

A Web Application Framework for Reservation Systems and its Reusability Evaluation

Feng Zhou and Takeshi Chusho[†]

Abstract— Web application is used in various business fields on the Internet and intranets. It is an efficient way to develop Web application on the base of a framework. In this paper, after the well designing of its architecture, a domain specific framework for reservation is developed based on a meeting room reservation system. Then, the framework is applied to two types of reservation systems, an online book store system and a soccer ticket reservation system, and its reusability is evaluated. With the result of 62% and 65% respectively, high reusability of the framework has been confirmed. In addition, another framework for time-based reservation, was developed, and the trade-off relationship between the range of the domain and the reusability has been confirmed. Finally, a visual tool is developed to generate the source code for the database access transaction.

Keywords: framework, reservation, reusability, trade-off

1. INTRODUCTION

With the widespread use of the Internet and intranets, the number of web applications such as BBS, shopping site, has been increasing. However, it is difficult to develop those web applications separately without any support technology. A web application framework provides an abstraction in which common code providing generic functionality can be selectively overridden or specialized by the developers. Furthermore, the overall control flow of the application is dictated by the framework. As the repetitious work is avoided, the more efficient it becomes to develop web applications based on a framework.

In our research [1][2] until now, compared to the more general frameworks like Struts[3] and Hibernate[4], we developed a framework for a more narrowed domain—reservation. It is a field that is experiencing great demand, like classroom reservation in a school, ticket reservation for a train and so on. Because the domain to be applied has been narrowed, it should be more efficient to develop such applications using this framework. The framework is applied with several types of reservation systems to confirm its reusability.

Related works include those like an Ajax framework developed and then applied with a sample system to evaluate its reusability [5]. A multi-layer framework aiming at high performance was designed and implemented for common information system based on .NET, and the performance

improvement was figured out [6].

This paper presents the architecture of the framework in Section 2, the description of reservation in Section 3, the development and reusability evaluation of the framework in Section 4 and Section 5, the trade-off relationship between the domain and the reusability in Section 6 and a visual code generation tool in Section 7.

2. ARCHITECTURE DESIGN

In a narrowed domain, because the data flow is similar from one application to another, we can design the architecture on the framework level. Therefore, the developers will be released from the architecture designing work with a well designed architecture already available from the framework.

2.1 Multi-layer architecture

Multi-layer architecture is a client-server architecture in which, the presentation, the application processing and the data management are logically separate processes.

The following are the principles we followed to design the framework's architecture.

- The lower layer provides its interfaces to the upper layer. It's not aware of upper layers.
- The upper layer doesn't need to know the detail of the lower layer. The only necessary information is the lower layer's interfaces.
- The upper layer only needs to know the interface of the layer directly beneath it.

Following these principles, we were able to make our framework more understandable, extensible, and hence, more efficient.

Our multi-layer architecture is made up from 3 layers. The presentation layer deals with the user interfaces. The business logic layer controls an application's functionality by performing detailed processing. The data access layer processes the data with the relational database.

2.2 Presentation layer

The presentation layer deals with the user interfaces. There are many existing general frameworks for this layer, among them, the most famous one is Struts.

Web applications based on Java Server Pages sometimes commingle database code, page design code, and control flow code, so they become difficult to maintain. Struts is a free open-source framework for creating web applications. It is designed to help developers create web applications that utilize a MVC architecture, which can solve the above problems well.

In our framework, we developed this layer based on Struts,

[†] Manuscript received December 8, 2008. F. Zhou and T. Chusho are with the Department of Computer Science, Meiji University, Kawasaki, Japan.

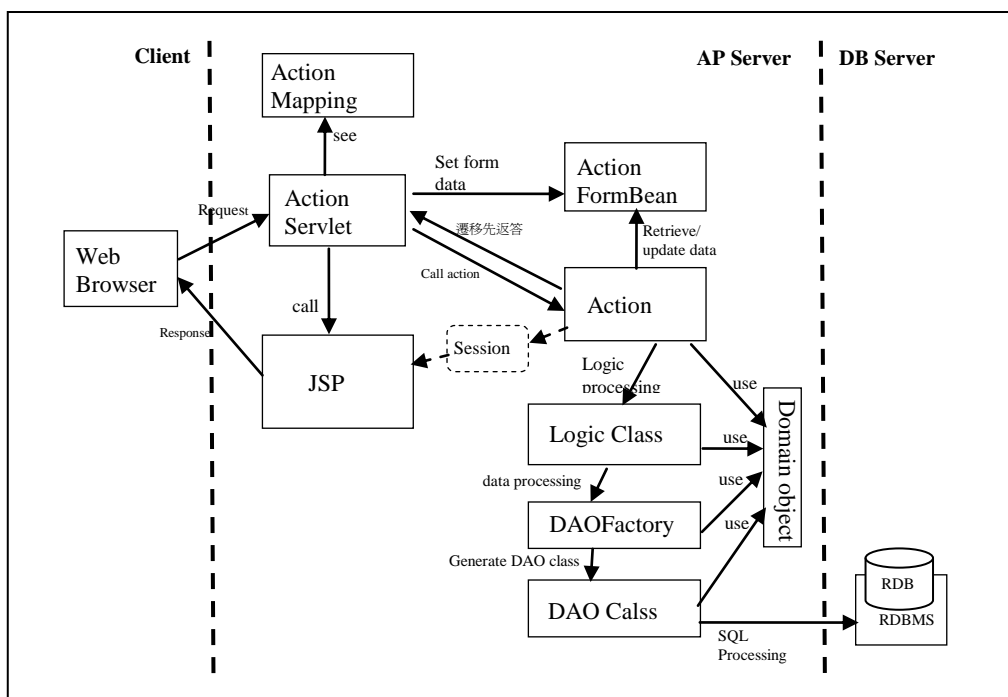


Fig.1. The architecture of our framework

using some of the libraries from it. So the MVC model is adopted in our framework.

2.3 Data access layer

The data access layer processes the data with the relational database. The common ways to handle this operation are JDBC or object-relational mapping framework like Hibernate, iBatis. In our framework, we adopted the Data Access Object (DAO) pattern [7] in this layer, and also in Section 6, a visual tool will be developed to generate the code automatically.

As one of the J2EE core patterns, DAO abstracts and encapsulates all access to the database. It manages the connection with the database to obtain and store data. DAO can be highly flexible by adopting the Abstract Factory and the Factory Method [8] patterns. As we only use relational databases, the Factory Method pattern adopted DAO is selected for our framework. In this strategy, a DAO Factory produces a variety of DAOs needed by the application, so it becomes much more convenient and easier for extension and maintenance.

2.4 Framework's architecture

With the 3 layer division, and after adopting Struts to the presentation layer and DAO pattern to the data access layer, the architecture of our framework is shown in Figure 1.

3. RESERVATION SYSTEM

3.1 The system of reservation

In this paper, reservation is defined as the claim to obtain a future use of certain resources for a period of time in advance. And compared to the conventional way by paper records, we realize this claim by Web application.

As the conceptual model being shown in Figure 2, there are end-users who follow certain conditions to make the

reservation. There are resources being divided into reservable frames.

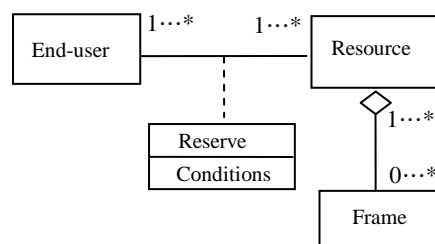


Fig.2. Conceptual model of reservation

According to the type of frame which the resources have been divided into, the system of reservation is divided into 3 categories. Space-based reservation refers to the systems that choose the resource first according to time, and then choose the frame according to space. Ticket reservation is a sample in this category. Time-based reservation refers to the systems that first choose the resource according to space and then choose the frame according to time. Meeting room reservation is a sample in this category. In addition, the systems that don't belong to these two categories, like the shopping system, refer to 'the others.' Figure 3 shows the system of reservation.

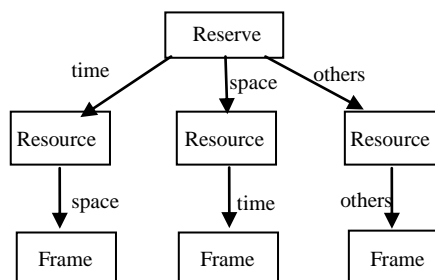


Fig.3. The system of reservation

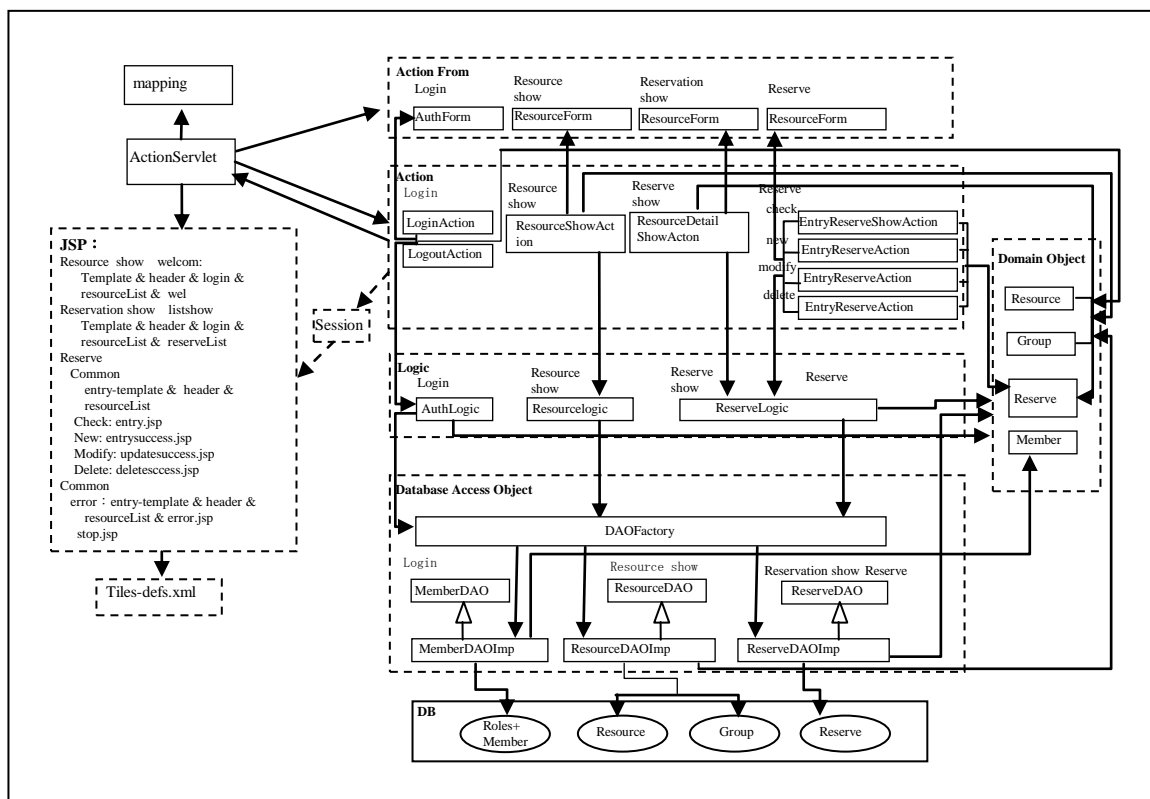


Fig.6. The extracted files for the framework

extracted.

Some of the front interfaces are shown in Figure 4 and 5.

3.2 Function specification

In reservation systems, there are common user functions and system administrator functions, respectively. The function specification is shown in Table 1.

Table.1. Function specification of reservation

	Common user	Administrator
User administrating	Login/logout	Login/logout User register/ modify/Delete
Resource administrating	-	Resource register/ modify/Delete
Reserve function	New reserve Reservation check/ modify/delete	New reserve All reservation's check/ modify/delete

Our framework only covers the most common user functions. These are user login and logout, a new reservation, and check, modify and delete of the reservations.

4. FRAMEWORK DEVELOPMENT

To develop the framework, first, a sample system is developed based on the architecture and function specification in Section 2 and 3. Then, the file is extracted from the sample system. Finally, the framework is realized completely.

4.1 A sample system

We chose a meeting room reservation system, which belongs to the time-based reservation category, as a sample system, from which the files the framework needs can be

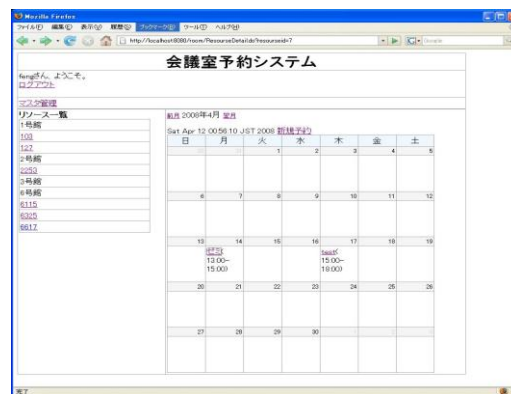


Fig.4. Reservations in monthly calendar format

The top page is made up of a header, a login part, a main part and a footer. When one of the meeting rooms has been chosen, the reservations of this room will be shown in monthly calendar format. Next the end-user can modify or delete the reservations when he/she logs in.



Fig.5. Details of a reservation

4.2 Extraction of the files

Based on the meeting room reservation system developed, the files which are needed in a reservation framework are extracted. The extracted files are shown in Figure 6.

4.3 Completion of the framework

When the files have been extracted, the detailed contents of each file should be added and modified so that they can be used as a framework.

We modified some of the contents of the file so that they can be overridden as base classes or configure files. And some of the files were made into libraries so the developers can choose to use. The details of the framework will not be included in this paper, however, except the libraries from the inner framework Struts, the number of the original steps written by us is 996.

5. REUSABILITY EVALUATION WITH APPLICATION TEST

In the former sections, we developed a reservation framework. Next, we will carry out applications to see if it is usable in reservation systems. As the sample system from which the framework was extracted belongs to the time-based categories, we apply it to two systems belonging to space-based reservation and the others, respectively.

5.1 Application test in space-based category

In the first application test, we will see if our framework is reusable in the space-based reservation category

A soccer ticket reservation system was chosen. The end-user first chose the match according to the time, then the seat which is a space notion will be reserved. The functions as a reservation system to common users such as reservation check, modification, deletion and so on are realized.

The percentage that the framework supplied in the system is shown in Table 2.

Table.2. Reusability of application test 1

	Action	Action Form	Doma in object	Logic class	DAO	View	Action Mapping	Total
Overall system	290	94	182	139	419	397	70	1591
Framework	245	47	84	90	294	175	61	996
specialized	45	47	98	49	125	222	9	595
Percentage of framework	84%	50%	46%	65%	70%	44%	87%	62%

According to Table 2, we can see that as a total the system has 1591 steps, among those 996 steps are supplied by the framework. In other words, a 62% reusability is realized. So, the reusability in space-based reservation systems is indeed confirmed.

5.2 Application test in the others category

Following the first application test, the second one will be carried out on the others category.

This time an online book store system is chosen. As nothing to do with time or space, the end-user chose the book first,

then, decided how many of the same book to buy. In addition, the other functions of reservation plus a search function was realized.

The percentage that the framework supplied in the system is shown in Table 3.

Table.3. Reusability of apply test 2

	Action	Action Form	Doma in object	Logic class	DAO	View	Action Mapping	Total
Overall system	286	84	168	124	402	402	70	1536
Framework	245	47	84	90	294	175	61	996
specialized	41	37	84	34	108	227	9	540
Percentage of framework	85%	56%	50%	72%	73%	43%	87%	65%

As shown in Table 3, among the 1536 steps of the total system, only 540 steps are customized. According to a reusability of up to 65%, we confirmed that the framework is also reusable in the others category.

5.3 Overall evaluation

Including the meeting room reservation sample system, the reusability of our framework is shown in Figure 7.

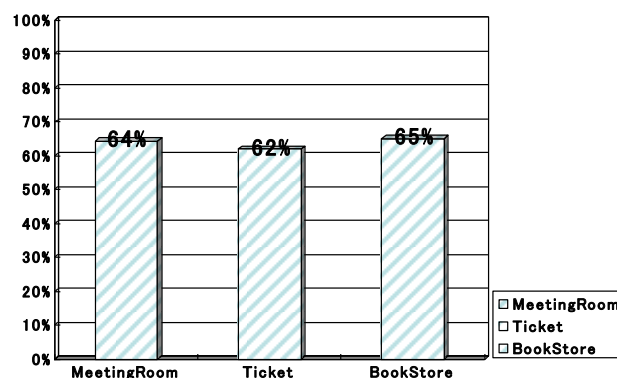


Fig.7. Reusability of the 3 systems

The percentages of the code that the framework supplied, are 64%, 62%, and 65% respectively and the high reusability was confirmed.

Also, it has the following advantages.

After one system from one category of the reservation system has been applied respectively, we confirmed that our framework can be applied to a wide range of reservation systems.

Because the architecture has been well designed on the framework level, the work can be released to the developers. As a result, a new well-designed system can be developed much faster, and easier for extension and maintenance.

6. THE TRADE-OFF RELATIONSHIP

Because we narrowed the domain to reservation, our framework can gain a much higher reusability than the general

frameworks. As all of us can see, the narrower the domain is, the higher the reusability will be. In this section we will develop another framework in a more narrowed domain, the time-based reservation framework, which is only one category in the reservation system, to confirm the trade-off relationship between the range of the domain and the reusability. The purpose of this section is to figure out the parts that can be made more reusable in a domain specific framework.

6.1 A time-based framework

A new framework for time-based reservation is developed. Because it is a child domain of reservation, we developed it based on the same sample system, meeting room reservation system, which also belongs to this domain. Compared to the reservation framework, the new parts added in this framework include the repeated time check function, calendar page generating library and so on. The number of steps for the new framework becomes 1227.

6.2 Application test

The new framework will be applied with a system named classroom reservation system. The main difference to the meeting room reservation system is that the reserve frame becomes time period, rather than the real time. Besides this, there are also other differences such like the information needed for a reservation.

The percentage that the new framework supplied in the system is shown in Table 4.

Table.4. Reusability of the new framework

	Action	Action Form	Doma in object	Logic class	DAO	View	Action Mappi ng	Total
Overall system	296	98	119	116	396	459	63	1547
Framework	245	47	84	100	316	372	63	1227
specialized	51	51	35	16	80	87	0	320
Percentage of framework	83%	48%	70%	72 %	79%	81%	100 %	79%

As we can see, the percentage of the code supplied by the framework is higher, up to 79%. So if a time-based reservation system needs to be developed, this new framework in a narrower domain is more useful.

6.3 The trade-off relationship

After the framework for reservation and another time-based reservation framework as a child domain of reservation, have been developed and applied with several applications, we will compare them in each part.

Figure 8 shows the compared results. The figures of the percentage of the reservation framework refer to the average of the two application tests - the soccer ticket reservation system and the online book-store.

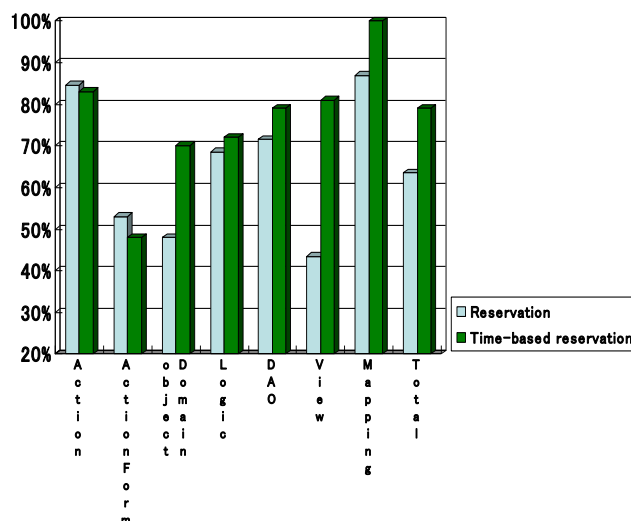


Fig.8. Reusability comparison

From Figure 8, we can see that, the view part changed mostly, that is to say that when the domain is narrowed, we can make this part more reusable with the framework.

As for the domain objects, the logic classes and the action mapping are also where we can pay more attention to make the framework more reusable.

From the details of the DAO classes, we know that only the properties of the entities are different. So we will develop a visual tool to generate the DAO code in the next section.

7. A VISUAL CODE GENERATION TOOL

The database access layer processes the data with the relational database. The basic functions of this layer include the functions to create, read, update or delete one or several rows of data in the database. If these functions are realized in this layer, then all of the other complex functions can be realized in the logic layer using these methods.

Since we have learned that only the properties of the entities are needed to be customized in this layer, and the minimal necessary entities needed in a reservation system using our framework are member, group, resource and reservation, we developed a visual tool so that the source code of this layer can be generated automatically.

The source code generated by our visual tool includes the database processing functions, those are creating, updating, deleting one row in the relative table, and retrieving one or several rows from the relative table. Since the domain objects will also be generated, the retrieved data will be returned as an object or a list of objects, so the object relational mapping function has also been realized with our visual tool.

A sample processing flow of the visual tool is shown in Figure 9. While all of the 4 entities needing to be customized, Figure 9 shows the processing flow of only one - Reservation. When the Reservation button is clicked on the top page, the customizing page will be shown. As the minimum necessary properties have already been added by the tool automatically, the developers only need to add the other properties like date,

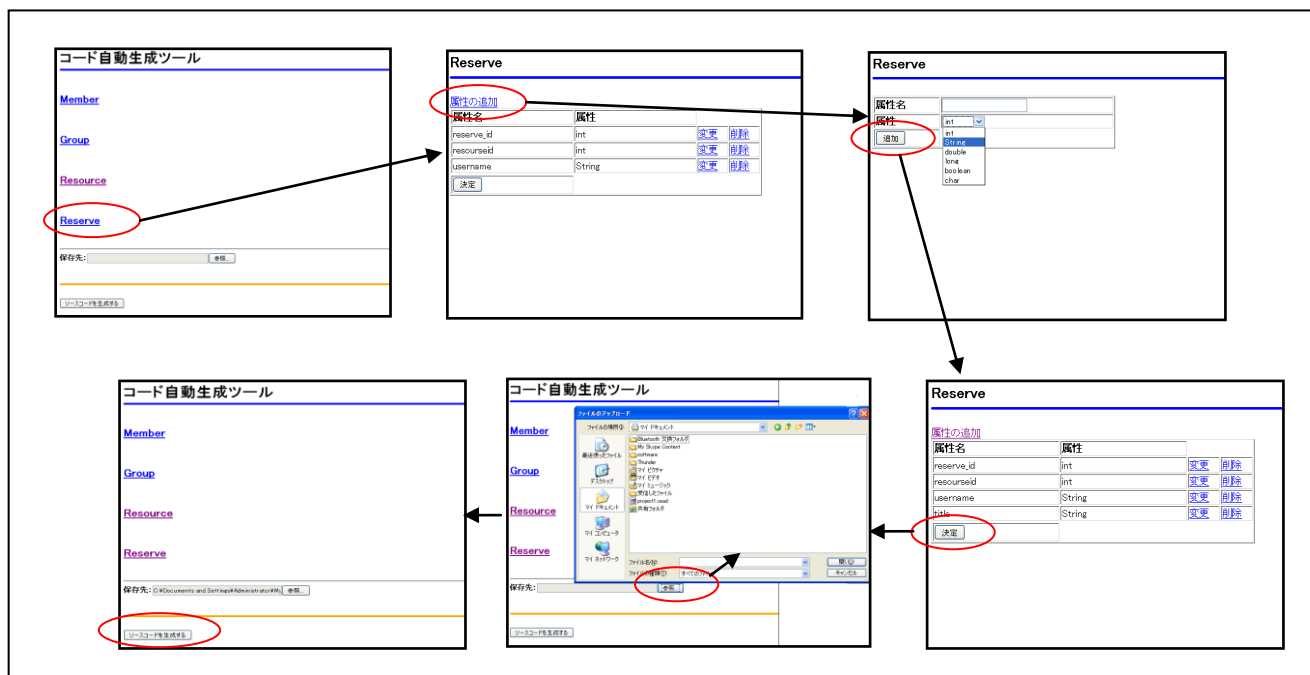


Fig.9. The processing flow of the visual tool

address, postal-code, they want. After the properties have been added, the location where the source code will be generated should be chosen. And after all the other 3 entities are customized similarly, the source code can be generated by clicking the generate button on the top page.

We take the online book store for an application test. After the DAO and domain objects source code are generated by the visual tool, the other parts were also developed, and the total system works as well as if it were all coded by hands.

Since the code for the data access layer and domain object were generated automatically, the reusability rate is higher up to 77.3% in this test. With the object relational mapping function, we can develop the other layers more conveniently by only using object oriented techniques.

8. CONCLUSION

In this paper, we developed a framework for reservation systems, with a visual code generation tool. Through the application tests, the high reusability in a wide range has been confirmed. In addition, the parts that can be made more reusable in a domain specific framework have also been confirmed, through the comparison between two frameworks.

REFERENCES

- [1] T. Chusho, H. Tsukui, "Reusability of Web application framework for reservation systems" (in Japanese), IEICE, Vol.J88-D-I, No.5, pp.930-939, 2005
- [2] F. Zhou, T. Chusho, "The reusability evaluation of a domain-specific Web application framework", (in Japanese), IPSJ SIG Technical Report, pp. 63-70, 2008
- [3] The Apache Software Foundation, <http://struts.apache.org/>
- [4] Red Hat, Inc, <http://www.hibernate.org/>
- [5] T. Matsutsuka, "A framework for AJAX-enabled business applications", (in Japanese), IPSJ, Vol.49 No.7, pp.2360-2371, 2008
- [6] J. Zhao, N. Xiao, "Study on Framework of Scalable High Performance Multi-layer and Multi-pattern Information Systems" Information

Processing (ISIP), 2008 International Symposium Volume , Issue , 23-25, 2008

- [7] Deepak Alur, John Crupi and Dan Malks, Core J2EE Patterns: Best Practices and Design Strategies, Prentice Hall/Sun Microsystems Press, 2003.
- [8] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software, Addison Wesley, 1995.