

Introduction to JBoss Seam

Ashish Kulkarni

01 October 2008

www.ebizss.com

1



How this presentation is organised

- A brief history of Java Web Applications
- What else has happened to the Web
- Where does JBoss Seam fit
- Demo
- Questions
 - Welcome anytime as long as you can accept the following answers:
 - I don't know but I will find out
 - Can we park that question and take it up after this presentation?

01 October 2008

www.ebizss.com

2

Gavin King and Pete Muir have been making a lot of presentations on JBoss Seam.

This presentation tries not to duplicate any of the concepts or ideas mentioned by either of them.

Instead, this presentation gives an insight into Seam from an Analyst and Developer's perspective.

I am assuming that the target audience has got some experience in developing Web Applications using Java.



A brief history of Java Web Applications

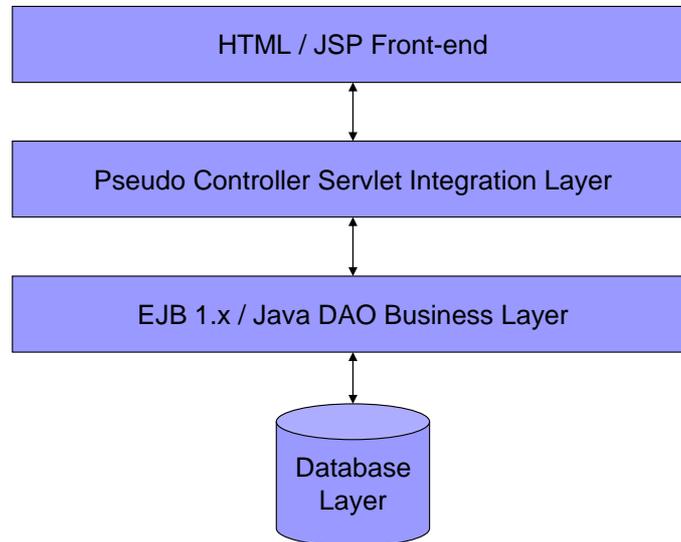
- Java Web Applications have been around for years now.
- They have undergone a radical change from the simple JSP / Servlet days to Frameworks allowing the use of MVC and similar design patterns.

The first generation of Java based Web applications had a simple HTML front-end that did HTML FORM posting to Java Servlets.

The arrival of JSPs on the landscape simplified Java Web Application development significantly. We could now embed Java scriptlets inside HTML code instead of embedding HTML code inside a Java class. And although at runtime, the JSPs get converted into a Servlet, it is this simplicity that is to this day at the heart of most frameworks such as Struts that rely on the use of Tag Libraries within JSPs.

Emergence of MVC frameworks such as Struts marked a revolution in the approach Java Developers took to developing Web Applications as they no longer had to develop a framework to manage the flow-of-control and validations any more. Instead, they could rely on a robust and freely available framework to do that.

The Model 1 Architecture



01 October 2008

www.ebizss.com

4

This slide describes the Model 1 Architecture used by the first generation of Java Web Applications.

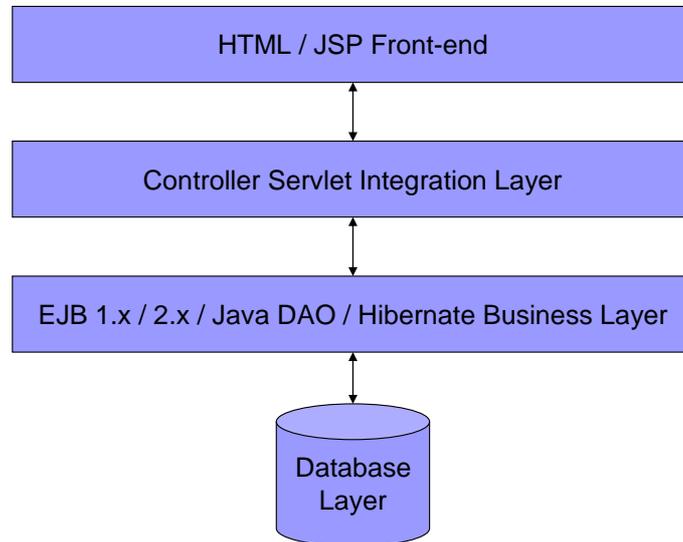
These had HTML or JSP front-end. The middleware was either a JSP or a Servlet that did the processing of submitted form data.

For this, it mostly relied upon the back-end Java Classes that usually implemented some form of a DAO mechanism.

The work of this Servlet was to convert the posted FORM data into Java objects understood by the back-end layer and vice versa.

This Servlet also had to do validations. Response messages were handled either by means of request attributes or by means of a request redirection or both.

The Model 2 Architecture - MVC



01 October 2008

www.ebizss.com

5

While the basics remained the same in Model 2, with the advent of complex frameworks such as Struts, most of the work was now delegated to the framework.

Under the hood, the framework still did input validation, request redirection and so on. But this was now hidden from the developer. The developer had to simply create xml configuration files.

As the popularity of these frameworks grew, tools quickly evolved to automate tasks such as creation of front-end, middleware and back-end artifacts and management of these xml configuration files.

Issues with traditional approach

- Too many artifacts
- Struts – JSPs, Action Classes, ActionForm Classes, validation.xml, struts-config.xml
- Hibernate – .hbm.xml file for every database table
- Integration Layer converting front-end java objects to business layer java objects and vice versa (too much plumbing code)
- Time consuming even with the best tools
- Ever-changing business requirements

01 October 2008

www.ebizss.com

6

Although, the use of frameworks such as Struts streamlined or standardised Java Web Applications, they also brought the need to create and maintain various artifacts.

We need Java Objects that can hold data being displayed on and/or being posted by the front-end. We need to specify validation rules for that data. We need to specify messages to display when these rules fail. And so on.

A lot of this can be automated by the proper use of correct tools.

In fact, at one point, I even initiated a project within my company to read database metadata and create JSPs, XML sciptlets and Java Classes for use with Struts with some elementary user input.

Number of artifacts is just one issue. From an architectural point of view, there is another bigger issue. The frond-end is unaware of the existance of back-end and vice versa. These two layers exist without the knowledge of each-other. And it is up to the integration layer to cast objects from back-end to front-end classes and back.

In fact, in most projects, that all the Integration layer really does. Convert objects, serialize/deserialize objects and pass them back and forth between front-end and back-end.

What else has happened to the Web

■ PHP

- Very easy deployment
- Agile

■ Ruby-on-Rails

- Agile
- Generates Scaffolding Code
- Deployment not as easy as PHP (debatable)
- Uses ActiveRecord Design Pattern

01 October 2008

www.ebizss.com

7

While we Java Developers and Architects were busy churning out tonnes and tonnes of repetitive plumbing code, the world had started to move on.

Complexity of deployment of a Java Web Application was and still remains a burning issue.

PHP came up with a brilliant solution - simply let the web server interpret PHP pages. As a result, a website created using PHP can be simply copied into the deployment folder and hey presto!!! Its available to the world!!!. No need for complex deployment descriptors or server.xml or similar files to wrestle with.

Testing a PHP web application is really simple. If you make a change, simply redeploy and retest. No need to shut down or restart server!!!

And then came Ruby-on-Rails. Ruby has been around for as long as Java. It was the advent of Rails that really made it popular.

Rails framework has tools that can generate most of the scaffolding code required for an RoR web application.

Also, its Active Record design makes it possible to have back-end objects that know how to do CRUD operations on the underlying tables in the database.

There have been a few other languages/frameworks that started to contend for the spotlight too such as Python.



Front-ends are changing

- More and more professional web applications are looking to provide Richer Experience
- There is an emergence of RIAs such as Flex
- Increasing number of websites are looking to provide AJAX functionality

01 October 2008

www.ebizss.com

8

When we moved from traditional Client/Server applications to Web applications, we had to sacrifice the richness of the Clients.

Even the thinnest of the clients in the client/server era were capable of rich user experience.

With the passage of time, the focus has started shifting to having the same richness of the user interface in web applications.

Adobe Flex and AJAX (Asynchronous JavaScript and XML) are two of the most talked about technologies that developers are coming to rely on to make web application interactions richer.



End users are still the same

- More demanding
- Smaller budgets
- Tighter deadlines
- “I want it all, I want it yesterday and I want to pay as little as possible”
- Low cost of ownership
- Low cost of maintenance

01 October 2008

www.ebizss.com

9

Most of us have heard it all before. And I am sure, we will keep hearing it until the day we retire!!!

The budgets are getting tighter. The deadlines are getting deadlier. The requirements are soaring.

Customers want web applications developed faster than ever before and for fraction of the price.

They have also become smart about cost of ownership. They don't want to pay an arm and a leg to host and support their web application.

If you put yourself in their shoes, you would want the same. Won't you?

After all, why do we use Java? Because JDK is free. Why do we use tools such as Eclipse? Again, because they are free. And the list goes on.

And the first time we find a bug in one of our free tools, we make a point of reporting it and chasing it up.

Where does JBoss Seam fit

- Generates of Scaffolding code
- Agile
- Collapses the need to have an Integration Layer
- Provides dependency bi-jection
- New variable scopes – Conversation and Business Process
- Works well with JSF and JPA
- JSR-299 – WebBeans (Seam Core) to be part of Java EE

01 October 2008

www.ebizss.com

10

Lets face it. We are not just looking for something free. We are looking for something free that frees up our time too and doesn't make us wrote more code.

If there is technology out there to generate scaffolding code, why not have it in Java? After all, at over 20%, Java does lead the TIOBE index!!!

How about some agility? If we are able to test what we are developing quickly, we can find and fix any bugs that much quicker. The only trouble is it is too addictive.

If all that the integration layer does is converts objects backwards and forwards and passes them between layers, why not make the back-end objects visible to the front-end? After all, less code = less bugs.

One of the revolutionary feature offered by the Spring Framework is Dependency Injection. Seam takes it one step further and makes Dependency Bi-jection (INjection as well as OUTjection) possible.

So once your user has logged in, you can OUTject the user object into the session and it will remain available throughout that session. And if another java class needs it, simply INject it there and the SEAM container will take care of it for you.

How many of us have had to create a wizard-like form? Nearly all of us at some point or other.

And where did we put the data to be passed between the pages? Yes, in the Session scope. And what if the user remained on the same site but navigated away to another part? You guessed it. Memory leak.

Seam has introduced a new context called Conversation that takes care of this for us. Objects within conversation scope get garbage collected after that conversation either ends or times out. The user may still happily browse other parts of our website.

JSR-299, which has been unanimously voted on positively, is a request to include Web Beans into Java EE. Web Beans is, simply put, Seam Core. So Seam is slated to become part of the Java EE Standard.

Some Misconceptions

- JBoss Seam applications can run only on JBoss Application Server – false. They can run on WebSphere, WebLogic and Tomcat. However, for Tomcat, we need an embedded JBoss Server Instance.
- JBoss Seam applications can use only RichFaces or ICEFaces JSF libraries as their front-end – false. They can use Wicket, GWT or Flex as their front-end as well.

01 October 2008

www.ebizss.com

11

Just because Seam is often called JBoss Seam doesn't mean that it runs only on JBoss.

There is documentation available on SeamFramework.org as to how to use Seam on various other Java Application Servers.

Seam does work with Tomcat as well. However, as it makes use of EJBs, it needs an EJB container. So for using Seam on Tomcat, the best option is to use it with an embedded JBoss Application Server.

Also, the Seam development team is working hard to ensure that it supports a growing number of front-ends.

Generated Scaffolding Code

- Is generated using Hibernate Tools and Freemarker Templates
- Is JavaEE compliant
- Supports Login Security out-of-the-box
- Can use RichFaces or ICEFaces front-end
- Built-in AJAX support
- Does validation using Hibernate Annotations
- Uses JPA back-end
- Runs on JBoss AS out-of-the-box
- Can be edited using Eclipse or NetBeans

01 October 2008

www.ebizss.com

12

The code generated by seam-gen, Seam's scaffolding code generation tool, is JavaEE compliant.

seam-gen makes use of Hibernate tools and Freemarker templates to generate the front-end JSF code as well as Java classes required by the application.

The front-end pages are Facelets compliant XHTML pages.

The java classes implement Home design pattern which is much cleaner than the ActiveRecord design pattern as it separates out the object and operations.

This code makes use of Hibernate-driven JPA for its back-end. If you are a diehard TopLink-driven JPA fan, you can simply replace the hibernate jars with toplink jars.

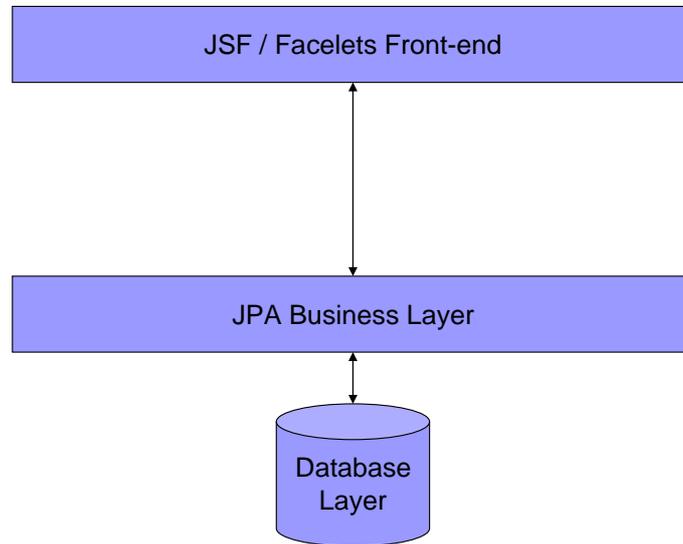
After all, the role of JPA is to provide a layer of abstraction over Hibernate and TopLink.

One note for those new to JPA. JPA makes use of annotations instead of XML files for representing database metadata.

And Seam uses these annotations to do validations.

For example, if a column as nullable=false set, then Seam will ensure that this column has an input value before passing it to the database.

The Model 2 Architecture – JBoss Seam



01 October 2008

www.ebizss.com

13

Architecture of a Java Web Application created using Seam is much simpler.



Look Ma – No Integration Layer!!!

- Seam exposes Business Layer objects and their attributes on the front-end using EL expressions.
- Hence, there is no need to have a separate set of objects in the front-end layer or to convert them back and forth in the Integration Layer

Since Seam can expose back-end objects on the front-end using EL (JSF expression language), there is no need for an integration layer to convert objects backwards and forwards anymore.



Demo

- This demo will demonstrate how quickly and easily we can generate a java web application using Seam and its tools from a database.

Steps – pre-requisites

- Install JDK 1.5.x (and not JDK 1.6.x)
 - http://java.sun.com/javase/downloads/index_jdk5.jsp
- Install ANT 1.7.x
 - <http://ant.apache.org/bindownload.cgi>
- Install JBoss 4.2.3.GA
 - <http://www.jboss.org/jbossas/downloads/>
- Install any JDBC compliant database (I am using PostgreSQL 8.1.x)
- Install JBoss Seam 2.0.2.SP1
 - <http://www.seamframework.org/Download>
- Copy the JDBC driver for your database to SEAM_INSTALL_DIR/lib

Steps (continued...)

- The following steps are for Windows
 - Go to command-prompt, change directory to SEAM_INSTALL_DIR and run the following commands:
 - seam setup (only once per project)
 - seam create-project (again, only once per project)
 - seam generate
 - seam explode (or seam deploy if you want to deploy as .ear file)
 - Start JBoss Server with default configuration
 - Visit http://localhost:8080/project_name/

Questions

- What do I need to know before I can start using JBoss Seam?
 - JSF
 - JPA
 - EJB3 (optional)
 - Hibernate (optional)
- Given the learning curve, what value have you seen from using JBoss Seam?
 - Simple CRUD in an hour (with every box ticked)
 - One-many or other kind of CRUD in a day (again, with every box ticked)
 - Extendable using Hibernate Search (Its like adding Google Search to your web application – has your boss ever asked for it?)
- How can we change the look-and-feel of a seam-gen generated application to something more presentable?
 - You WILL need a web designer with good CSS skills for this
 - Use layout/template.xhtml as a starting point and create your own template
 - Either overwrite layout/template.xhtml with your template or use your template as the template in the ui:composition tag in the generated.xhtml pages.

Where to go from here

- Me – ashish@ebizss.com
- My company's website – www.ebizss.com
- My website – www.ashishkulkarni.com
- My company's wiki – www.ebizss.com/wiki
- My company's public SVN repository – www.ebizss.com/svn/public
- www.seamframework.org
- JSR-299 - <http://jcp.org/en/jsr/detail?id=299>
- Books:
 - Manning – Seam in Action by Don Allen

01 October 2008

www.ebizss.com

19

I would attempt to answer questions as long as it is practical for me to do so.

I would also welcome your feedback on this presentation.

In case you or your company is interested to hire my services and need a more detailed CV, please feel free to contact me for that as well ☺.



Bonus

- I have created a utility called sg2me to migrate seam-gen generated applications in an exploded state to MyEclipse. This utility is available from my company's public SVN site at:

- <http://www.ebizss.com/svn/public/seam-gen2mews/>

As you can guess, I have been using MyEclipse. The project created by seam-gen is not in a structure that everyone likes. Hence, I created this utility to migrate a seam-gen generated project to a project that MyEclipse can understand.



A bit about me

- I am a Java Contractor currently contracting with JP Morgan in Glasgow. Apart from contracting, I am actively working with other developers at my company to develop bespoke software for private clients.
- I have been using JBoss Seam for just about a year now and am absolutely loving it.
- Having worked on FoxPro, Oracle, Visual C++ and SAP in the past, I have 14 years of development experience.
- I have been using Java since 2000 which to this day remains my favourite computer language.
- I have worked in a variety of industries such as Retail Banking, Investment Banking, Financial Institutions, Pharmaceuticals, Manufacturing, Automobiles and Telecommunications.