

Thesis: JEE and JEE Compatible Frameworks

David Kelly

November 30, 2015

Markus Hofmann

Geraldine Gray

Dept. Informatics and Engineering
Institute of Technology Blanchardstown

Contents

1	Project Overview	5
1.1	Project Title	5
1.2	Project Type	5
1.3	Project Client	5
1.4	Project Supervisor	5
1.5	Project Manager	5
1.6	Project Timescale	5
2	Executive Summary	5
3	Introduction	6
3.1	Background	6
3.2	Existing Technologies	7
4	Main Research Question	8
4.1	Does the choice of framework affect the application?	8
4.2	Can a project accommodate a frameworks learning curve?	8
4.3	Can frameworks be compared?	8
4.4	What are the major differences between the researched frameworks?	8
4.5	Can one of the researched frameworks be implemented in a project	8
5	Benefits	9
5.1	Developer Benefits	9
5.2	Project Benefits	9
5.3	Future Benefits	9
6	Feasibility	10
6.1	Technical Requirements	10
7	Proposed Methodologies	10
7.1	Researching existing Java frameworks	10
7.2	Comparing frameworks	10
7.3	Prototype Implementation	11
7.3.1	Advantages	11
7.3.2	Disadvantages	11
8	Expected Results	12
8.1	Successful Research	12
8.2	Successful Application	12
8.3	Successful Project	12
8.4	Failed Project	12
9	Conclusion	13

10 Project Plan	14
10.1 Project table	14
10.2 Gannt Chart	15
11 Abstract	16
12 Introduction	16
12.1 General Topic	16
12.2 Introduction to Literature	16
12.3 Reason for Literature Review	17
13 Literature Review	18
13.1 Patterns of Cross Language Linking in Java Frameworks	18
13.1.1 Introduction to Cross Language Support	18
13.1.2 Choosing Frameworks	18
13.1.3 Defining and Testing	18
13.1.4 Research Paper Conclusion	19
13.2 A Presentation Framework to Simplify the Development of Java EE Application Thin Clients	19
13.2.1 Introduction to MVC Model	19
13.2.2 Researched Frameworks	20
13.2.3 Creating the JEE Application	21
13.2.4 Research Paper Conclusion	22
13.3 Developing Java EE Applications Based on Utilizing Design Pat- terns	22
13.3.1 Introduction to Application	22
13.3.2 Introduction to Design Patterns	22
13.3.3 JEE and Framework Selection	22
13.3.4 Design Pattern Selection	23
13.3.5 Research Paper Conclusion	23
13.4 J2EE development frameworks	23
13.4.1 Introduction to JEE and JEE compatible Frameworks . .	23
13.4.2 Why Use Frameworks	24
13.4.3 Reseach Paper Conclusion	25
13.5 Spring framework for rapid open source J2EE Web application development: a case study	25
13.5.1 Introduction to JEE Frameworks	25
13.5.2 Spring Framework	27
13.5.3 Case Study Development	28
13.5.4 Case study Specifics	28
13.5.5 Research Paper Conclusion	29
14 Research Paper Connection	29
15 Gaps in Research Papers	30

16 Literature Review Conclusion	31
17 Introduction	33
18 Proposed Methodologies	33
18.1 Tutorials	33
18.2 Framework Features	33
19 Use Cases	33
19.1 Diagram for Turotial 1	33
19.2 Diagram for Turotial 2	33
19.3 Diagram for Turotial 3	33
19.4 Diagram for Turotial 4	33
20 Sequence Diagrams	33
20.1 Diagram for Turotial 1	33
20.2 Diagram for Turotial 2	33
20.3 Diagram for Turotial 3	33
20.4 Diagram for Turotial 4	33
21 Tutorial Documentation	33
21.1 Documentation for Framework 1	34
21.2 Documentation for Framework 2	34
21.3 Documentation for Framework 3	34
21.4 Documentation for Framework 4	34
22 Conclusion	34
23 Introduction	35
24 Methodology	35
24.1 Choosing Frameworks	35
24.2 Framework Specifics	35
25 Spring Tutorial	35
25.1 Background	35
25.2 Documentation	35
26 GWT Tutorial	35
26.1 Background	35
26.2 Documentation	35
27 Hibernate Tutorial	35
27.1 Background	35
27.2 Documentation	35

28 Vaadin Tutorial	35
28.1 Background	36
28.2 Documentation	36
29 Conclusion	36
30 Introduction	37
31 Framework Comparrison	37
31.1 How to Compare Frameworks	37
31.2 Existing Frameworks	37
32 Community Support	37
32.1 Spring Forum Support	37
32.2 Hibernate Forum Support	37
32.3 GWT Forum Support	37
32.4 Vaadin Forum Support	37
32.5 External Forum Support	37
33 Prototype Implementation	37
33.1 Design UML	37
33.2 Process Documentation	37
34 Prototype Testing	38
34.1 Documentation	38
35 Conclusion	38
36 Introduction	39
37 Achievments	39
38 Personal Gain	39
38.1 Project Gain	39
38.2 Java Gain	39
39 Further Work	39
40 Conclusion	39
41 Images	40
42 Diary	40
43 code	40
44 Bibliography	40

*Introduction

1 Project Overview

1.1 Project Title

Research Application Development Frameworks for JEE Development, and Develop JEE Prototype with Framework

1.2 Project Type

Research and JEE Prototype application

1.3 Project Client

Institute of Technology Blanchardstown

1.4 Project Supervisor

Geraldine Gray

1.5 Project Manager

David Kelly

1.6 Project Timescale

14/9/2015 - 30/4/2016

2 Executive Summary

This proposal explores the potential to incorporate JEE compatible frameworks in to application development to reduce the development time and increase the application performance.

This proposal introduces what frameworks are and how they can be implemented. In order to choose the correct framework to implement in JEE app development it is important to research what components the framework manages.

This proposal suggests learning about four potential frameworks and building a prototype application with one of the frameworks.

A successful project would analyze cost, time, compare frameworks and examine their implementation within a JEE environment.

3 Introduction

The aim of this project is to research four popular Java Enterprise Edition compatible frameworks and develop a JEE prototype application from one of the researched frameworks. The Java EE platform provides an API and runtime environment for developing and running large-scale, multi-tiered, scalable, reliable, and secure network applications [1]. With many multifaceted components making up a JEE application it is easy for a developer to be obstructed from creating the application they wish and can find themselves bogged down in the technical processes of simply getting the application to function.

A framework's primary purpose is to aid and ease an applications development process. It should allow for an application to develop quickly and easily and should result in a superior finished application [2]. Upon researching these frameworks, I will evaluate their core features and analyzing how each framework is suited to be implemented within a JEE application.

I will develop a JEE prototype application using one of the frameworks I feel can further my understanding. While developing the application I will document and compare how the framework provides an advantage over the standard Java API, further I will asses the learning curve involved with implementing the chosen framework.

3.1 Background

Frameworks have been around since the release of J2EE (Java 2 Enterprise Edition) in late 1999. Struts framework was created by Craig McClanahan (later donated to Apache Foundation) to improve the development experience over pure Java Server Pages (JSP) utilization [3]. With the many updates and releases of what is now known as simply JEE (Java Enterprise Edition) so too came many frameworks looking to improve on standard JEE components.

Frameworks accommodate developers who do not have specific knowledge of all components of JEE application development. The ability to plugin a framework to a project can reduce overhead related to app development. Modern developers do not need to specialize in persisting data, database management, management of dependencies, security. Developers can simply select a framework that can manage these components and write programs that utilize these framework classes.

Choosing the correct framework to develop a JEE application can be difficult and there is no single correct answer. This project proposes to find out what popular current frameworks exist for JEE and when they should be incorporated in to a JEE project.

3.2 Existing Technologies

In order to select the frameworks to be researched we must look at some of the existing and more popular frameworks for JEE.

In May 2014 software development community www.zereturnaround.com [4] commissioned a survey to find out what Java frameworks are most used by current developers. The survey was carried out on 2164 Java developers. The results are shown in figure 3.1

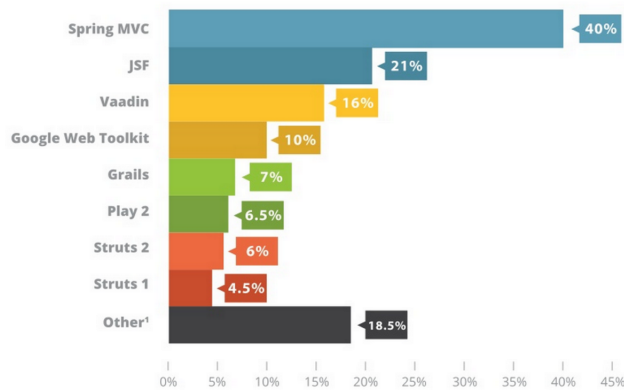


Figure 3.1

In Nov 2014 software developer blog www.vitalflux.com [5] conducted a survey of the most sought after positions in JEE framework development. Figure 3.2.

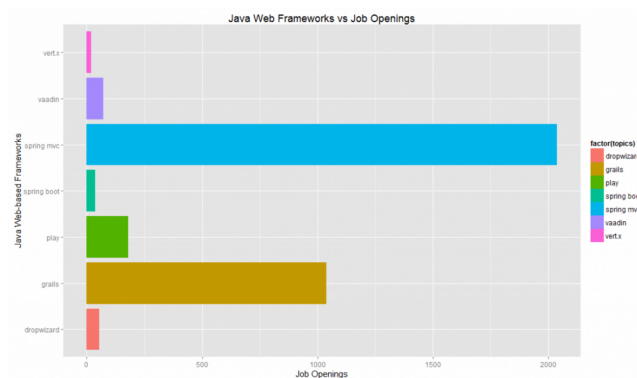


Figure 3.2

These websites are not the final say on the community of Java framework developers or the professional market that use frameworks. these surveys are good indications of what frameworks are popular, have large communities and are established in the current developer environment.

In choosing the frameworks to study I would lean toward cross referencing both of the surveys above and selecting frameworks present and well represented in both.

4 Main Research Question

4.1 Does the choice of framework affect the application?

With many different standardized frameworks available to incorporate into a JEE project, can the choice of framework impact the performance of the final application?

4.2 Can a project accommodate a frameworks learning curve?

Mastering a framework comes with a cost, is it worth learning an entire framework for a single project?

4.3 Can frameworks be compared?

What are there commonalities between frameworks that allow them to be compared?

4.4 What are the major differences between the researched frameworks?

From the four frameworks researched what area of application development to they best suit?

4.5 Can one of the researched frameworks be implemented in a project

Can a prototype JEE application be developed to display the abilities of the chosen framework?

5 Benefits

5.1 Developer Benefits

Understanding of existing frameworks

Knowledge of existing frameworks allow a developer to plan a project while incorporating a framework that can manage a component of the application. Understanding what frameworks do and how they can be utilized increases a developers' ability to create applications. Researching four frameworks will give a strong foundation on what frameworks are, what they do, how they can be incorporated and when.

Critical ability to compare features of frameworks

Critical analysis of frameworks and their features will be a benefit. Becoming familiar with frameworks and how they are comprised and implemented can further the ability of a developer to build an application employing the correct framework with maximum impact with the smallest overhead. It is not necessary to have large amounts of classes that are unimplemented if only few classes are needed

Learning how to develop with chosen framework while building prototype

After choosing the framework to develop the JEE prototype, the developer will learn valuable skills regarding the implementation of the framework. Using an IDE and downloading the plugin and classes needed then learning about the modules available to create a complete JEE application.

5.2 Project Benefits

Ability to develop application faster

Correctly implementing the framework in to the prototype project should accelerate application development while increasing the functionality of the final application.

Example of chosen framework in prototype application

Upon completion of this project an example application created with a chosen framework should show the capabilities and improvements possible when incorporating the correct framework in to the JEE application

Knowledge of other frameworks that have potential use

While this project intends to use one of the researched frameworks to create a prototype application, It is possible to incorporate multiple frameworks in to a single project. Knowing what frameworks can do will benefit the projects ability to incorporate more than one if extended or upon removal and replacement of existing framework.

5.3 Future Benefits

Creating a prototype application will show the capabilities of the framework. Further mastering of the chosen framework can be used on larger projects in future. After the initial cost of the learning curve for the first project, future

project development will be quicker and can use other frameworks touched upon in research project.

6 Feasibility

6.1 Technical Requirements

In order to undertake this project it is required that the lead role has experience with Java and has previously worked on projects of similar size. Researching four unique Java frameworks will require library resources and an aptitude for developing and learning simultaneously. The knowledge of a Java IDE is necessary and access to download plugins needed. A working understanding of JEE applications is required as the prototype will be developed in a JEE environment.

The project has a fixed timeframe marked with frequent deliverables and meetings to assess project progression.

I have worked on projects in the past developing applications through many different Java IDE's. I have produced documents and literature on topics such as Java GUI design, web based games built with PHP and I have spent 3 years developing with Java. I am currently studying in Institute of Technology Blanchardstown and have college access to resources including, computer labs, library resources and a project supervisor (MSc Geraldine Gray).

7 Proposed Methodologies

7.1 Researching existing Java frameworks

A literature review will be carried out on four of the chosen frameworks. The purpose of this literature review is to understand and become familiar with the features of each framework. Access to existing projects through public GitHub repositories can provide information on Java projects that use these frameworks and how other developers have incorporated them in to their applications. Researching current websites that implement Java frameworks will show a working environment of what each framework is capable of doing.

7.2 Comparing frameworks

Comparing characteristics of frameworks is a bases for overall understanding of their core features. Comparing the size of the framework plugin classes, The impact the framework has on the existing Java code, The learning curve involved in working with a framework, the removal of a framework from a project, how a framework manages dependencies, community support for the framework.

7.3 Prototype Implementation

The prototype JEE application should with developed with the waterfall SDLC model.

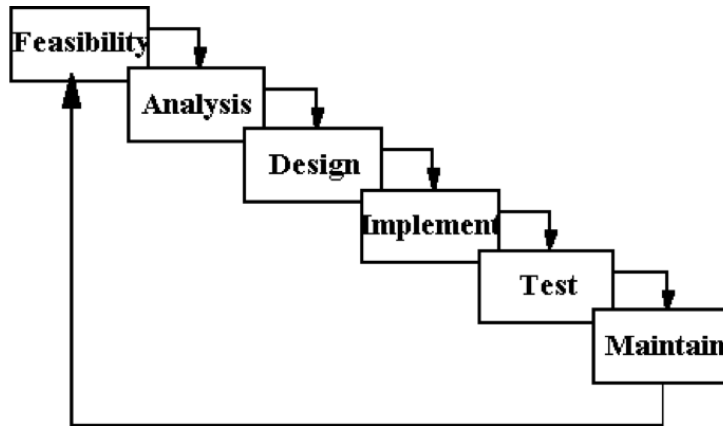


Figure 7.3

7.3.1 Advantages

- Simple and easy to understand and use.
- Easy to manage due to the rigidity of the model – each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.

7.3.2 Disadvantages

- Difficult to move back stages
- No working software is produced until late during the life cycle.
- Poor model for long and ongoing projects.

8 Expected Results

8.1 Successful Research

Successfully researching all frameworks will entail the lead researcher developing a firm understanding of each individual framework. Becoming familiar with the characteristics of the framework, critically being able to compare each individual framework and distinguishing their abilities within a JEE application. The research phase of the project should deliver the framework to be used in developing the prototype. Knowing what JEE application to be developed and how the framework will improve the development.

8.2 Successful Application

A successful implementation of the JEE prototype application should incorporate one of the four researched java compatible frameworks. The application will be fully functional with all components of a JEE application fully involved some at least one of which managed by the framework. The final version of the prototype will be submitted before the deadline on the 30/4/2016.

8.3 Successful Project

In order for the project to be considered successful, the research should improve the researcher's knowledge of Java compatible frameworks. Four frameworks in particular should be known with great detail. One of the frameworks should be used to develop a prototype application that displays the capabilities of the framework within a JEE environment application.

The entire development process need to be documented with industry standard code and a .war file that can be deployed on a Java server as a standalone.

8.4 Failed Project

A failed project can occur if the research phase does not correctly investigate four Java compatible frameworks. If the research process is too vague and knowledge of frameworks is not documented correctly.

The ability to create a prototype JEE application may not be hindered by the lack of research. However without a complete research phase to support the reasoning and choice of framework used to develop the prototype app the application will be considered unsuccessful. Should the final prototype application not be submitted before the deadline date, the project will also be considered unsuccessful.

9 Conclusion

In order to begin the next phase of the project it is important to understand the most important concepts of this proposal.

1. The project should center on the research of four Java EE compatible frameworks.
2. All unique features of each framework should be assessed and all commonalities compared.
3. After researching the frameworks, one should be chosen to develop in a complete prototype JEE application.
4. The prototype should follow the waterfall SDLC with frequent reviews from the project supervisor.

I have the ability and resources available to complete this project within the allocated timeframe following the timeline and gannt chart proposed.

10 Project Plan

10.1 Project table

Date	Task	Name	chk
21-Sep	2	Choose Supervisor	
28-Sep	3	Research Java EE	
05-Oct	4	Research existing java frameworks	
12-Oct	5	Select SDLC	
19-Oct		Submit Research proposal	
26-Oct		Research individual frameworks	
02-Nov	6	Evaluate Websites using JEE frameworks	
09-Nov		Literature review	
16-Nov	7	Choose framework to create prototype	
23-Nov		Search GitHub projects for project scale	
30-Nov	8	Begin Designing JEE app	
07-Dec		Design UML	
14-Dec	9	ORM Database design	
21-Dec		ORM Database design	
28-Dec	10	Begin coding app	
04/01/2016		Plugin framework	
11-Jan		Document system map	
18-Jan		Document task completion	
25-Jan	11	Compare code on researched frameworks	
01-Feb			
08-Feb			
15-Feb	12	Fill app with test data	
22-Feb	13	Choose testing process	
29-Feb			
07-Mar	14	Test app module connectivity	
14-Mar		Release alpha	
21-Mar			
28-Mar	15	Populate database and design website	
04-Apr		Release beta	
11-Apr		Documentation	
18-Apr	16	Test database	
25-Apr	17	Test session and security	
02-May			
09-May	18	Final testing	
16-May	19	Documentation	
23-May		Testing	
30-May		Final release - Deploy	

10.2 Gantt Chart

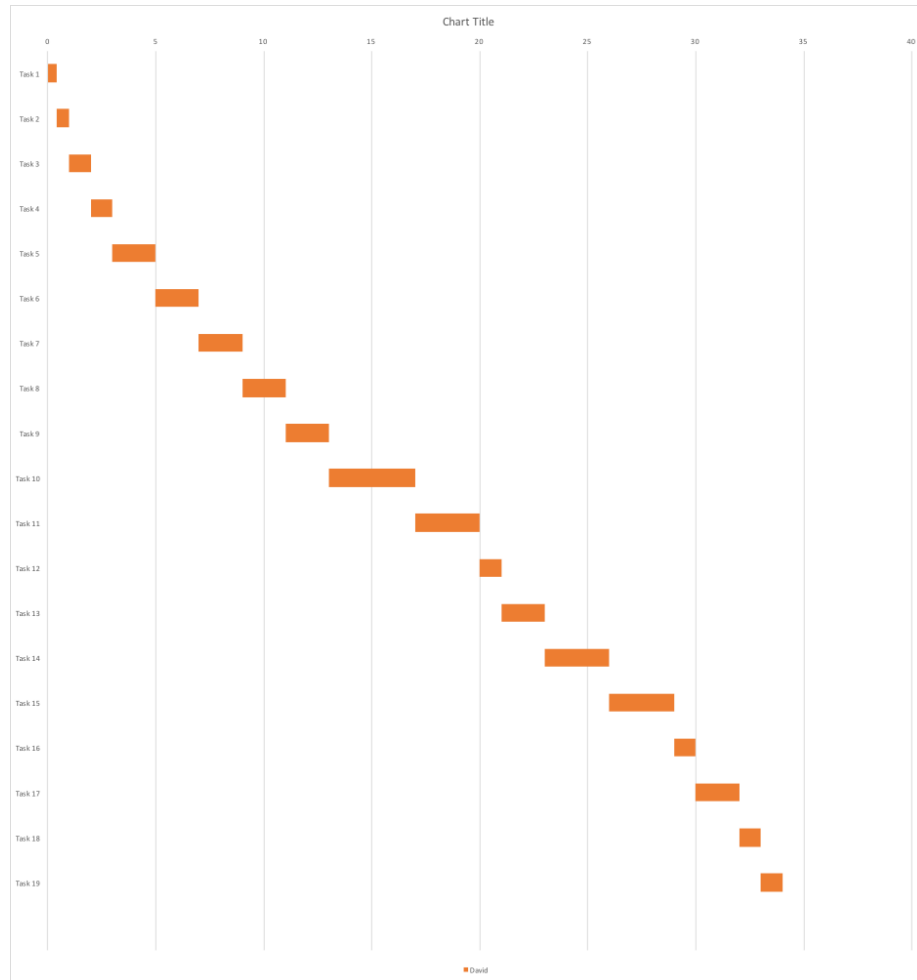


Figure 11.1

*Literature Review

11 Abstract

The purpose of this literature review is to analyse information gathered across five different papers related to Java compatible frameworks. The papers titled,

- “Patterns of cross-language linking in java frameworks” [9]
- “A Presentation Framework to Simplify the Development of Java EE Application Thin Clients” [4]
- “Developing Java EE Applications Based on Utilizing Design Patterns” [7]
- “J2EE development frameworks” [5]
- “Spring framework for rapid open source J2EE Web application development: a case study” [1]

Will help illuminate and further develop my understanding of JEE applications, JEE compatible frameworks, design patterns in Java and indepth knowledge regarding specific frameworks.

I will start by reviewing the paper with the least relevance and connect the points most closely related to my project. Linking features of each paper to display an understanding of what has been covered and what is undocumented in the field my project resides.

12 Introduction

12.1 General Topic

Java EE applications are large and often difficult to understand due to their complexity. Employing Java compatible frameworks in the development of an application can reduce the complexity and assist in creating a more efficient, cost effective application. It’s not enough to just begin developing an application with any compatible framework. Frameworks come in many different shapes, sizes, tools, components. Many factors must be considered in selecting the correct framework to incorporate in to the application development.

12.2 Introduction to Literature

“Patterns of cross-language linking in java frameworks” argues that little research has been carried out on the development of cross language links. During the development and maintenance of applications using a framework to connect multiple languages it is imperative to keep links intact. The inability to understand the connections in the framework where different languages meet affects productivity, efficiency and future development. The paper identifies common

patterns in cross language linking in the domain of Java frameworks. Although the aim is to further explain cross language linking, the authors develop through their main research question exclusively in Java.

“A Presentation Framework to Simplify the Development of Java EE Application Thin Clients” has the aim of ultimately creating a framework to develop JEE applications that support the presentation tier. In doing so the authors carry out an extensive survey on all existing presentation tier frameworks compatible with Java. In order to create the framework, they examine and compare features of Struts and Spring frameworks. Further this paper examines Java Server Faces (JSF) and JEE Design Patterns.

“Developing Java EE Applications Based on Utilizing Design Patterns” concerns the selecting and implementing of design patterns in Java EE applications. The paper describes the Model View Controller (MVC) with integration from Spring and Hibernate. Analyzing multiple design patterns and validate the feasibility of the framework model for implementation.

“J2EE development frameworks” was published in 2005 and discusses the topic of J2EE providing a lack of programming support for developers. This paper explains many of the factors that led to open source framework being created and implemented. Topics in this paper cover the emergence of open source frameworks, the reason why open source frameworks are needed and 3 popular open source Java compatible frameworks (namely Struts, Hibernate and Spring). Further the paper discusses what is next for open source frameworks and standard J2EE frameworks.

“Spring framework for rapid open source J2EE Web application development: a case study” compares the features of Spring and J2EE Enterprise Java Beans (EJB). The authors examine the EJB architecture necessary to run an EJB and lightweight container architecture which can run the Spring framework. Describing the goal of the Spring framework as developing an application as accurately, efficiently and as economically as possible.

12.3 Reason for Literature Review

I am currently a 4th year student compiling a thesis on Java compatible frameworks. In my project I am comparing four Java frameworks, cross referencing features, components, usability, community support and application development. After comparing the four frameworks I will implement one of the frameworks in to a prototype application to showcase the features of the chosen framework.

I hope that reviewing these five papers will further my understanding of the core principles of integrating a JEE compatible framework in to my prototype application. From the papers I have chosen there are clear connections between how to compare frameworks, why frameworks came to exist, how frameworks should be implemented and good design principles that coexist with framework development.

13 Literature Review

13.1 Patterns of Cross Language Linking in Java Frameworks

13.1.1 Introduction to Cross Language Support

To understand Mayer and Schroeder’s motivation for undertaking this project we must understand that very little has been written in the area of cross language support. This lack of support is detrimental to the developers and the systems stability [8][11]. Although Mayer and Schroeder set out to understand how links between artifacts (in this case class, subclasses and components) are formed among different programming languages. They also ask the question if links share characteristics and can be similarly described.

13.1.2 Choosing Frameworks

Instead of choosing two different programming languages fearing that the scale of the project may be too large. The project incorporates frameworks all written in one language but incorporating unique classes. For the chosen frameworks the authors used theoretical sampling. This allowed the inclusion of three well defined and well used industry frameworks, Android UI, Spring and Hibernate. This seemed a logical choice due to the good documentation, support and open source resources of all three frameworks.

13.1.3 Defining and Testing

Defining what a link is was an important step. Having tangible characteristics for a link allowed Mayer and Schroeder to draw UML diagrams displaying how links interacted, where they interacted and what was expected to happen. Without coding anything it could be possible to understand what a line of java code from a framework interacting with an artifact in the standard library might look like.

Three applications were implemented with each framework. The applications were connected to each framework respectively (OpenMeetings, Spring), (JTrac, Hibernate), (Wordpress, Android). Connectivity was checked to ensure that language adapters could be connected to the application to view the entire UML to ensure discovery any unforeseen links. Figure 3.1.1 shows the three frameworks and existing lines of code.

Understanding of how a Java object might call another Java object is key to the understanding of this project. Three patterns were defined to determine link categories. Eight sub patterns define the link characteristics further. Well structured definitions accommodated good documentation and clear link interactions.

An example of a Structural link would be if an Object in a Class belonging to OpenMeetings called an artifact of the Spring framework class it would use the defined “For and Find” pattern under Structural. What happens in this

TABLE I
MLSAs: LINES OF CODE

App. (Framework)	LOC/Artifacts (Java)	LOC/Artifacts (DSL)
OpenMeetings (Spring)	55698 / 238391	405 / 377
JTrac (HQL)	17737 / 82317	27 / 375
Wordpress for Android	20609 / 97007	3633 / 1173

Figure 3.1.1

interaction is the Object may call a Class in Spring (For) which can call an Object (Find) but that object may have a default constructor so another (Find) is needed with the correct parameters. Figure 3.1.2 shows the structural pattern break down.

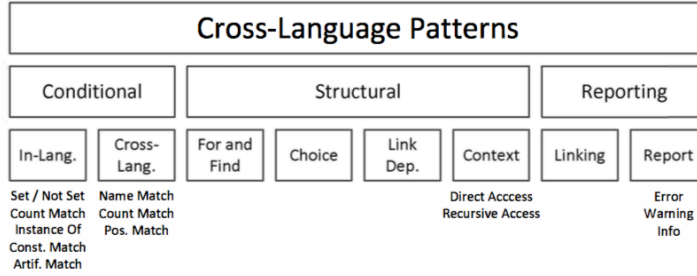


Figure 3.1.2

13.1.4 Research Paper Conclusion

The paper showed a clear understanding for what was required to identify the characteristics of cross language links between Java and domain specific languages in frameworks. Further the second research question was also able to be answered by categorizing the links regarding their properties.

13.2 A Presentation Framework to Simplify the Development of Java EE Application Thin Clients

13.2.1 Introduction to MVC Model

This paper details the building of a software framework to simplify development at the presentation layer of a Java EE application thin client. The paper begins by explaining design principles of presentation layer in applications namely the Model View Controller (MVC). The MVC is a design model used with graphical user applications to separate data access from business logic displayed to users. The authors agree that the MVC model brings a lot of advantages for application

design but concede that the flaws within the MVC are the reason they wish to develop a framework to overcome deficiencies in the MVC design.

13.2.2 Researched Frameworks

To begin developing the framework the paper describes existing Java compatible frameworks designed to simplify development at the presentation layer. As a starting point they survey and analyze these frameworks. The first framework reviewed is Struts 2 [2]. Struts architecture can be seen in Figure 3.2.1

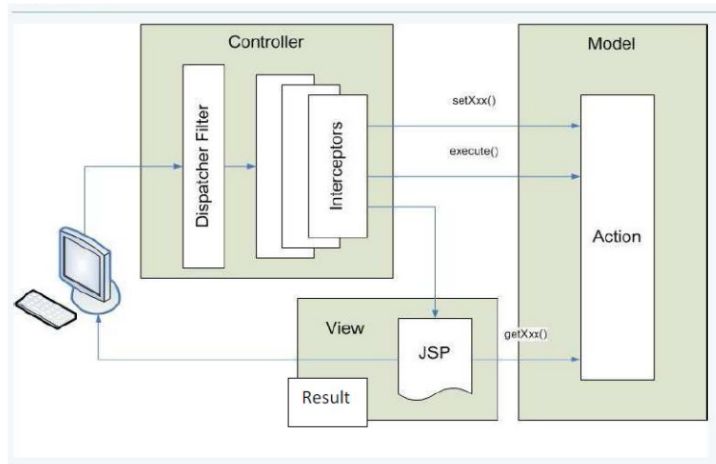


Figure 3.2.1

Apache Struts 2 is a framework intended to aid in the development of web applications. The paper suggests that Struts automates and reduces tedious development work. Struts also provides an architectural workflow within JEE applications making them more robust and flexible. The paper continues examining the features of Struts with further details regarding Struts ability to work with other frameworks, how actions are used as a core feature to provide functionality back to the user on the view end and finally discuss how Struts use tag libraries to validate inputs for return of dynamic data.

The next framework reviewed is Spring [3]. Spring uses simple JavaBeans instead of Enterprise JavaBeans. Reducing the number of complex methods as all objects in Spring are considered Plain Old Java Object (POJO) equal to a simple Java class. The authors provide the four strategies that spring proposes to reduce complexity. Use of templates, declarative code, weak coupling by injecting dependencies and light weight development with POJO's. Springs core architecture can be seen in Figure 3.2.2

Before moving on to design patterns the authors examine the final framework is Java Server Faces (JSF). JSF's are the Java standard class that was incorporated into Java EE 5. The purpose of JSF's is to allow development of Java web

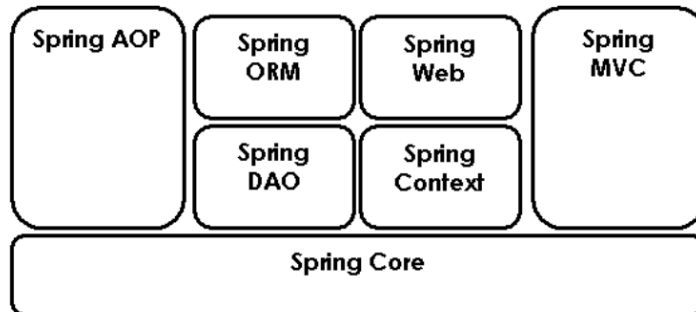


Figure 3.2.2

applications in a similar fashion to desktop applications. Webpage updates are managed by the framework which translates user actions in to messages sent from the server. Advantages with JSF include being the Java EE standard, having a set of UI tools and allow integration of third party components.

These frameworks provide features that the authors wish to implement in to the framework they are developing. Some features of certain frameworks lend themselves closer to developing at the presentation layer of JEE applications but no framework is a perfect match. Design patterns exist to provide solutions to problems with a given level of functionality[6]. In order to correctly complete the framework, the paper next describes the design patterns to be implemented along with the chosen framework.

13.2.3 Creating the JEE Application

The ability to define the requirements needed for the framework greatly improves the accuracy of the deliverable. The authors declare a list of features they wish to incorporate in to their framework. During the description of the existing frameworks, each examined framework had a bullet point list of advantages and disadvantages. So comparing the features they wish to develop in to their own framework against the features in the existing frameworks can help narrow the path of the project. This was a good step for project development. Clearly defining features to cross reference.

After listing the requirements for their framework it was concluded that the existing framework they would follow closest was Struts. In order to develop their framework further to accommodate the simplification of development at application layer the paper next describes incorporating the design models in to the framework design.

The framework developed used features of struts in forms and actions. With each action defining the result much like the JSF. Along with the code from existing frameworks, following design patterns allowed this framework to simplify development at presentation layer. The authors declare that their frame-

work achieves better results than simply using Struts. They pair their hybrid framework with the service to worker design model, allowing easy parameterized navigation through results, attributes, invocation and validation.

13.2.4 Research Paper Conclusion

I feel this paper gives good detail behind the processes need when developing an application. The authors correctly identified a design pattern to develop their application around. Researching and explaining the frameworks they chose and why they were chosen to benefit the development was well documented. Additional information was provided with patterns they chose not to use. The information was helpful in explaining why they didn't choose the intercepting filter or JSF.

The paper explains future work being related to developing larger full web applications in various domains to gain feedback within multiple areas of the framework. This shows that external of the allocated project time the authors had developed the idea of a complete application.

13.3 Developing Java EE Applications Based on Utilizing Design Patterns

13.3.1 Introduction to Application

This paper is based on a theoretical project to build a pilot management application in JEE. The application manages ship captains and drivers at a busy port in China. Storing data on captains working and available. Development of the application will be done through Spring and Hibernate frameworks along with several design patterns to be adapted in to the application.

13.3.2 Introduction to Design Patterns

The paper begins with discussion of design patterns explaining how important patterns fit in to the software development process and how design patterns are leveraged to solve system design problems. When selecting a design pattern the first consideration should be how does the design pattern solve the design problem. The authors suggest that after assessing multiple possible solutions, the option of a hybrid variable mixing components of multiple design patterns together should be considered.

13.3.3 JEE and Framework Selection

The project uses frameworks which the authors claim lends its self to greater abstraction and therefore further reuse. Using the MVC[10] pattern and Spring, an open source framework comprised of a layered architecture. This feature allows spring developers to be selective over which components are used and yet still maintains a robust framework for JEE applications. For storage the

project uses Hibernate framework as a relational mapping tool for Java. Hibernate accommodates standard Java expressions with abstraction, inheritance, polymorphism, and the Java collections framework. Hibernate greatly reduces development time relating to persistence data storage. Hibernate provides commands to store and retrieve data along with the mapping of classes to database tables.

The authors discuss the importance of EJB in JEE application design but with the rise of open source frameworks along come the abilities to create superior applications with less cost. Declaring the open source frameworks provide more tools, are more simple to learn, more simple to use and are in the end more flexible. The authors claim that Spring and Hibernate are the appropriate tools for developing this application.

13.3.4 Design Pattern Selection

When selecting their design pattern the authors suggest following a 7 step rule.

- Study the pattern
- Study the structure, participants and collaborations
- Look at concrete example of the pattern
- Choose names that are meaningful to application for classes
- Define the classes
- Define application specific names for methods in operations in the pattern
- implement the operations to carry out the responsibilities and collaborations in the pattern

13.3.5 Research Paper Conclusion

The paper gives good basis for designing application with frameworks. The paper tends not to go in to specifics of the frameworks chosen to develop applications but does explain how design patterns should be assessed when designing an application. Seven steps when choosing a design pattern are provided. Explaining their specific platform is well documented with code snippets and generic information about connecting the frameworks to the application. The paper does not explain any future implementation of the application for further development.

13.4 J2EE development frameworks

13.4.1 Introduction to JEE and JEE compatible Frameworks

In this paper the author discusses JEE compatible frameworks and reasons they have become so popular. Covering three well used frameworks in the Java developer's community the author discusses their features, their formative stages and

their place in Java application development. Building on top of Java Standard Edition (JSE), JEE fails to import features that allow JSE to work so well with relational databases. Which led to Sun Microsystems promoting Entity Java Beans (EJB) as a standard to aid in the development of applications working with Java Transaction API, Java Naming, Directory Interface etc. Unfortunately, EJB's remain cumbersome and complex in implementation. EJB's put constraints on inheritance and complications in unit testing.

The inability to provide a standard framework that meets all needs of Java application development has proved the impetus for creating open source frameworks that focus on tiers, and individual components of application development. It can be complex to develop a framework for building entire applications and have the end developers experience remain unaffected. Many smaller vendors opt against standard development tools and choose open source solutions that reduce cost and complexity during development. A framework is made of classes that allow applications to have utilize reusable design (normally for one tier).

13.4.2 Why Use Frameworks

The paper provides advantages of using frameworks in application development. The ability for developers to focus exclusively on code specific to their application and not infrastructure. Frameworks can provide structure to code and clear for future development. Easy to follow can promote best practice through community support and documentation. With these advantages along with popular frameworks like Hibernate, Spring and Struts having an important role in JEE development, developers have recognized an area that can reduce cost, provide support, reduce licensing fees all on open source frameworks that are better tested than in house code.

Going in to detail on the frameworks mentioned above, the author provides context and history for the frameworks. Beginning with Struts a web application framework developed in 2000. Deficiencies were evident in the standard Java Server Faces (JSF) model. The connection between business logic and HTML restricted features. Creating a web MVC to move business logic to its on class and separate the HTML. Struts solved this issue well enough to become the popular choice when developing web applications.

Hibernate is the next framework examined. Java Database Connectivity (JDBC) is the standard provided by Java for communication with databases. The JDBC forces developers to use relational concepts in code, this restricts object oriented design. Further, JDBC is complex and prone to errors when used directly e.g. resource management and exception handling. Hibernate provided a fully featured Object Relational Mapping (ORM) solution to persistence storage. Hibernate provided an intuitive model if not the most innovative. Building on an already existing ORM model but reinforcing and making a more robust framework for persistence solutions within application development.

With Struts and Hibernate in place a Java web application now just needed support for business logic. Java provides a EJB's but as the paper discussed

earlier, EJB's introduce restrictions and complications during testing. An open source framework exists namely Spring. Spring allows Plain Old Java Objects (POJO) along with Inversion of Control (IoC) and Aspect Oriented Programming (AOP) to decouple the classes from JEE environments. Allowing the POJO to be reused in many different environments. Spring is considered a lightweight container, combining IoC and services. Lightweight containers lend themselves to a much simpler unit testing experience and have been a major factor in the frameworks popularity.

13.4.3 Research Paper Conclusion

This paper provides information on how frameworks developed out of necessity. Three prominent frameworks in JEE application development are described in detail. Although the paper does not implementation of frameworks it is well documented what the frameworks features are and when they are used in applications. The paper doesn't provide any references but has been cited in many external papers. Concluding with what is next for frameworks gives an insightful view in to how frameworks are constantly evolving through open source rather than specifications. The author explains how framework features such as Aspect Oriented Programming (AOP) are becoming more important.

13.5 Spring framework for rapid open source J2EE Web application development: a case study

13.5.1 Introduction to JEE Frameworks

In this paper the authors argue that creating JEE applications as economically, efficiently and as accurately as possible often depends on implementing a framework during development. The implementation of a framework can increase productivity and decrease complexity[12]. The paper focuses on Spring, the foremost alternative to creating a JEE compatible web application without using Enterprise Java Beans (EJB). The paper described the architecture behind Spring and presents a Spring case study.

In order for an application to be considered JEE compliant, three core features of a JEE application must be featured namely the presentation layer, the business layer and the data storage layer. Before introducing the Spring architecture the authors present the descriptions for the EJB architecture and the lightweight container architecture to help familiarise the differences between both developments. Regarding the EJB architecture more detail concerning the three core features components are given. The presentation layer contains the User Interface (UI) tier, the business services contains the business logic methods to allow the UI to interface with the data access layer containing the Enterprise Information System (EIS). A lightweight container invokes an Aspect Oriented Programming (AOP) interceptor to provide services. Doing so enables the ability to add additional behaviour either side of the business methods execution. AOP interceptors are usable outside of any container therefore have no

need to use the containers API. Figure 3.5.1 shows EJB architecture with all 3 core components. While Figure 3.5.2 shows EJB lightweight container with MVC framework

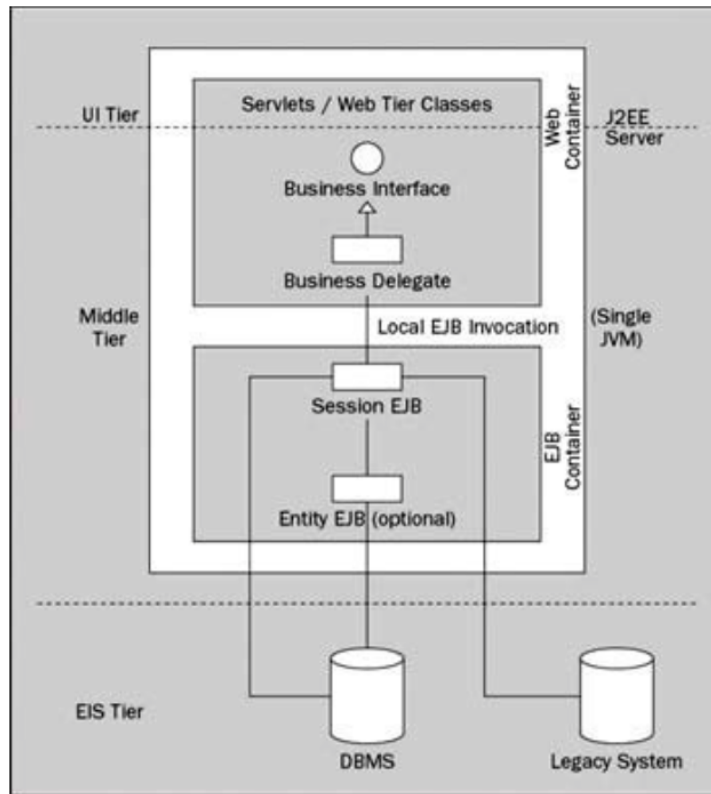


Figure 3.5.1

Comparing the differences between standard EJB development and development with a framework comes down to three important objectives, simplicity, testability and portability. EJB's are complex and require an EJB container meaning they need a high spec server that can handle additional resources. Lightweight frameworks only require a servlet engine reducing administrative complexity and addresses the issue of portability. When testing business logic EJB's become obstructed by the fact implanting classes are dependant on an EJB container. With lightweight framework being written as Plain Old Java Objects (POJO)'s it means testing can be carried out external of the application server.

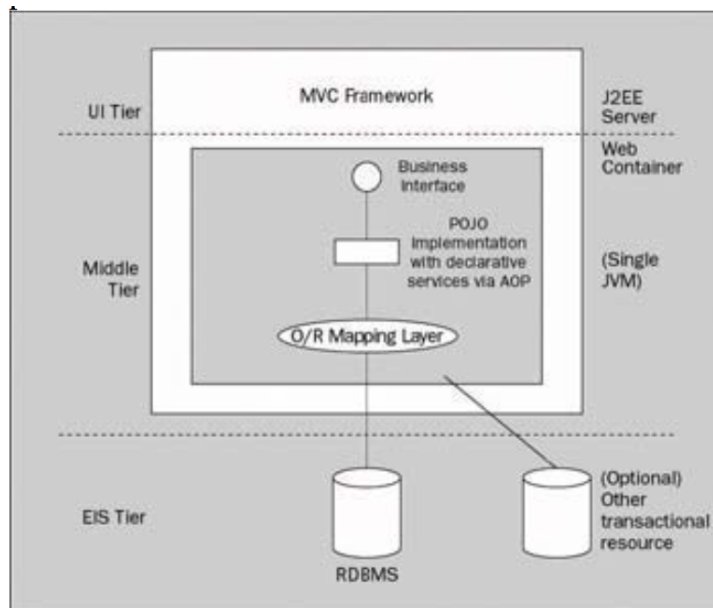


Figure 3.5.2

13.5.2 Spring Framework

The authors argue the case that Spring architecture provides a more intuitive developer experience citing claims from one of the authors that, he tried to develop an EJB application in the past and found the experience confusing and obtuse. Later the author developed an application implementing Spring and found he was able to produce more complex code faster and more easily.

Being able to develop code faster and with more ease is dependant on a key feature of Springs framework; Inversion of Control (IoC). The authors describe IoC with a Hollywood principle: “Don’t call us, we’ll call you”. In JEE development there are two types of Ioc, dependency lookups and dependency injections.

First the authors define what differentiates the dependency lookup from the dependency injection. Dependency lookups use container dependant APIs to look up resources. The container provides component call-backs and a lookup context. Managed objects must do their own lookups. This form of IoC differs from dependency injection where the objects are not responsible for looking up their resource or other dependencies.

The application code is separated from the resource lookups. This allows the objects to be used outside of the container with no need for APIs. Further information regarding types of IoC dependency injection are defined; Setter injection and constructor injection. Setter injections are invoked once the object

is instantiated by the container. Constructor injection needs arguments passed through the constructor to define the dependencies.

13.5.3 Case Study Development

Developing the case study in Spring allowed the authors to showcase the setter injection method. Although Spring accommodates both forms of dependency injection. Spring features an adaptable web application context concept which is a lightweight IoC container to adapt the web environment, providing the business layer containing business logic methods. Although the Spring business layer can be accessed from any web tier (Struts, JSF, WebWork, Tapestry) the authors decided to use Spring's web MVC after again finding development easier and more intuitive. The last step in deciding the components of the case study was implementing data access layer (EIS) in the form of a MySQL database. The authors studied Spring supported object-relational mapping (OR) frameworks and found Hibernate framework to be intuitive with a soft learning curve and well documented development support.

The authors next decide their development environment, providing a thorough description of all the tools they considered while building the case study. Without going in to too much detail I feel it is important to name some of the tools they decided to develop their application with. Eclipse was chosen as the Integrated Development Environment (IDE), CVS was used for version control, Maven was used to create the project administration site and TomCat was used as the servlet container.

13.5.4 Case study Specifics

The authors wished to create a basic but flexible issue tracking system to showcase the different functionalities of Spring. The biggest differences between EJB's and Spring are housed in the business logic layer, thus the paper goes in to further detail in this section. By using attribute-oriented programming, defining associations and collections with other objects of the issue class allow the Hibernate properties to be defined directly.

The next step in development was to create data access object and because Spring programs to an interface the objects must be extended. Mapping the Issue object to the Hibernate factory session allows Hibernate to access the Issue object through the Hibernate interface which in turn allows Spring access to the SessionFactory bean. Next, business service managers are created to decouple the business layer from the persistence layer. With no reference to a particular persistence strategy, the ability to change the persistence layer is made easier.

Using an AOP interceptor to handle transactions from the Issue manager class accommodates access to resources for users of privilege "admin". This interceptor is defined as a property and added to the Spring bean where the value is declared. The ability to use AOP for security and transaction management is one method for Spring to overcome the need for EJB functionality without using EJB.

13.5.5 Research Paper Conclusion

The paper concludes with the authors declaring the use of a lightweight container (in their case Spring) has definite advantages over standard EJBs in most cases. The authors recognise features of Spring that when compared to EJBs show clear development advantages. Spring does not need an application server which removes the over head of server administration and allows for application portability. Using Spring accommodates code reuse because code written in EJB can only run in the EJB container. This feature prevents EJBs being tested outside of the EJB container which is a problem during unit testing. Lastly the authors describe Spring as non invasive as it uses the dependency injection IoC pattern.

14 Research Paper Connection

Starting with the paper least relevant to my project I will discuss “Patterns of cross language linking in Java frameworks”. The authors go in to great detail on how frameworks connect and the path in which object calls traverse links in the languages. For my project the strongest connection is the core understanding of how frameworks operate together. The authors display a strong grasp of Java programming through their methods and testing of object calls. Reading this paper furthered my ability to conceptualize what is happening when frameworks call one another.

Although the first paper touches on frameworks in Java the second paper I reviewed “A Presentation Framework to Simplify the Development of Java EE Application Thin Clients”, gives much more information regarding the implementation of frameworks. One common framework is discussed in both papers but in very different ways. Spring is described in a far broader scale in the second paper, the authors detail how four features of spring reduce complexity against standard EJBs. A strong common thread between the majority of papers reviewed first appear here. The ability to compare frameworks is core to my thesis and the reason I have reviewed these papers.

Unlike the first two papers I reviewed the third paper actually describes implementation. “Developing Java EE Applications Based on Utilizing Design Patterns” focuses on design patterns which is relatively untouched in the other papers (but we will go in to that later). In this paper the authors create a JEE application implementing Spring and Hibernate frameworks. This again is third time Spring has appeared in my literature review and the second time Hibernate. Spring is a well documented and well supported open source framework. The authors decide not to spend too much time detailing Spring or Hibernate but tend to focus on their specific implementation. Good detail with code are provided showing how their application was designed with a MVC design.

The fourth paper I reviewed focuses on JEE frameworks. “J2EE Development Frameworks” is another step in the understanding of JEE frameworks, although no code or application examples are provided. Three frameworks are

described in broad but well informed detail. Struts, Hibernate and Spring are all on display in this paper. The third time Hibernate has appeared but this time with more information regarding its formative years and its place in JEE applications. The paper is very closely related to my thesis project and provides an overview of JEE frameworks, their features and from reading this paper I gained the ability to assess frameworks and their functions with JEE applications.

The final paper I reviewed was “Spring Framework for rapid open source J2EE Web Application Development: A case study”, this is the paper I focused on as I felt it was closest related to my thesis project. I will be developing a prototype application using Spring, with this paper going in to great detail comparing Spring to standard EJBs I felt it was a good opportunity to understand the Spring framework further.

As I mentioned earlier design patterns would come up later and after reviewing all five papers I can see that design patterns have a large roll in JEE application development. Three of the five papers I reviewed develop JEE applications with Spring and the MVC design. Further, three of the applications developed also implement Hibernate.

The core connection through all of these papers is the study in to JEE compatible frameworks and design patterns. Most notably the frameworks Spring and Hibernate along with the MVC design pattern.

15 Gaps in Research Papers

I feel very little gaps were left in the research paper “Patterns of cross language linking in Java frameworks”. The authors explained their main research question very clearly and developed methods to answer such question. The detail provided is concise and methodical. Perhaps the ability to use a different programming language could have been closer to the true question but still different frameworks have different classes and the interaction of these classes proved a similar enough comparison to garner the expected results. The findings are well documented, well structured and had similarity across all three applications proving their methods to be correct.

In the research paper “A Presentation Framework to Simplify the Development of Java EE Application Thin Clients “, the goal was clear, to create a unique framework for the presentation layer in a JEE application. The authors researched and compared frameworks thoroughly aiding in their ability to select features from many frameworks and create their own. The authors created a successful implementation of a framework and also asked the question of future development. Although originally being developed for a prototype application the authors felt their framework simplified and made it easier to create applications, the next step was to create full applications.

The research paper creating a pilot management system for a port in china “Developing Java EE Applications Based on Utilizing Design Patterns” went in to good detail regarding design patterns. The research question asking if it could be built was answered. However little code was provided and no area of

future development was explored.

The fourth paper reviewed “J2EE Development Frameworks”, contained lots of detail regarding JEE and JEE frameworks. The explanation of frameworks was clear and in depth. With out showing code snippets or examples I feel the author clearly conveyed a strong knowledge for JEE frameworks. The questioned was asked and answered regarding the future of JEE frameworks. Perhaps the author could have provided references and citations for the facts presented but as a basis or broad analysis for frameworks I felt the paper was strong.

The final paper reviewed was the paper I focused most of my attention on “Spring Framework for rapid open source J2EE Web Application Development: A case study”. This paper was closest to my thesis and I felt it had lots of information regarding implementation, assessing Spring against standard EJBs. The differences between EJB containers and lightweight containers. How dependency injections and dependency lookups are processed was very clear and helped my understanding of IoC. The paper presented lots of code snippets explaining though example how the framework operates. The main research question was answered; lightweight containers have advantages of EJB containers restricted to connect API’s. If gaps existed in the research paper they would be due to the lack of external reference to back up information provided.

16 Literature Review Conclusion

Frameworks are a huge component of application development. Choosing the correct framework before developing can be key to an applications success. Through this literature review I feel I expanded my ability to compare frameworks against one another but also compare them against the JEE standard EJBs. It’s clear that for many applications the over head need to implement EJBs is simply not needed. Applications can be developed, easier, faster and better equipped for test and portability outside of the JEE standard.

In creating applications, it is not only the framework that affects the development phase but also the design pattern. Some of the papers I reviewed went in to great detail on design patterns and the importance of creating a strong model using critical analysis of the classes needed in the application.

References

- [1] John Arthur and Shiva Azadegan. Spring framework for rapid open source j2ee web application development: a case study. *Software Engineering and Artificial Intelligence and Networking and Parallel/Distributed Computing, 2005 and First ACIS International Workshop on Self-Assembling Wireless Networks. SNPD/SAWN 2005. Sixth International Conference on*, pages 90–95, May 2005.
- [2] D. Brown, M. D. Chad, and S. Scott. Struts 2 in action. *Dreamtech Press*, 2008.
- [3] Walls C. Spring in action. *Manning*, 2008.
- [4] Santi Caballé, Jose-Arturo Ortega, Josep-Maria Camps, Leonard Barolli, Elis Kulla, and Evjola Spaho. A presentation framework to simplify the development of java ee application thin clients. *Complex, Intelligent and Software Intensive Systems (CISIS)*, pages 421–426, 2014.
- [5] Rod Johnson. J2ee development frameworks. *Computer*, 38(1):107–110, 2005.
- [6] C. Larman. Uml y patrones. *Prentice Hall*, 2004.
- [7] Shuang Liu and Peng Chen. Developing java ee applications based on utilizing design patterns. *Information Engineering*, 2:398–401, July 2009.
- [8] P. Mayer and A. Schroeder. Cross-language code analysis and refactoring. *Proceedings of SCAM*, pages 94–103, 2012.
- [9] Philip Mayer and Andres Schroeder. Patterns of cross-language linking in java frameworks. *Program Comprehension (ICPC)*, (13):113–122, 2013.
- [10] W. Pengfei and H. Zhiming. Application of three popular open source software in mvc. *Journal of Jiangxi University of Science and Technology*, 27(6)(35-38), 2006.
- [11] R.-H. Pfeiffer and A. Wasowski. Taming the confusion of languages. *Proceedings of the ECMFA*, pages 312–328, 2011.
- [12] Johnson R and Hoeller J. Expert one-on-one j2ee development without ejb. 2004.

*Analysis and Design

17 Introduction

Provide introduction in tot he the analysis and design phase. Discuss how I will assess the chapter and what frameworks i will be using

18 Proposed Methodologies

Discuss the tutorials I will be following to create small Java applications with each of the four frameworks

18.1 Tutorials

18.2 Framework Features

19 Use Cases

Draw use case diagrams for each tutorial. This will give visual aid to help understand the process involving each framework

19.1 Diagram for Turotial 1

19.2 Diagram for Turotial 2

19.3 Diagram for Turotial 3

19.4 Diagram for Turotial 4

20 Sequence Diagrams

May or May not need depending on how complex the tutorials for each framework are.

20.1 Diagram for Turotial 1

20.2 Diagram for Turotial 2

20.3 Diagram for Turotial 3

20.4 Diagram for Turotial 4

21 Tutorial Documentation

Document how all details about what the tutorial should achieve and how it's planned to achieve it. try get tutorials that showcase frameworks specific ability

21.1 Documentation for Framework 1

21.2 Documentation for Framework 2

21.3 Documentation for Framework 3

21.4 Documentation for Framework 4

22 Conclusion

Provide a compilation of how this chapter took shape, how it evolved and how it changed Did I meet my goals and did everything work?

*Implementation

23 Introduction

Provide introduction to the chapter explain in brief what will happen in this chapter.

24 Methodology

Discuss that I am choosing 4 tutorials to develop 4 small applications specific to each framework. Each tutorial will showcase the frameworks features

24.1 Choosing Frameworks

24.2 Framework Specifics

25 Spring Tutorial

Develop JEE application with small to medium size Spring tutorial. Document process

25.1 Background

25.2 Documentation

26 GWT Tutorial

Develop JEE application with small GWT tutorial Document process

26.1 Background

26.2 Documentation

27 Hibernate Tutorial

Develop JEE application with small to medium size Hibernate tutorial Document process

27.1 Background

27.2 Documentation

28 Vaadin Tutorial

Develop JEE application with small Vaadin tutorial Document process

28.1 Background

28.2 Documentation

29 Conclusion

Discuss the chapter and how the development of the four applications took shape. Explain difficulties, Changes, easy part.

*Testing and Evaluation

30 Introduction

Discuss the chapter and how I am comparing the tutorials I made last chapter. How I plan on comparing, Developing one of the frameworks in to a bigger application

31 Framework Comparrison

Comparing the frameworks by intuitive, learnable, speed of coding, complexity, size of classes, community support, removal of code affect

31.1 How to Compare Frameworks

31.2 Existing Frameworks

32 Community Support

Talk about the evaluation of community support after asking questions in frameworks relative forums and open forums like stack overflow

32.1 Spring Forum Support

32.2 Hibernate Forum Support

32.3 GWT Forum Support

32.4 Vaadin Forum Support

32.5 External Forum Support

discuss open forums not directly associated with any framework

33 Prototype Implementation

Develop a larger application in either Hibernate or Spring (probably). Application doesnt have to be huge but should be prototype for this project

33.1 Design UML

use case, sequence diagram, EER diagrams

33.2 Process Documentation

Detail steps durning development

34 Prototype Testing

Unit test the application, Web test the application, Fix potential bugs

34.1 Documentation

35 Conclusion

Describe the components of this chapter, difficulty comparing frameworks, community support and the development process for project prototype

*Conclusions and Further Work

36 Introduction

Discuss this chapter and how it took shape over the last 4 chapters.

37 Achievements

What has gone right with the project?

38 Personal Gain

What have I learned and how have I furthered my understanding of frameworks and testing?

38.1 Project Gain

related to working on project

38.2 Java Gain

related to further understanding of Java

39 Further Work

Develop prototype into full application? Continue comparing aspects of frameworks more in depth with perhaps further statistical analysis

40 Conclusion

How the project went from start to finish, am I happy with outcome

Appendix

41 Images

Images of UML and screenshots of web applications

42 Diary

Potentially upload diary of each weeks tracking

43 code

Upload source code for all 4 tutorials and final prototype

44 Bibliography

references