Javabeans." *Linux J.* 2001(92):8.

Lerner, Reuven M. 2002. "At the forge: Entity beans." *Linux J.* 2002(93):7.

Lightbody, Patrick. 2005. "WebWork joining Struts." *OpenSysmphony Blog* . Available at: http://blogs.opensymphony.com/webwork/2005/11/webwork_joining_struts.html.

Massol, Vincent. 2005. "Building J2EE Projects with Maven." *O'Reilly ONJava.com* . Available at: http://www.onjava.com/pub/a/onjava/2005/09/07/maven.html.

Massol, Vincent, Jason va Zyl, Brett Porter, John Casey and Carlos Sanchez. 2006. *Better Builds with Maven*. Mergere Inc.

McClanahan, Craig. 2006. "Interview of Craig McClanahan on The Struts Framework." Tech Talk Video of the TheServerSide.com. Available at: http://www.theserverside.com/tt/talks/library.tss#mcclanahan.

Raible, Matt. 2006. "Comparing Web Frameworks Struts, Spring MVC, WebWork, Tapestry & JSF." Virtuas Presentation. Available at: http://www.chariotsolutions.com/slides/spring-forward-2006-web-frameworks.pdf.

Redmond, Eric. 2006. "The Maven 2 POM demystified." *JavaWorld Magazine* . Available at: http://www.javaworld.com/javaworld/jw-05-2006/jw-0529-maven.html.

Richardson, Chris. 2006. "Untangling enterprise Java." *Queue* 4(5):36–44.

Rubio, Daniel. 2006. "JSF frameworks: Shale and Seam." *JavaWorld Magazine* . Available at: http://www.javaworld.com/javaworld/jw-05-2006/jw-0515-jsf.html.

Shan, Tony C and Winnie W Hua. 2006. Taxonomy of Java Web Application Frameworks. In *ICEBE '06: IEEE International Conference on e-Business Engineering*. Los Alamitos, CA, USA: IEEE Computer Society pp. 378–385.

Sharma, Anil. 2004. "EJB 3.0 in a nutshell." *JavaWorld Magazine* . Available at: http://www.javaworld.com/javaworld/jw-08-2004/jw-0809-ejb.html.

Sierra, Kathy and Bert Bates. 2003. *Head First EJB*. O'Reilly.

Siggelkow, Bill. 2005. *Jakarta Struts Cookbook*. O'Reilly.

Smart, John Ferguson. 2005. "An introduction to Maven 2." *JavaWorld Magazine* . Available at: http://www.javaworld.com/javaworld/jw-12-2005/jw-1205-maven.html.

Smart, John Ferguson. 2006. "Get the most out of Maven 2 site generation." *JavaWorld Magazine* . Available at: http://www.javaworld.com/javaworld/jw-02-2006/jw-0227-maven.html.

Sriganesh, Rima Patel, Gerald Brose and Micah Silverman. 2006. *Mastering Enterprise JavaBeans 3.0*. Wiley Publishing, Inc.

Sun. 2006. *Enterprise JavaBeans Specification Documentation 3.0 Final Release*. Sun Microssystems. Available at: http://java.sun.com/products/ejb/docs.html.

Tiwari, Shashank. 2006. "Migrating EJB 2.x applications to EJB 3.0." *JavaWorld Magazine* . Available at: http://www.javaworld.com/javaworld/jw-08-2006/jw-0814-ejb.html.

XDoclet. 2006. *XDoclet2 User Documentation*. SourceForge. Available at: http://xdoclet.codehaus.org/User+Documentation.

Yuan, Michael Juntao. 2005*a*. "On the road to simplicity: JBoss 4.0 simplifies middleware development." *JavaWorld Magazine* . Available at: http://www.javaworld.com/javaworld/jw-02-2005/jw-0221-jboss4.html.

Yuan, Michael Juntao. 2005*b*. "Simplify enterprise Java development with EJB 3.0, Part 1." *JavaWorld Magazine* . Available at: http://www.javaworld.com/javaworld/jw-08-2005/jw-0815-ejb3.html.

Yuan, Michael Juntao. 2005*c*. "Simplify enterprise Java development with EJB 3.0, Part 2." *JavaWorld Magazine* . Available at: http://www.javaworld.com/javaworld/jw-09-2005/jw-0912-ejb.html.