Abstract

The software industry is a very dynamic market that forces software companies to react as fast as possible and to adjust their software products to the actual market requirements. To enable this, one of the most important factors of software development is the underlying architecture of the software and component reuse. An enterprise application is very valuable for the enterprise. Enterprise applications usually involve persistent data and many people access data concurrently. There is also a rich set of graphical user interfaces to offer the users a convenient way to do their work. In addition, it is also necessary to integrate other enterprise applications scattered around the enterprise.

With the rise of object-oriented thinking, frameworks became more popular and sprung out then never before. A framework is an approach of software reuse and enables developers to establish the software on an abstract layer. On the web there are a lot of frameworks for web development. Currently frameworks include Java Server Faces, Struts, Spring, Hibernate, Cocoon, OpenLaszlo, IBJM, Maven and many others. Software developers recognize that due to the use of such frameworks most of the code (or an application becomes declarative which enables software companies to react fast to new requirements in their software product. To develop successful framework-based applications, developers should know what frameworks are and how to deal with them. This is an important part, because the use of frameworks influences the underlying architecture of the application. It depends on the requirements of an application, which set of frameworks will be used. Hence you take a different set of frameworks for a server side application than for a rich internet application (RIA).

This diploma thesis gives an introduction to the world of frameworks related to enterprise web application development and how to use and combine them. Therefore, a couple of frameworks will be evaluated by studying their capabilities, each one pursuing different goals. The diploma thesis is attended by a practical example which demonstrates the employment of the frameworks in an enterprise application, and explores their compatibility and practical usability. This practical example uses a set of frameworks to make the software components maintainable, re-useable, extensible, and configurable through declarative programming.

Open Laszlo

OpenLaszlo is an open source technology with the goal to develop Rich Internet Applications. In October 2004 the Laszlo Presentation Server became open source and is now available in version 3.2. Since version 3.2 OpenLaszlo can produce both Flash and DHTML, from a single LZX source. In the case of Flash this technology uses the flash player as common rendering engine for their components. Behind the engine a set of XML files are consumed that define the components. Applications in OpenLaszlo are written in a Laszlo specific markup language called LZX. These LZX files consists of several XML tags and a Java script code. OpenLaszlo provide mechanism for animation, layout, data binding, server communication and declarative UI.

JavaServer Faces

JSF is the first official standard for web application development and was developed under the Java Community Process (JCP) as JCP 1.2. The first early draft of the specification was released in September 2002. In March 2004 the final release of the JavaServer Faces specification has appeared J2EE consists of three parts:

- A set of prefabricated UI components
- A component model that enables third party developers to supply additional components
- JSF is based on technologies like JavaServer Pages (JSP), Java Servlets, Tag Libraries and XML.

Spring

The key points of Spring are:

- Lightweight Spring is lightweight in terms of both size and overhead. Spring consists of several modules that can be used as required. These modules are distributed as JAR files. Furthermore, Spring is a no invasive objects in a Spring enabled application, typically have no dependencies on Spring specific classes.
- Inversion of control Applying IoC, the container gives the dependencies to the object at instantiation with waiting to be asked.
- Aspect-oriented programming Especially in Version 2.0 the framework comes with rich support for aspect-oriented programming that enables cohesive development by separating application business logic from system services.
- Container Spring is a container in the sense that it contains and manages the lifecycle and configuration of application objects.
- Framework Spring makes it possible to configure and compose complex applications from simpler components.
- Easy access to other APIs like EJB, JSP, and others.
- Integration of other frameworks, like Hibernate, EJBs, JSP or Struts.

Spring Web Flow

MVC frameworks, such as Struts or JavaServer Faces, are not well designed to image business processes. Building a process that takes place over a series of steps with Struts, results in a crowd of struts configurations. To facilitate such scenarios, Keith Donald organizes Spring Web Flow. Spring Web Flow is one of the spring web components and focus on the definition and execution of page flows within a web application. Spring Web Flow do not replace existing MVC frameworks. Furthermore it can be used to complement the framework with a flow manager.

There are some major benefits coming with Spring Web Flow:

- Flow definitions should be naturally decipherable by business analysts.
- A flow definition should be easily engeneerable to from a visual diagram, such as a UML state diagram.
- Flow definitions should be self-contained
- A flow execution and its associated artifacts should be easily testable in isolation
- Build on Spring MVC, Struts, or other MVC Frameworks

Struts

The Struts project was born in May 2000 by Craig R. McGlashan and a version of MVC [18] that was adapted for web development, called Model 2. Struts 1.0 was finally released in 2001 in the form of the Jakarta project from the Apache Foundation with 26 java packages and more than 210 java classes. The Struts framework is one of a couple of simple web frameworks available on the internet. The Struts framework provides built in support for many common elements of web application like internationalization, exception handling, message and error logging.

Hibernated

Hibernate is one of the leading ORM technologies for enterprise development and joined. Hibernate is also a Java API (Application Programming Interface) that allows developers to interact with relational databases to persist objects to a database. Hibernate consists of two parts: the persistence layer and the object-relational mapping framework.

Spring

Spring is no intrusive: objects written in Spring are typically have no dependencies on Spring specific classes. The Spring framework provides built in support for many common elements of web application like internationalization, exception handling, message and error logging.

Presentation Layer

Controller Layer

Business Layer

Data Mapper Layer

Data Layer

Possible Combination