Research on J2EE Teaching Based on Mainstream Open Source Frameworks

Xiaoliang Xu, Weihua Hu
College of Computer Science
Hangzhou Dianzi University
Hangzhou, Zhejiang, 310018, P.R.China
xxl@hdu.edu.cn, hzhuhm@yahoo.com.cn

Abstract—Students are difficult to get an effective way to master essential knowledge of J2EE course because they are lack programming experiments and teaching content is too much. In this paper, a layered, loosely coupled and lightweight J2EE application architecture is proposed and introduced based on the layered architecture theory and open source framework of Spring, Struts and Hibernate. On the basis of the architecture, the course construction and teaching innovation is put forward, and the application design and implementation is also given by example at the lesson. The practice shows that the architecture provides high reusability, expansibility and efficiency. Thus it helps students to understand J2EE knowledge, to obtain effective learning methods, and fleetly to guide themselves to carry on practical programming.

Keywords—architecture; open-source; J2EE; mainstream framework, teaching

I. INTRODUCTION

The J2EE platform is a multi-tiers, distributed application framework that provides system level services to facilitate application development. However, classical Java EE architecture suffers poor performance for business applications due to its heavyweight EJB beans. Many in the open source community, especially smaller vendors, have chosen the alternative of developing frameworks designed to simplify the experience of building J2EE applications. Popular lightweight frameworks such as Struts, Hibernate, and the Spring Framework play an important role in many of today’s J2EE development projects.

A. The advantage of the open source frameworks

Using a well-designed open source framework offers many advantages [1]:

• With a good framework, developers write only the code they need to write; they don’t get bogged down working directly with low level infrastructure APIs. This is the key value proposition.
• A well-designed framework can provide structure and consistency to an application. The structure will be clear to additional developers joining the project.
• An easy-to-follow framework can promote best practice through examples and documentation.
• Successful open source frameworks are better tested than in-house code.
• Frameworks usually become popular only if they have something to offer. In-house frameworks are often mandated, while a J2EE project is likely to adopt an open source framework only if it delivers clear benefits.

• In addition, many high-quality frameworks are now available that offer outstanding documentation and the support of a focused development team, without imposing licensing fees.

B. The “miscellaneous” of the open source frameworks

Growing recognition and use of J2EE development frameworks is measurably reducing cost in many projects, as well as delivering better speed to market and higher maintainability. But the diversity of needs and design ideas will inevitably lead to “miscellaneous” of framework products. For example, presentation tier frameworks such as struts, webwork, jsf, and echo compose a competing flora. It’s easy to swap out JSP with alternative view-tier languages such as Velocity, FreeMarker, etc. There are also a great many of Ajax frameworks in the front rank of area with names Prototype, Dojo, JQuery, Extjs. And some projects still built their own persistence frameworks, the existence of Hibernate, IBATIS, JPA and JDO implementations made renown “miscellaneous” indefensible. On the one hand, full range of frameworks at different tiers is propitious to increase J2EE platform’s maturity by building architectures with appropriate frameworks according to actual demand of project. On the other hand, it’s adverse for students to learn and comprehend essential of J2EE knowledge, because beginners don’t have the ability of distinguishing common ground and differentia between frameworks.

II. J2EE TEACHING OVERVIEW

With J2EE was regarded the most popular platform for commercial software development, many schools expect their graduates could be able to work competently in companies relative to J2EE development. As a result, various Java courses about J2EE Programming are set in all kinds of colleges and universities. For the first time in 2003, Hangzhou Dianzi University (HDU) enrolled "Java application development" major direction. As a key course in this direction, "J2EE Programming" was offered to the junior and graduate students in 2005 and web components technologies are emphasized including JSP, Servlet, JDBC, XML and software frameworks. During our practical teaching on J2EE programming course, several kinds of problems haven’t been resolved well all the time.

A. The knowledge system of J2EE is so big

J2EE uses a multi-tiers distributed application model. There are three tiers in the typical J2EE application model: web presentation tier, business logic tier and data tier[2].
• The web presentation tier provides complex services for the requests from a browser or an application
programming client. There are some components in this tier, such as JSP, Servlet and so forth.

- The business logic tier is the core part in J2EE model. It includes the application programming servers and the EJB components processing business rules on them.
- The data tier is composed of database systems, and some logic modules dealing with data access.

As introduced above, J2EE includes so many technologies that the learning curve for fresh men may be somewhat steep if they are not kept in good guide and instructions.

B. Not enough practical programming experience

J2EE Programming is an application-oriented project-driven course. Practical teaching for programming courses has played an important role in domestic university. Although more researches have paid more attention to programming practical teaching and have obtained some good achievements, focusing on student subject and making their effects on teaching reform haven’t been paid more attentions in domestic university [3-5]. Although students try to master the programming methods, when they meet some difficulties which they can’t overcome easily, they will lose their confidence and give up study.

C. Lack of good J2EE study habits

It is very difficult for single student to finish all tasks. Hence, cooperation ability should be emphasized in software development. But a great many students have little this ability or have no at all. If student do program only by himself without any communication with others, his programming achievement is very limited. Documentation and expression ability is also very important to students. Though sufficient detailed documentation which are available on the Internet or books, a great number of students have no idea of searching information or taking advantage of them to solve problems what they encounter.

Aiming at figuring out these problems, helping students master the essential of J2EE knowledge system, cultivate a good study habit and quickly show them the way to practical programming, this course analyzes drawbacks of traditional Java EE architecture and points out advantages of adopting lightweight Java EE architecture, and design a lightweight J2EE application architecture based on open source framework of Spring, Hibernate and Struts, as with the advantage of development framework of loosely coupled, the operating system and database independence, low cost, easy to maintain and upgrade.

III. TECHNOLOGIES IN CASE ARCHITECTURE

Through studying the concept and structure of MVC pattern, three-tier J2EE architecture, IoC(Inversion of Control) and Aspect-oriented programming, on the basis of popular SSH (Struts, Spring, Hibernate) Open Source Frameworks, a case architecture for teaching in course is planned out. Then, base on the case architecture, we make some useful attempts at teaching methods, curriculum design, etc.

A. MVC pattern

Model-view-controller (MVC) is an architectural pattern used in software engineering. In MVC, Model is the main business entity of the application, it stand for Business Data and Business Logic; View is with responsibility for showing web pages and interacting with the User; Control receive user’s input, then invoke Model and View to perform the user’s request. The View sends updates to the Controller, the Controller updates the Model, and the View gets updated directly from the Model [6]. The MVC pattern works as shown in Figure 1.

![The structure of MVC pattern](image)

The MVC design pattern provides a host of design benefits. MVC separates design concerns (data persistence and behavior, presentation, and control), decreasing code duplication, centralizing control, and making the application more easily modifiable. MVC also helps developers with different skill sets to focus on their core skills and collaborate through clearly defined interfaces. For example, a J2EE application project may include developers of custom tags, views, application logic, database functionality, and networking. An MVC design can centralize control of such application facilities as security, logging, and screen flow. New data sources are easy to add to an MVC application by creating code that adapts the new data source to the view API. Similarly, new client types are easy to add by adapting the new client type to operate as an MVC view. MVC clearly defines the responsibilities of participating classes, making bugs easier to track down and eliminate.

B. SSH Frameworks

In order to mask the technical complexity of J2EE systems development, and satisfy the requirements of reuse for the system design and development framework, SSH (Struts, Spring, Hibernate) Open Source framework are commonly used.

1) Struts is a famous MVC based web application development framework. Struts integrates Servlet, JSP, Tag library and application resources seamlessly and supports the MVC model via Struts Action, Struts Form Beans, Struts tag libraries, and Struts Controller to smooth the application logic. The structure works as shown in Figure 2. With Struts-based, enabling developers to concentrate its attention on building a business application, without having to concern the question of architecture.

2) Spring is an excellent POJO container that embodies many brilliant ideas including IoC and AOP. It is a layered architecture consisting of seven well-defined modules [7]. Each module can exist individually or unite each other: The Spring modules are built on top of the core container, which defines how beans are created, configured, and
managed. The main component of the core container is BeanFactory, which is an implementation of the Factory pattern. BeanFactory applies the Inversion of Control pattern to separate an application's configuration and dependency specification from the actual application code. Since its release, Spring has become popular as an alternative, replacement, or even addition to the EJB model.

3) Hibernate is the most successful O/R mapping middleware that handles mappings between domain objects and database relations, bridging the gap between Java EE applications and DBMS. Hibernate solves the Object-Relational impedance mismatch problems by replacing direct persistence-related database accesses with high-level object handling functions [8]. It completely comes close to achieving transparent persistence. Transparent persistence frees domain objects from the responsibility of managing their persistent representation, enabling them to contain business logic where appropriate, without mixing that with persistence operations. It can also greatly increase productivity by eliminating the need to write verbose and often error-prone persistence code.

C. Integrate case Architecture

An integrated case architecture is designed to support a platform for quickly guiding students into practical programming, then help them learn more related knowledge about the framework. According to the above introduction of Struts, Spring and Hibernate, during the integrated architecture, the presentation tier is Struts. Business tier is Spring, and persistence tier is Hibernate. The case architecture can be showed as Figure 3:

The case architecture regards Spring as the core and integrates Struts2 presentation tier components and Hibernate persistence tier components. Plain Old Java Objects (POJO’s) run as data carriers inside the lightweight container through every tier. It achieves strictly division between presentation tier, business tier, and persistence tier. It achieves loose degree of coupling through Spring’s inversion of control. It also provides effective supports for components extendibility, components management and transaction processing.

1) Presentation tier

Struts2 is adopted as the framework of the presentation tier. Beside, Ajax Frameworks based on JavaScript, Template Engineers such as Freemarker, velocity are assistant to gain more friendly and convenient user experience. Various points of presentation technologies designed in the tier are useful to help students master the basic knowledge of J2EE area. Furthermore, it’s benefit for widening students’ eyereach.

2) Business tier

Spring serve as the backbone for the business object tier, provides a web application context concept, a powerful lightweight IoC container that seamlessly adapts to a web environment. The main component of the core container is BeanFactory, which is an implementation of the Factory pattern. BeanFactory applies the Inversion of Control (IOC) pattern to separate an application's configuration and dependency specification from the actual application code. With sufficient AOP support, it provides more powerful declarative facilities and services.

3) Persistence tier
Hibernate is introduced in persistence tier, which is most useful with object-oriented domain models and business logic in the Java-based persistence tier. It provides data query and retrieval facilities which can significantly reduce development time, or else spent much time with manual data handling in SQL and JDBC. In recent years, Hibernate undoubtedly has been the criterion in the Java ORM field. And mastering Hibernate is almost a basic requirement in recruitment of software companies. So Hibernate learning is meaningful to students in school.

In addition, the POJO make it easy to unit test outside the application server and can even incorporate JUnit to assist with testing.

IV. TEACHING BASED ON CASE ARCHITECTURE

A. Teaching methods

First, we adopt the “Build-and-Fix mode” teaching method. The way of this method doesn’t require students must be familiar with knowledge about the framework in each tier with the order of presentation, business, and persistence in advance before they begin their inchoate practical programming. The purpose is advocating means that guiding students quickly throw themselves into practical programming and learn related knowledge in the process of development by themselves. The case architecture is only a platform for them to practice on and find problems, solve problems. The most importance is that the case architecture works as vane in the J2EE learning, instructing the ken of technology hotspots.

Besides, the course especially emphasizes teamwork and communion. When students do exercises or assignment under the instruction, they always encounter problems, the solutions maybe are not mentioned or taught at all in the course and single doesn’t have the capacity of comprehending all key points. In this case, teamwork and communication with other students are obviously so important.

B. Case IMPLEMENTATION

Based on the case architecture, the case implementation has been assigned for students-Designing and developing a Project Application system. In the entire process of the system development, student is asked to work in a team and accomplish some modules independency. One of many students’ implementations which have been completed can be showed as Figure 4.

![Figure 4. The case implementation based on the case architecture](image)

The evaluation of the implementation is also a mirror reflecting advantage and disadvantage of students in mastering the J2EE knowledge.

C. Evaluation of achievement

From the early course of J2EE in 2007 up to now, there are three batches of junior and graduate students that have studied the course. The effect of the course is prominent and remarkable and the teaching method was popular and enjoyed by students. To some extent, course innovation has greatly shortened the period of students spending in mastering the mainstream open source frameworks and the essential of J2EE knowledge. Furthermore, it brought about the outstanding promotion of students’ ability in practice development and cultivates students with good software development habits. And a great many graduates who had studied in the course have been engaged in J2EE development in famous IT corporations such as Alibaba, Hundsun, Insigma, etc.

V. CONCLUSION

The J2EE course based on the case architecture which is integrated by mainstream open-source framework of Spring, Hibernate and Struts is effective. And through several years’ practical teaching, many beneficial results have already been achieved. The course satisfies the students’ demands, conform to the J2EE software development trend, and involve the advanced practical teaching method. The innovation teaching of J2EE course is just an initial exploration, which needs to be strengthened in future. From now on we will still continue to carry on the innovation and improve the efficiency in J2EE teaching.

ACKNOWLEDGMENT

This research is supported by the Special Funds for Key Program of the China No. 2009ZX01039-002-001-04, and the Science & Technology Research Program of Zhejiang Province, China No. 2008C11102.

REFERENCES


