Migrating from Struts 1.x to Spring MVC

Hundekar MohammedJunaid¹, Chavan Santosh², Gondikar Shreyas ³, Shah Vaibhav⁴ and Dr. Prof. Prasanna Joeg⁵

¹,²,³,⁴ Pune University, MIT College of Engineering, MIT Campus, Rambaug Colony, Pune-38, India
³Professor, MIT College of Engineering, MIT Campus, Rambaug Colony, Pune-38, India

Abstract: In the last couple of decades web development is becoming more easier and simpler due to introduction of powerful web application development frameworks. Most of the web application development frameworks offers wide variety of services and features for developers to develop the applications with ease and simplicity. Popular web frameworks include ASP.NET, J2EE, Struts, Spring MVC etc. Every framework has some pros and cons. Developers often migrate from one framework to other according to their requirements for application development. In this paper we included our analysis about the two currently most popular frameworks namely Struts and Spring MVC. As Spring is becoming the choice of developers, we presents several approaches to migrate from Struts applications to Spring MVC. It will be useful to understand the logical mapping between the two frameworks and how to transform Struts applications into Spring MVC applications. The technology direction going forward is to use spring MVC as the basis for the client layer of application.

This paper is divided into four sections. First section gives the overview of Struts and Spring frameworks, merits and demerits of these frameworks. Second section briefly concentrates on why it has become essential to migrate from Struts to Spring and the benefits that this migration will offer. Different approaches can be followed to do such migration in framework. In Third Section we discuss about the approaches that can be used for migration in framework. Manually converting any application from one framework to other is always tedious and time consuming task. Manual conversion requires the developer to be expert at both the frameworks, understand the logical mapping between the components of the two different framework, identify project flow and working by going through code. Developers have to invest lot of effort in manual process. To ease the process of migration a Struts application to Spring MVC, we implemented a solution, a tool “Framework migrator” which will automate the task of migration. Our tool “Framework Migrator” yields better results in terms of time and cost investment in the process of automation. In final section we focus on framework migrator architecture, analysis about its efficiency and percentage completeness of automation. To fully appreciate the subjects discussed, readers should have a working knowledge of the Struts framework.

Keywords: Struts 1.x, Spring MVC, Framework Migrator, Automation in Migration

1. INTRODUCTION

Struts: - Apache Struts is an open-source web application framework for developing Java EE web. It uses and extends the Java Servlet API to encourage developers to adopt a Model–View–Controller (MVC) architecture. Model-View-Controller Architecture:

![Figure 1-Model-View-Controller Architecture](image)

Model–View–Controller (MVC) is an architecture that separates the representation of information from the user's interaction with it. The model consists of application data and business rules, and the controller mediates input, converting it to commands for the model or view. A view can be any output representation of data, such as a chart or a diagram. Multiple views of the same data are possible, such as a pie chart for management and a tabular view for accountants. The central idea behind MVC is code reusability and separation of concern [8] [9]. Struts component includes ActionForms, Tag libraries, Resource management and Configuration. Struts also provide developers Front controller servlet, Tag libraries for logic and Form binding [10], [13].

However Struts 1.x has some limitations such as:
1. Struts has very rigid inheritance structure, so it limits the reusability of code. This limits the usefulness of struts in complex systems where reusability of code is must.
2. Struts 1.x form the beans and duplicate the existing objects, hence memory requirement is heavy and it introduces redundancy in objects.
3. Struts does not support built in error handling, so errors may occur at inconvenient time. Programmer or Developer has to deal with the tedious task of error
handling and validation.

4. Testing is the difficult process when working with struts as compared to other alternative technologies available.

5. Struts has confusing naming scheme which makes developers job more complex if developer are working in teams and in distributed environment.[11].  
(Note- Some of these limitations are resolved in struts 2 but still application needs to be migrated from struts 1.0 to struts 2.0)

Spring MVC:-

Spring Framework is an open source application framework and Inversion of Control container for the Java platform. The core features of the Spring Framework can be used by any Java application, but there are extensions for building web applications on top of the Java EE platform. Although the Spring Framework does not impose any specific programming model, it has become popular in the Java community as an alternative to, replacement for, or even addition to the Enterprise JavaBean (EJB) model.[3][12].

Benefits of Spring MVC over Struts:-

1. Spring MVC provides business objects instead of form beans.
2. Spring MVC supports built in validation and error handling very well, thereby reduces the job of the developer.
3. Flexible mapping to controllers (not just to URL patterns).
4. It allows you to build application on top of spring letting you use one framework/configuration scheme in all tiers.
5. It is easy to use Aspect Oriented Programming (AOP) and other spring features in web tier.
6. It is compatible with newer technologies and tools [12].

2. WHY TO MIGRATE

Experts in the web frameworks field suggest that the Struts framework has reached the point where its innovation and adoption will start declining. In fact, the Craig McClanahan, chief architect and author of Struts, has left the project for good, and he is actively urging sticking with struts implementation classes and struts tag libraries, struts tiles (used primarily for static content). By learning the existing Struts application and Spring MVC, identify the minimal set of changes required to successfully migrate the functionalities[15].

Both frameworks have their merits and drawbacks. Many development teams have learned to rely on Struts as the foundation for building quality software under strict deadlines. With so much momentum behind Struts, even development teams that would like to integrate features of the Spring framework don’t want to switch to Spring MVC. The Spring architecture allows you to connect Struts as your Web framework to Spring-based business and persistence layers [15].

One of the major advantages of combining Struts and Spring, and doing it by delegating Struts actions to controller. This gives application designers more flexibility in the design of components. [4]

2. Flexible:- At the framework component level, Struts requires use of Struts-specific objects, such as Form Beans (static or dynamic), Actions, Action Mappings, Action Forwards, and Request Processors. Spring MVC is more flexible, as all its major components are defined as interfaces.

3. Struts Is a Web Framework Only: - Struts addresses only the presentation aspects of application development. On the other hand, Spring MVC is an integral part of the Spring framework, which fully integrates Spring with the rest of the frameworks that manage business components as well as other aspects of Spring enterprise development.

4. No action Forms: - One of the biggest and the most positive differences in the Spring framework is that it has no specialized ActionForm objects. The framework supports binding of HTTP form values directly into POJOs (Plain Old Java Objects). This feature greatly simplifies application maintenance by limiting the number of classes to create and maintain.

5. Future scope: - Spring MVC is compatible with more tools and software and it has bright future, whereas popularity of struts has been steadily declining, while on the other hand Spring MVC is steadily attracting the attention of Java developers. Many large organizations (like IBM), which were working on Struts earlier are migrating to Spring MVC.[12]

3. APPROACHES

There are two approaches for migrating struts applications to Spring MVC.

1. Integrating struts with spring
2. Complete migration

1. Integrating struts with springs:-

This approach focuses on how our current Struts assets can be reused in Spring MVC configuration. Areas to consider are struts implementation classes and configurations, struts tag libraries, struts tiles (used primarily for static content). By learning the existing Struts application and Spring MVC, identify the minimal set of changes required to successfully migrate the functionalities[15].

Both frameworks have their merits and drawbacks. Many development teams have learned to rely on Struts as the foundation for building quality software under strict deadlines. With so much momentum behind Struts, even development teams that would like to integrate features of the Spring framework don’t want to switch to Spring MVC. The Spring architecture allows you to connect Struts as your Web framework to Spring-based business and persistence layers [15].

One of the major advantages of combining Struts and Spring, and doing it by delegating Struts actions to
the Spring framework, is that you can apply Spring’s AOP interceptors to your Struts actions. By applying Spring interceptors to Struts actions, you can tackle cross-cutting concerns with minimal effort. One major disadvantage of this approach is as both frameworks are used, cost and efforts increases.

2. Complete Migration:
Complete migration means total replacement of all the components of the Struts framework with Spring MVC. At the end of the process, no Struts-specific components would remain in the application.

A simple struts application can be divided into six different categories of files.
- JSP (ex. Login.jsp)
- Action classes (ex. LoginAction.java)
- FormBeans (ex. LoginForm.java)
- TagLibs (ex. Struts-html.tld)
- Config Files (ex. StrutsConfig.xml)
- Library files (ex. Struts.jar)

To migrate from Struts to Spring we have to change the source code of above mentioned files so that they can work on spring MVC.

Following Flowchart shows the algorithm that we follow for the migration.

![Flowchart for migration process](image)

**Step 2: Changing Struts Form Beans to Spring Command Classes**
Struts provide FormBeans as interface between the JSP and the controller servlet. In Spring these are replaced by POJO (Plain Old Java Objects) Command Classes. Form beans extend the ActionForm class of struts framework. POJO (Plain Old Java Object) Command Classes do not extend any class.

**Step 3: Changing Struts Action classes to Spring Controller classes**
Struts provide ActionServlet by default as the controller where as in Spring the default controller is DispatcherServlet. Spring has different controllers which handle individual modules. These can be seen as replacement for Action Classes. Annotations can be used within the class definition to make them act as controller.

Ex. public xxxClass extends Action
[...............................]

Will change to
public xxxClass
@controller
[...............................]

In Struts ActionMapping, objects are pointers into presentation resources (Actions, JSPs, Tiles, HTML files, etc.). The closest component in Spring MVC to ActionMapping is the ModelAndView interface. Spring Controllers return implementations of the ModelAndView interface, which like a Controller can be custom implemented. Or, if appropriate, you can use the ModelAndView implementation supplied by Spring MVC[3].

Ex.

```java
public ActionForward execute( ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response)
{…………………}
```

Will be written as

```java
Protected ModelAndView onSubmit( HttpServletRequest request, HttpServletResponse response, Object cmd,BindException exp) throws Exception
{…………………}
```

As the name implies, ModelAndView objects have Model and View components. Model components contain the business object to be displayed via the View component. Depending on the scenario, ModelAndView implementations may not have any Model components.
included. They may simply forward into some form of an actual View component (JSP, XSLT, Tiles, HTML, XML, etc.).

Ex. Replace return mapping.findForward("success");

With Return new ModelAndView(getSuccessView());

Step 4: Changing Struts Configuration files to Spring Configuration files
Struts use web.xml & struts-config.xml to direct the flow of the application
- In struts we have at least one struts-config.xml file (or more if we are using modules) that holds all the forwards, action mappings, form definitions, and plug-in declarations. In Spring MVC, all the Web application-related controller declarations are configured as Spring Beans.
- One or more Dispatcher Controllers dispatch all requests for the Web resources to the appropriate Controllers. For instance, if you want to remap your "do" application into a Spring MVC application, you register the following servlet mapping in the web.xml of your application.

step 5: Replacing Struts jar files with Spring jar files
- Removing the struts specific jars (struts.jar, struts-legacy.jar etc)
- Adding Spring specific jars (standard.jar, spring.jar etc) into lib folder of the application
- All *.tld remain same.
- Updating classpath entries.

Validation and Error messages:-

If Commons Validator is used in Struts, it can be completely reused in Spring. Spring 1.2 does not officially support the Commons-based validation framework, but the "sandbox" version of Spring MVC supports the reuse of validation definitions written in Commons Validator markup (validator.xml and validation-rules.xml). In any case, do not throw away your XML files with validation declarations. They could be reusable in Spring. Spring recognizes Struts message bundles in an identical format. In order to reuse your existing Message resources within Spring MVC, you just configure it as message Source in the Spring MVC configuration file.

4. AUTOMATION

Why Automation:-

Converting any application from one framework to other is a tedious task. Developer who has to convert the application, need to study the existing application, understand the working, flow, characteristics, expected outcome of the application by going through the code of application line by line. After analyzing the application he has to map every component of that application to its equivalent component in the other framework. This process needs the developer to be the expert in both the frameworks. Developer has to invest lots of efforts and time to do this. Even after completing the migration, this conversion process cannot be reused in other similar projects, developer has to follow the same series of action for next set of similar projects.

The solution to this is building a tool that will automate the task of migration from one framework to other, in our case Struts to Spring. The approach discussed for the migration of Struts application to Spring, in above section can be automated. We present an architecture for a tool "Framework Migrator". Existing Struts application will be provided as input to the tool. Depending on the application content, the tool will generate equivalent Spring application using the algorithm mentioned in last section.

Benefits of Framework Migrator:-

1. As we are building the tool for conversion, conversion process will be automated. Tedious task of manually converting the Struts application to Spring application will be eliminated.
2. As the tool will be used for migration, efforts needed to migrate will be reduced; hence cost of conversion will also be reduced.

There are five major components in the architecture of Framework Migrator.
1. File Locator:- It Checks the input folders structure and file types. It also Stores input and output project folder path in mapping database.
2. File Converter:- Converter Checks the file types and calls appropriate file parsers.
3. File Parsers:- Individual parsers are designed for individual file types like XML parser, JSP parser, JAVA parser. These Parsers will convert Struts specific code, tags to Spring equivalent one.
4. File Creator:- File Creator creates Spring project Folder and copies the files to appropriate directory.
5. Mapping database:- Database will store the mapping of the tags required for the Struts-Spring migration.
Analysis about Framework Migrator:-
Cost-effort reduction analysis: - Let us take an example, Suppose for migration of a Struts application having average complexity level, and size of 10,000 LOC (lines of code) or 200 function points, a team of 4 person needs 10 days, then same Struts application can be migrated to spring in few hours by using the tool. (‘Few hours’ because even after automation some of the components must be set up manually such as configuration files, plug-ins).

Framework migrator cannot achieve 100% migration in all cases. Success of automated migration of application in percentage depends on complexity level of application, software tools we are using to run the application, their version differences and other execution environmental factors. After the changes done by the framework migrator tool, some of the configuration files may need to be changed, libraries may need to be added as framework differences exist. Library files need to be imported as some of the libraries that are inbuilt in Struts may be missing in Spring. Overall we can say framework migrator achieves the automated migration of application about 85-90 percent, remaining amount of work we need to do manually.

5. CONCLUSION
Web application development has become simpler due to the web application development frameworks. Many web frameworks are being designed and released for the use of developers to develop application with ease. Developer often needs to migrate from one framework to other. In this paper we highlighted the differences between the two frameworks. Spring MVC Framework offers more advantages over Struts framework. Most of the industries which worked on Struts earlier are now undergoing the process of migrating to Spring framework; as they believe that the Struts framework has reached the point where its innovation and adoption has started to decline. Whereas the highly popular Spring framework (with Spring MVC as its core component) is nicely designed, productive and innovative. We presented the different approaches which can be followed for migrating existing Struts applications to Spring MVC.

Finally we discussed a solution “Framework Migrator” that we have implemented, which simplifies the process of migration and minimizes the efforts needed for migration. As all the web developers are highly likely to migrate from Struts to Spring , our solution “framework migrator” will be useful for them to convert their existing applications to spring MVC with relative ease and less time.

REFERENCES