Reusability Evaluation of a Domain-Specific Web Application Framework

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, we made a proposal of a web application framework in a specific domain so that its reusability can be higher than the common frameworks . After the well designing of its architecture, a framework for reservation was developed based on a meeting room reservation sample system. Then, the framework was applied to two types of reservation systems, an online book store system and a soccer ticket reservation system, and its reusability was evaluated. With the result of 62% and 65% respectively, high reusability of the framework has been confirmed. In addition, in order to make the influences to each part by the range of domain clear, 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 has been developed to generate the source code for the database access transaction.**

Keywords: framework, domain-specific, reservation, reusability, trade-off, code generation

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, and the cost will be very high. So there are lots of support technologies, such as design pattern[1], libraries, component[2] and so on. 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.

There are lots of already existing general frameworks like Struts[3] and Hibernate[4]. However, because they all aimed to be applied in wide areas of domains, the reusability of them are not so high. In our research[5][6][7] until now, compared to the more general frameworks, 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[8]. 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[9]. A framework in the domian of Web service, which can be used to execute adaptive service processes[10].

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

2 Framework Realization

To develop our framework, first,the specific domain, that is reservation, must be defined. And then, the architecture should be designed, after a sample system having been developed, the files that are needed in the framework will be extracted, and finally, the framework will be realized by some additions and modifications.

2.1 Domain Definition

We choose the "Reservation" systems as the specific domain in which our framework can be applied to.

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 1, there

*F. Zhou and T. Chusho are with the Department of Computer Science, Meiji University, Kawasaki, Japan. And F.Zhou joined the Software Division, Hitachi,Ltd from April, 09.

are end-users who follow certain conditions to make the reservation. There are resources being divided into reservable frames.

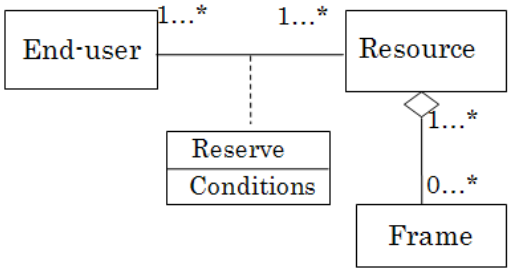

Figure 1: Conceptual model of Reservation

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 2 shows the system of reservation.

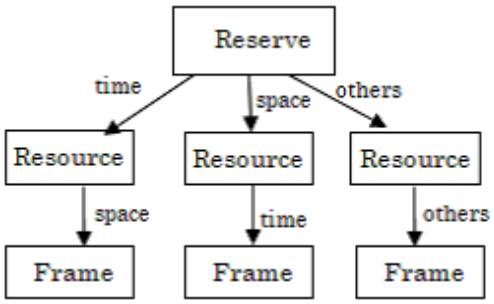

Figure 2: The system of Reservation

2.2 Function Specification

Since the domain has been specified, we can specify the functions that are needed in all of the reservation systems, so that these functions can be provide by our frmework as liraries or components, and the control flow of them also can be provided.

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

Until now, our framework covers the common user' functions. These are user login and logout, a new reservation, and check, modify and delete of the reservations.

Table 1: Function specification of reservation

	Commmon user	Administrator
User administrating	Login/ Logout	Login/Logout User register/ modify/delete
Resource administrating	-	Resource register modify/delete
Reserve functions	New reserve Reservation check modify/delete	New reserve All reservations' check modify/delete

2.3 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.

2.3.1 3-layer architecture

3-layer architecture is the most popular web application architecture, in which, the presentation, the application processing and the data management are logically separate processes. To take more advantage of this architecture, we follow the following principles to design the framework 's architecture.
1. The lower layer provides its interfaces to the upper layer. It's not aware of upper layers.
2. The upper layer doesn 't need to know the detail of the lower layer. The only necessary information is the lower layer 's interfaces.
3. The upper layer only needs to know the interface of the layer directly beneath it.
Following these principles, we are able to make our framework more understandable, extensible, and hence, more efficient.

The architecture in our framework is made up from the following 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.

2.3.2 Presentation layer

The presentation layer deals with the user interfaces. There are lots of already existing general frameworks for this layer, that can be be used in almost all of the domians. 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 appli-

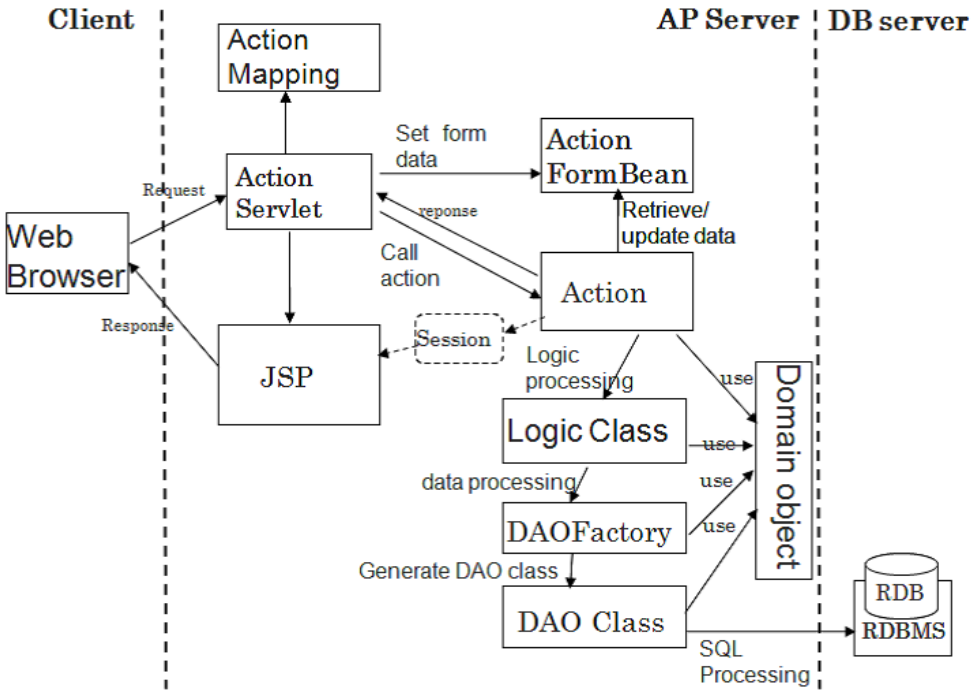

Figure 3: The architecture of our framework.

cations that utilize a MVC architecture, which can solve the above problems very well.

In our framework, we developed this layer based on Struts, using some of the libraries from it. So the MVC model is adopted in our framework.

2.3.3 Data 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[11] in this layer, and also in Section 5, a visual tool will be developed to generate the code automatically.

Design patterns are the abstract of the past experiences trying to solve some problems in a common model, they are very useful to help to design the system architecture.[12] 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. Because we only use relational databases, the DAO pattern with Factory Method pattern adopted is selected for our framework. In this strategy, a DAO Factory produces a variety of DAOs that are needed by the application, so it becomes much more convenient and easier for extension and maintenance.

2.3.4 Framework's architecture

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

As we can see in figure 3, with our framework, an application can be made up from 8 parts.

Action This part process a request, usually by calling logic classes, via its execute method, and return an ActionForward object that identifies where control should be forwarded to provide the appropriate response. The properly requests and reponses are based on the action mapping files. And the action classes also needs to get the data from the action forms.

Action Form This part provides the function that the data from the client be carried to the action classes for processing. And then, transfer data between the Logic classes and View layer.

Domain object Domain objects normally represented as a set of one or more JavaBeans classes. A shopping cart system, for example, will include a bean that represents the cart being maintained for each individual shopper, and will (among other things) include the set of items that the shopper has currently selected for purchase.

Logic class This part carries on the business logic processes of the application. It is called by the action classes, and usually need to call the DAO classes to exchangge data with the database.

DAO This part processes the data with the relational database. The basic functions are creating,reading, updating, deleting data with a database.

View The user interface part of the application.

Action Mapping These are XML files which have the mapping information between the action classes and the url.

DB The database where the data of the application are being saved.

2.4 Sample System Development

To extract the files that are necessary in a reservation framework, we chose a meeting room reservation system, which belongs to the time-based reservation category, as a sample system. Using the architecture we designed, and the detailed functions of meeting room reservation system based on the reservation functions in section 2.2, we developed the sample system. Figure 4 is an example of the GUI page which shows the reservations in a monthly calendar format.

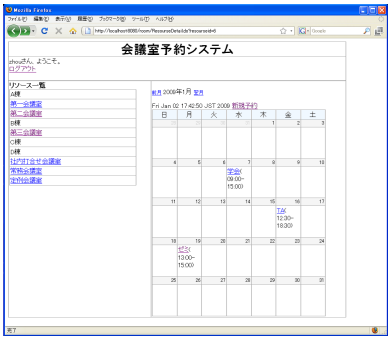

Figure 4: Reservations in monthly calendar format

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.

2.5 Files Extraction

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 5.

2.6 Framework Completion

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. And also, some of the files must be added. 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 detailed explanation of the framework is not the most important thing in this research, so it 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.

As a domian specific framework, some of the process flow have been decided. For example, in the case of checking the existing reservations, as being shown in figure 6, we should follow the following process flow.
(1) The request from the client is received by the action servlet class.
(2) The resource id that need to be checked is set to a correspond action form.
(3) The proper action class is called by the action servlet.
(4) By the action class, the logic class is executed.
(5) The logic class calls the DAO classes to fetch the data.
(6) The DAO classes exchange data with the database.
(7) The data is returned to the logic class, and will be made some logical process if necessary.
(8) The data is returned to the action class.
(9) The proper view page is generated.
(10) The html page will be send to the client as a response.

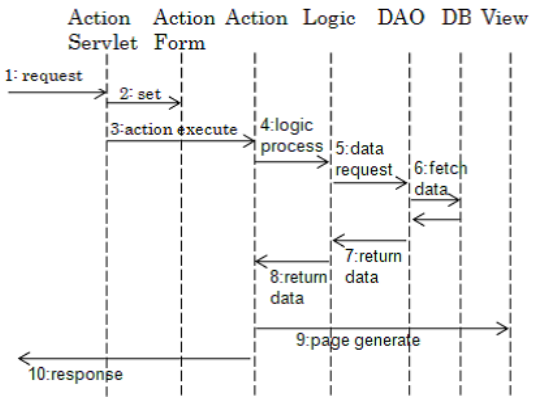

Figure 6: The flow of processing

3 Reusability Evaluation

In the former sections, we developed a reservation framework. Next, we will carry on application tests 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. The

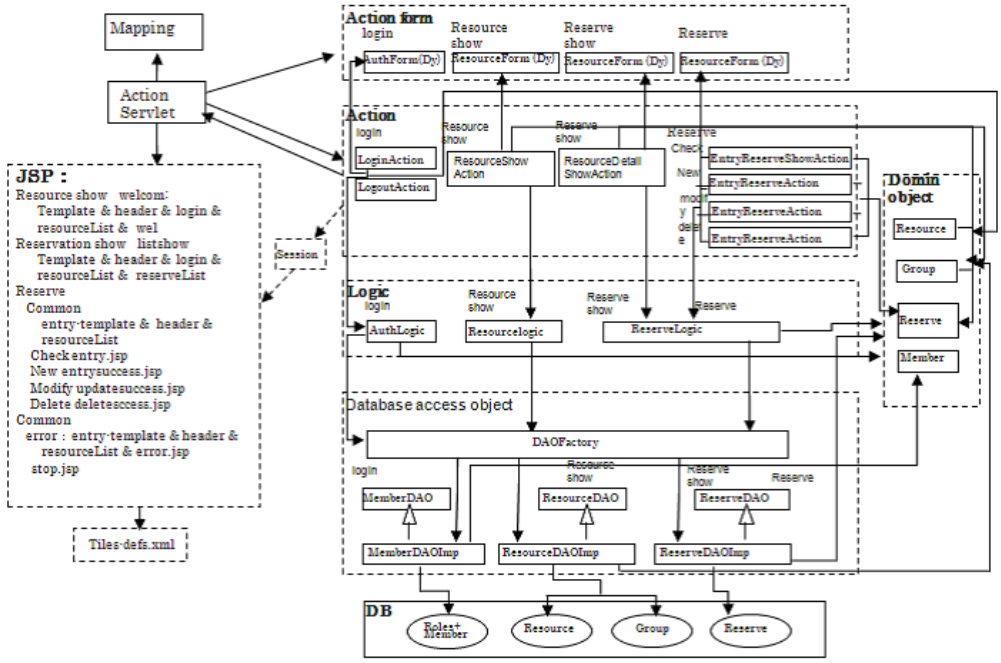

Figure 5: The extracted files for the framework.

main point is to check the percentage of the code that are provided by our framework in the whole application, in other words, the reusability of our framework.

3.1 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 has been 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.

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 62.6% reusability is realized. So the reusability in space-based reservation systems is indeed confirmed.

3.2 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.

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.

3.3 Overall Evaluation

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

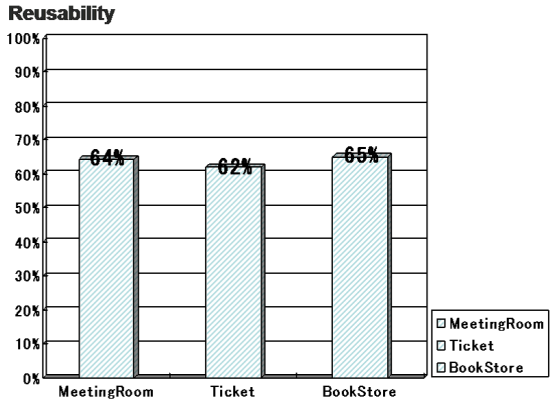

Figure 7: Reusability of the 3 systems

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

Table 2: Reusability of application test 1

	Action Class	Action Form	Logic Class	DAO Class	View	Domain Class	Action Mapping	Total
Overall system	290	94	139	419	397	182	70	1591
Framework	245	47	90	294	175	84	61	996
Specialized	49	47	49	125	222	98	9	595
Percentage of Framework	84.5%	50.0%	64.7%	70.2%	44.1%	46.1%	87.1%	62.6%

Table 3: Reusability of application test 2

	Action Class	Action Form	Logic Class	DAO Class	View	Domain Class	Action Mapping	Total
Overall system	286	84	124	402	402	168	70	1536
Framework	245	47	90	294	175	84	61	996
Specialized	41	37	34	108	227	84	9	540
percentage of Framework	85.7%	55.9%	72.6%	73.1%	43.5%	50.0%	87.1%	64.8%

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.

4 Reusability Examination:The Trade-off Releationship

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 make clear that how the change of the domian range influences each part of the framework, so that the parts that can be made more reusable in a domain specific framework can be figured out .

4.1 A Time-based Framework

As a narrower domian than the domian of reservation, we chose the time-based reservation. As we can see from Figure 2, it is a child domain of reservation.

The method to develop the time-based reservation framework is the same to how we developed the reservation framework. The steps are "Architecture design", "Sample system development", "Files extraction", "Framework completion". Because it is a child domain of reservation, and the meeting room system we developed also belongs to this domain, we based on this same sample system, to extract the files that are needed. The details will not be explained in this paper. However, 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 the original steps for the new framework becomes 1227.

4.2 Application Test

The new framework will be applied with a system named classroom reservation system, which belongs to the domain of time-based reservation. 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 to make a reservation and so on.

After the test application developed, we can see the percentage that the new framework provided in the system in Table 4.

As we can see, the percentage of the code provided by the framework is higher, up to 79.3%. So if a time-based reservation system needs to be developed, this new framework in a narrower domain is more useful.

Table 4: Reusability of the time-based framework

	Action Class	Action Form	Logic Class	DAO Class	View	Domain Class	Action Mapping	Total
Overall system	296	98	116	397	459	119	63	1548
Framework	245	47	100	316	372	84	63	1227
Specialized	51	51	16	81	87	35	0	321
Percentage of Framework	82.8%	47.9%	86.2%	79.6%	81.1%	70.6%	100%	79.3%

4.3 The Trade-off Relationship

After the framework for reservation and another time-based reservation framework as a child domain of reservation, having been developed and applied with several applications, we will compare them in each part, to see the influences to the reusability by the change of the domian range.

Figure 8 shows the compararision 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.

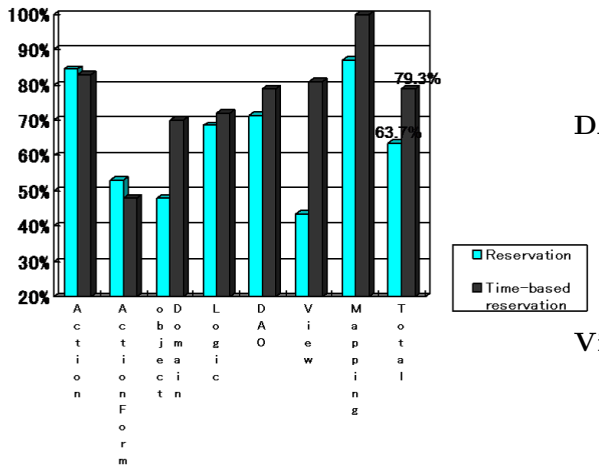

Figure 8: Reusability comparison

According to Figure 8 and the details of each part, Let's see how the range of domian influence the reusability of each part.

Action This part just calls the logic classes, gets the data from the action forms, and forwards the url, and as the reusability in a narrower domain even got a lower reusability, we know that there's no influences by the range of domian. Actually, because the logic classes have the common interfaces in our framework, the processes are very easy here, just calling these interfaces.

Action Form This part exchanges data between the client and the system, usually in the form of beans with getter and setter methods, so there's no influences by the range of domian.

Domain object The narrower the domian is, the more common properties the objects will have. So this part receives a strong influence by the domian of range.

Logic class This part carries on the business logic processes of the application. It is supposed to have a low reusability because each application has very different logic processes from another. However, in a specific domian, there're common functions, so we can provide some libraries and components to make the development work easier.

DAO The basic functions in this part are creating,reading, updating, deleting data with a database. In a narrowed domian, if the more complicated processes be dealed in the logic classes, we even can develop a code generation tool, only need to defined the properties of each entities by the users.

View The user interfaces vary from one application to another. So it got a very low reusability in the reservation framework. However, in the narrower time-based reservation framework, as some of the common pages can be generated automaticly, and some of the templates for the layout of pages can be provided, we got a high reusability. So it has a strong influence by the range of domian.

Action Mapping In a domian specific framework, lots of the control flows can be decided by the framework, so it got a high reusability. In a very low domian, even all of the control flows can be decided by the framework.

As we have confirmed the influences of the reusability to each part by the chang of domian range, we now know that we should pay more attention on the View, Domian object, Logic and DAO parts, when we are developing a domain-specific framework. And how each part can be made more reusable should also be noticed.

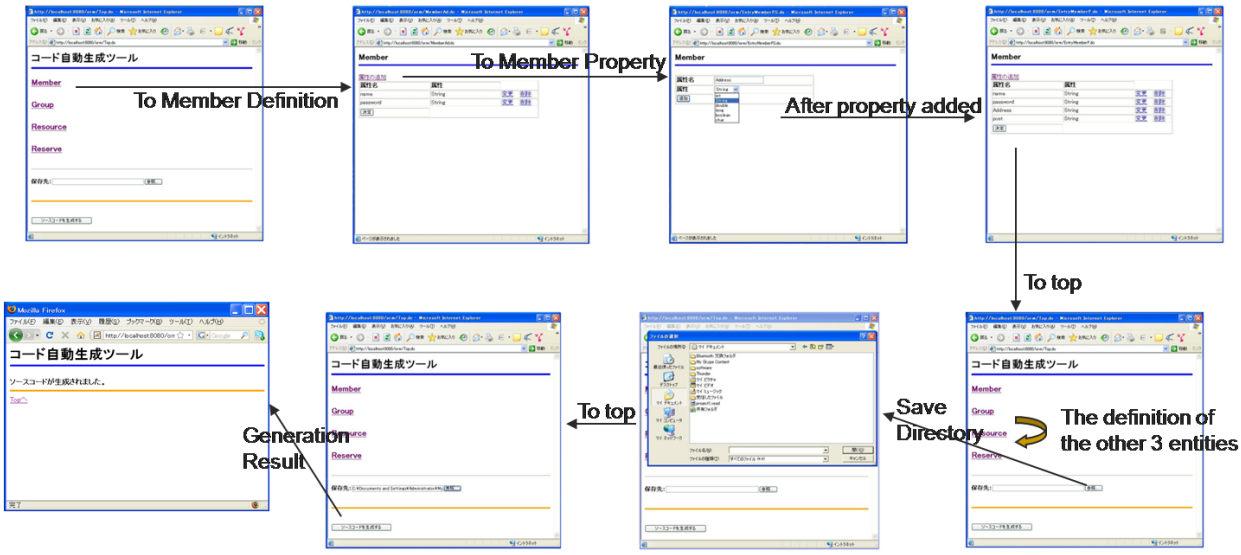

Figure 9: The processing flow of the visual tool.

5 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. Common technologies are such like JDBC, O/R Mapping, and so on, the disadvantage of them are that they are too difficult to master and use, so it needs skilled developers to take a long time to develop an application.

Since we have learned that 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, so 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 - Member. When the Member button is clicked on the top page, the customizing page for this entity 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 address, postal-code, and others they want. Also, the already defined entities can be midified on the customizing page. After the properties have been added, the location where the source code will be generated should be chose. And after all the other 3 entities having been customized similarly, the source code can be generated by clicking the generate button on the top page.

In order to see if this tool is usable, we take the online book store for an application test. After the DAO and domain objects source code are generated by this visual tool, the other parts were also developed, and the total system works as well as the system in section 3.2 which was all coded by hands on the base of our framework without the using of this tool.

Since the code of the data access layer and domain objects were generated automatically, the reusability rate of our framework with this tool is higher up to 77.3% in this test. And as the generated code also make a mapping between the objects and the relational database, we can develop the other layers more conveniently by only using object oriented techniques.

The advantage of this visual tool is that the developers even don‘t need to have any SQL knowledges, and with the visual interfaces, the development of the applicaiton becomes much easier and faster with less mistakes.

6 Conclusions and Future Work

In this paper, we made a proposal of developing a domain-specific framework. A framework for reservation systems have been developed, with a visual code generation tool. Through the application tests, the high reusability in a wide range has been confirmed. In addition, the influ-

ences to the reusability of each part by the domian range have also been confirmed, through the comparison between two frameworks.

The future work in our research includes the extension of the reservation framework, that is to cover not only the common users‘ functions but also the administrators‘ of reservation sytems, to provide more libraries and componets to make the reusability more higher , to make the visual tool more powerful and so on. Also, we are planning to provide the XML interfaces in the framework, so that the applications developed can be used as a service.

References

[1] Erich Gamma., Richard Helm., Ralph Johnson., John Vlissides., *Design Patterns: Elements of Reusable Object-Oriented Software*, Addsion Wesley, 1995.

[2] A.W.Brown., *Component-based software engineering*, IEEE CS Press, 1996.

[3] The Apache Software Foundation., http://struts.apache.org/

[4] Christian Bauer., Gavin King., *Hibernate in Action*, Manning Publications Co, 2004.

[5] T. Chusho., H. Tsukui., “Reusability of Web application framework for reservation systems,”(in Japanese) *IEICE Journal*, Vol.J88-D-I, No.5, pp. 930-939, 5/05

[6] F. Zhou., T. Chusho., “The reusability evaluation of a domain-specific Web application framework,”(in Japanese) *IIPSJ SIG Technical Report*, pp. 63-70, 11/08

[7] F. Zhou., T. Chusho., “A Web Application Framework for Reservation Systems and its Reusability Evaluation,” *Proc. The 2009 IAENG International Conference on Software Engineering (ICSE’09)*, pp. 1027-1032, 3/09

[8] T. Matsutsuka., “A framework for AJAX-enabled business applications,”(in Japanese) *IPSJ Journal*, Vol.49, No.7, pp. 2360-2371, 7/08

[9] J. Zhao., N. Xiao., “Study on Framework of Scalable High Performance Multi-layer and Multi-pattern Information Systems,” *Information Processing (ISIP), 2008 International Symposium*, pp. 23-25, 5/08

[10] Danilo Ardagna., Macro Comuzzi., Enrico Mussi., Barbara Pernici., Pierluigi Plebani., “PAWS:A Framework for Executing Adaptive Web-Service Processes,” *IEEE Software*, Vol.24, No.6, pp. 39-46, 12/07

[11] Deepak Alur., John Crupi., Dan Malks., *Core J2EE Patterns: Best Practices and Design Strategies*, 2nd Edition, Prentice Hall/Sun Microsystems Press, 2003.

[12] Linda Rising., “Understanding the Power of Abstraction in Patterns,” *IEEE Software*, Vol.24, No.4, pp. 46-51, 7/07