

UDC 004.4'23

V.V. Gerasimov, I.V. Ponomarev, O.O. Lishchenko

ANALYSIS OF MAIN TECHNOLOGIES FOR DEVELOPMENT OF WEB APPLICATIONS ON JAVA PLATFORM

Annotation. A comparative analysis of modern technologies for development of web applications on Java platform, which are JavaServer Pages, JavaServer Faces, JavaFX, Apache Struts and Spring Framework, is conducted. Features of these technologies and conclusions about their possible areas of usage are provided.

Keywords: JavaServer Pages, JavaServer Faces, Apache Struts, Spring Framework, Java, MVC.

Foreword. The Internet keeps growing, and along with it the demand for new web sites and Internet systems is growing. The amount of web programmers and competition between them is growing. This leads to reduction of prices of their services. Customers' requirements for these services are also rising.

Since web applications don't require an installation and can be easily customized, their popularity is rising. They have lesser requirements for client hardware and are easier to control. Many desktop applications already use web interface to interact with user. Overall, web applications already achieved the functionality that was available for desktop applications only before.

Nowadays, customers demand more and more functional and complex web applications. To resolve these tasks, technologies with high code reuse index have to be used. Java language and its key features like scalability, reliability, versatility are very helpful. It can easily be transported to different operation systems and has syntax easily understood by most C/C++ and C# programmers.

The aim is to review modern technologies for development of web applications, such as JavaServer Pages, JavaServer Faces, JavaFX, Apache Struts, Spring Framework and conduct a comparative analysis of these technologies with chosen features.

The main part. First, the key features of technologies mentioned before were reviewed. The first technology, JavaServer Pages (JSP), is the result of development of Java Servlet technology. It allows web developers and designers to quickly develop and easily maintain dynamic web pages that manage business systems. Since JSP is a

part of Java technology, it allows for quick development of platform independent web applications. JSP technology separates user interface from content generation, because of which developers can change the page content without changing the dynamic content.

JSP specification is a product of cooperative work of business applications and development tools industry leaders led by Sun Microsystems. To provide the support of JSP specification in every web application or server, Sun made JSP specification free for developers.

Developers and designers that have experience in working with HTML can:

- Use JSP technology without any need to learn Java language;
- Expand JSP language;
- Easily develop and maintain pages.

The second technology is JavaServer Faces (JSF). It's a framework and technology for Java language web applications. It makes the user interface development for Java EE applications easier.

JavaServer Faces technology includes:

- A set of APIs for representation of user interface (UI) components and their state management, event handling and input data validation, navigation, internationalisation support (i18n) and accessibility.
- A special JSP tag library JSF interface realisation on a JSP page. In JSF 2.0 the Facelets technology instead of JSP is used as view handler.

Unlike most MVC frameworks that use requests, the JSF method is based on using components. The user interface components state is saved when a user requests a new page, and then restored if the request is repeated. JSP is usually used for data presentation, but JSF can also be used with other technologies, for example XUL.

The third technology is JavaFX. It's a platform for RIA (Rich Internet application) applications development that allows for creation of unified applications with rich graphical user interface for direct launch from operation systems, work in browsers and mobile devices, including the ones that work with media data.

JavaFX technology was first demonstrated on JavaOne conferention in May 2007 by Sun Microsystems corporation. On December 4th 2008 version 1.0 of the platform was released.

JavaFX is a powerful platform aimed for Java interface, capable of processing large scale data-driven business applications. It provides a large set of UI controls,

graph and media APIs, with high performance graphics and media hardware acceleration to make the development of interactive applications easier.

The next technology is Apache Struts. It's a free opensource framework for development of web applications on Java platform. The framework was created by Craig McClanahan and handed over to Apache Foundation in May 2000. First, it was a part of Apache Jakarta Project and known as Jakarta Struts. Since 2005 it's a high level Apache project.

The framework provides three key features:

- Requests handler provided by application developer;
- Response handler that hands over the control to another resource that completes the request;
- Tag library that helps developers to create interactive applications using JSP.

Struts was created to distinctly separate model (business logic), view (HTML pages) and controller responsible for transmitting data from model to view and back. Struts provides a standard controller-servlet named ActionServlet and different view pages creation tools. Web application developer is responsible for writing model code and struts-config.xml configuration file creation that binds model, view and controller together.

The last technology is Spring Framework. This platform is widely used for opensource applications development and designed to make Java EE development easier. The first version was written by Rod Johnson, who released the framework along with his "Expert One-on-One J2EE Design and Development" book publication in October 2002. The structure was first released under the Apache License 2.0 in June 2003. The first major release 1.0 was released in March 2004, with the next major releases in September 2004 and March 2005.

Spring Framework consists of a container, an elements management platform and a set of services for user web interfaces development, transactions support and state storage realisation. Spring also includes Spring Web MVC — an extensible MVC platform for web applications development.

The central part of Spring Framework is the Inversion of Control container, which provides Java objects view configuration and management tools. The container is responsible for the object's lifecycle management: objects creation, calling of objects' initialization and configuration methods by binding them together.

Then next are the results of comparative analysis of the technologies chosen. For each characteristic a number of points («+») from 1 to 5 was set. The technologies are compared using the next characteristics:

- Development simplicity — describes how easy the technology is in learning and further development;
- Functionality — shows how “powerful” an application developed with the technology can be;
- UI richness – describes how rich is the interface provided by the technology;
- Server-Client – this characteristic describes how well the technology can be used to develop both server and client parts of an application;
- Screen resolution – shows how independent the technology is from screen resolution;
- Separation between view and logic — describes how interface is separated from the functional part of an application.

We should also note that all the technologies chosen and their development tools are free. Also, all the technologies chosen are supported by all modern browsers with JavaVM virtual machine installed.

Table 1

Comparison of technologies

Characteristics	JSP	JSF	JavaFX	Struts	Spring
Development simplicity	+++++	+++	++++	++++	+++
	A JSP contains simple tags similar to HTML. JSF, as well as Struts, is simple in development, but takes a lot of time to learn. JavaFX has rich interface, which makes development more comfortable.				
Functionality	++	+++	++++	+++	+++++
	JSP is mostly used as interface, leaving logic to other components. JSF is mostly aimed to (but not limited by) creation of the server part of an application. JavaFX, unlike Struts and Spring, has rich graphics functionality, including 3D.				
UI richness	+	+++	+++++	+++	+++
	Among the mentioned technologies, JSP has the most primitive interface. JSF, Spring and Struts provide more abilities in this, while JavaFX provides the best potential of interface development thanks to CSS support.				
Server-Client	++	+++	+++	++++	+++++
	JSP is mostly aimed to work as the client. JSF is more designed for the server part, while JavaFX – for the client part. Struts and				

	Spring are good for both server and client parts of an application.				
Screen resolution	+++	+++	+++++	+++	+++
	JSP, JSF, Spring and Struts only support raster graphics, while JavaFX supports both vector and raster graphics.				
Separation between view and logic	++	++++	+++++	++++	+++++
	JSP, as well as Struts, uses the MVC (model-view-controller) pattern. JSP pages serve as view, while servlets are responsible for logic. JSF also uses MVC, but it uses facelets as view. JavaFX uses cascade style sheets (CSS) to separate view from logic. Spring uses containers for separation and even has the subsidiary Spring MVC technology designed for that purpose.				

So, the results are: JSP — 15, JSF – 19, JavaFX — 26, Struts — 21, Spring – 24.

Conclusion. It can be seen that it’s pretty hard to choose a certain leader from the reviewed technologies. Each technology has it’s own area of usage. For instance, if you need a powerful server part — it would be better to choose JSF or Spring. You should also keep in mind that Spring also provides an ability to use the Inversion of Control principle, which allows for changing the structure of an application while it’s running. If the application is supposed to work with graphics and 3D, or if the interface is supposed to be described using CSS, then in this case JavaFX is the best option. Or, if you need a technology for development of a simple application — JSP should do. It’s the simplest of the technologies mentioned, but you need to note that application developed with using JSP will have an overall simple functionality and primitive interface. Struts technology also is simple in development, but it allows for better interface and functionality, which is why it will take longer to learn.

REFERENCES

1. JavaServer Pages Overview — Access mode: <http://www.oracle.com/technetwork/java/overview-138580.html>
2. Бобаченко М. Введение в JavaServer Faces — Access mode: http://www.javaportal.ru/java/articles/java_Server_Faces.html
3. Pawlan M. What is JavaFX? JavaFX 2 Documentation — Access mode: <http://docs.oracle.com/javafx/2/overview/jfxpub-overview.htm>
4. From a birds eye. The Apache Struts Web Framework — Access mode: <http://struts.apache.org/birdseye.html>
5. Spring Framework — Access mode: <http://projects.spring.io/spring-framework/>