

Design and Development of Multimedia Information Systems* on Integrated Teaching and Education Environments

† Wuyi Yue, ‡Yasuhiro Umakoshi and ††Shinya Ooka

†Department of Information Science and Systems Engineering, Konan University

‡‡Graduate School of Natural Science, Konan University

† yue@konan-u.ac.jp, ‡ zashiki@anet.ne.jp, †† sin@yue.math.konan-u.ac.jp

Abstract

Development of efficient integrated teaching and education systems involves good knowledge of multithreading and network communication such as multimedia text, domains often closely related to the operating system. However, the reusability of components could be limited by some system specific techniques or limitations used in the design. To enhance the reusability and to easily reuse these multimedia integrated education and training systems mutually, this paper presents a new framework for design and development of multimedia information system on such integrated teaching and education environments. We use Java, Servlet, XML, design pattern technologies to develop system interface of the information system for knowledge sharing and reusing. By using this system, upgrades and extensions are simplified and ones can be improved, for instance, the message format is without knowing the way services work. On the other hand a developer in charge of maintaining or creating a new kind of service does not need to know how the connections are concretely established.

1. Introduction

Network environments are growing, fueled by extensive enhancements of computer hardware and software as well as the rapid growth of the Internet and World-Wide Web (WWW). More systems are being connected to networks, the education using a computer or the Internet carries out in a certain inside, advanced distributed educational environment being realized. Such growth impacts on the performance of many network related user applications.

Moreover, the improvement in the education and training effect by the useful information which can always also access somewhere is expected to the information on various databases, or the real world and fruit time [1], [2].

However, the software systems and databases which different teachers or system engineers can not be reused. Even if the advanced knowledge database system with high quality knowledge is realized, the share of knowledge is difficult, since construction of the educational

environment on the Internet is not easy. Moreover, since the system with the similar feature is developed in various places, systems are not unified, and they are not necessarily easy to use for users. This can be said that share use of a system was not completed since it was developing uniquely, respectively, but great cost started, and efficiency is also bad for each researcher and development person. On the other hand, it is not rare to become what being late of construction of a system etc. is conspicuous, and the contents of development are also restricted, and lacks in practicality in the place which runs short of special knowledge.

From these reasons, construction of the teaching materials database aiming at education and training support, a long distance communication system, and a sharable information system is demanded. By realizing a sharable information system, it is provided in the form where everyone tends to use the scientific information which educational facilities, such as a university, have, and a knowledge database, and it is expected that large interdisciplinary communication is attained.

The software pattern which enables it for people to share such scientific information and a knowledge database, and an information system, and to reuse them repeatedly is developed. This software pattern attracts attention as the method of reusing advanced knowledge efficiently especially in recent years.

Use peculiar to a pattern is caught in pattern inclination development, saying "If the result which one developer considered is abstracted in the intelligible form, other developers do not need to consider the same thing and can reuse." [3]-[5]. That is, for the purpose achievement of sharing and reusing knowledge, the activity which makes patters and reuses original Know-how of each development project is put into practice, and development work must be advanced for a pattern to a core.

In this paper, by applying the design pattern inclination development for the reuse in object-orientation for the purpose of solution of the above problems, it is easy to use for a student or a teacher, the system for making the environment in which share use is possible is built, and the modeling of the system is performed. And the system is implemented by utilizing the program mod-

*This work was supported by Special research grants for the development of characteristic education, the Promotion and Mutual Aid Corporation for Private Schools of Japan.

ule of Java language or Java language, XML, etc. [6], [7]. Moreover, extendibility is given to a system, and it designs so that various functions can be unified.

2. The problems on existing information system construction

As use of an information communication network, power is put into the teaching materials development which utilized base maintenance and multimedia contents for education for the lesson which utilized a teacher's intramural information infrastructure variably in educational facilities, such as a university. Also in it, the system which utilized various multimedia, such as a long distance report system, a long distance database search engine, and a long distance communication support system, is developed, and practical use is presented. Moreover, a teacher exhibiting the contents of a lecture on a homepage uniquely, or making a student utilize it and giving a lecture to him is advanced.

However, the experimental element of the educational environment using many information systems is also strong, each measure has been independent completely, and it is not necessarily easy to use for the side of users. The problems are pointed out as follows. For example: as the problems first seen from the use side:

- (1) If URL (Uniform Resource Locator) for every homepage of a subject is not recorded to somewhere, even if it is widely opened to the Internet, only lecture time is used.
- (2) A report system exists for every subject and the password must also be memorized to each.
- (3) Users have to reinput personal information repeatedly for every subject.
- (4) Since the separate system is used for every subject, Users have to memorize how to use each system. and so on is mentioned. Moreover, they are as problems seen from the teacher side:
- (5) It takes time, in order to have to begin from the system development for exhibiting the contents of a lecture or collecting reports using the Internet.
- (6) Since form unification takes time or the knowledge of a network is searched for, common use of knowledge and contents becomes difficult.
- (7) Creation of teaching materials is greatly dependent on the information system, and the reuse is very difficult. It needs great cost and is always asked for correction and an improvement in the field especially with an intense change of the contents, and teaching-materials offer becomes difficult are mentioned.

It is asked for support of the teaching materials development process in accordance with target clarification and target it what thing the information system of education and training support should be in order to cope with these problems.

3. Modeling of system

Since the present system corresponds to the above situations, construction of the environment which is easy

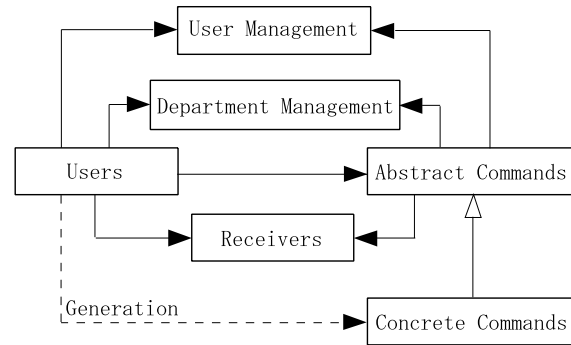


Figure 1: The whole system class figure

to use where various functions were unified, i.e., the environment with the two directions which can send information also not from one direction but from other side of users, is considered. Furthermore, it is desirable for a system developer's original functional addition to be possible.

3.1 Class

The class specifies the object's internal data and representation and defines the operations the object can perform. Information and operation required for a system are collected by the defined class. This system model consists of six classes, a user class, a user management class, a department management class, an abstract command class, a concreteness command class, and a receptacle side class.

The object to a certain class is defined as an instance. The process which builds a system model is taken, generating these instances. An instance is the model of the substance made inside a computer, and the class has described the restrictions which the instance itself and the relations between them (connection, how to combine, etc.) should fill.

The class figure of the model of this system which shows the logical and static relation between these six classes is shown in Fig. 1 using modeling capacity UML (Unified Modeling Language).

The message actually exchanged between objects is shown in Fig. 2 (the user to whom the underline is attached in the figure, user management, a concreteness command, etc. express an object). While is related, a message is sent to another class from a class, or the arrow in a class figure shows that it is possible to refer to information. Moreover, it is shown that the dashed line arrow in a sequence figure is the return value to the object of a calling agency.

Two or more concreteness command classes in a class figure exist in fact. It means that it can have two or more commands, and there is. The relation between classes shows from Figs. 1 and 2 between these classes what interaction will be done.

Next, the details of each class are explained.

- (1) User class :

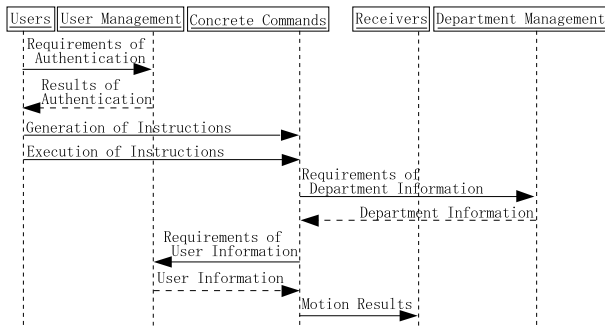


Figure 2: The whole system sequence figure

The portion which consists of students and teachers using service.

- (2) User management class :
The portion which manages students and teachers
- (3) Department management class :
The portion which manages the subjects belonging to the department.
- (4) Abstract command class :
An abstract class is one whose main purpose is to define a common interface for its subclass. An abstract class will defer some or all of its implementation to operations defined in subclasses; hence an abstract class cannot be instantiated. The portion which collected the information and operation common to all services.
- (5) Concreteness command class :
Classes that are not abstract are called concrete classes. The portion which inherits the function of an abstract command class and offers concrete service
- (6) Receptacle side class :
The receptacle side of service (the usually same student as a user class, and teacher)

3.2 The instance of each class

Objects are created by instantiating a class. The object is said to be an instance of the class. The process of instantiating a class allocates storage for the object's internal data (to make up of instance variables) and associates the operations with these data. Many similar instances of an object can be created by instantiating a class. The instance of each class is described as follows:

- (1) Instance of a user class :
Since two or more students and teachers share the instance of a user class, who cannot discriminate whether it is going to demand service. Then, the instance of this class uses a password etc. for the instance of a user management class, and receives attestation, and attestation demanded. Next, this class generates and performs the instance of the concreteness command class which the user who

received attestation. A command of the user who has not received attestation here is not executed.

- (2) Instance of a user management class :
As for the instance of a user management class, only one is generated. The mismatching of user information etc. is avoided by making it such. The instance of this class replies to a demand of the personal information from meeting the attestation demand from the instance of a user class, and the instance of a concreteness command class.
- (3) Instance of a department management class :
The instance of a department management class accepts a demand of the lecture name list by the subject of study from the instance of a user class, the list of concreteness commands according to the subject of study, etc. The instance of a user class generates the instance of a concreteness command class according to the list obtained from the department management class. As for the instance of this class, only one is generated to one user class. It is for the reason's being the same as that of the case of a user management class, and avoiding the mismatching of user information etc.
- (4) Instance of an abstract command class :
An abstract class is one whose main purpose is to define a common interface for its subclass. An abstract class will defer some or all of its implementation to operations defined in subclasses; hence an abstract class cannot be instantiated. The operations that an abstract class declares but does not implement are called abstract operations. Classes that are not abstract are called concrete classes. The instance of an abstract command class is not generated, but a concreteness command class inherits the function which this class has, and the instance of a concreteness command class is generated. Thus, by making the common portion of a concreteness command class into an abstract command class, a user class can use any commands by the same method.
- (5) Instance of a concreteness command class :
The instance of a concreteness command class inherits the function which an abstract command class has, and is generated. A concreteness command class plays the role which is connecting with an external database, mail service, etc., and absorbs various services on a network to this system.

As mentioned above, in this system, since the operation method of all services is unified by the abstract command class, from the user side, it seems that the system was unified. However, the developer of a system can add various functions freely by inheriting an abstract command class and making a concreteness command class. Moreover, since the instance of a department management class has managed, the added function can be freely used in every subject of study. Furthermore, functions, such as dispatch of the information

from a user, are realizable with extension of this concreteness command class.

3.3 Pattern inclination development

A pattern describes and reuses the problem which arises in software development repeatedly, and the solution method which everybody has used to it repeatedly.

Generally, a design pattern has two techniques and gives a remedy. One is having given the name to each pattern and another is enabling it to check an intention of the pattern always quickly by the brief target description. In this way, it is enabled to write down an intention of a designer efficiently by describing the name at the program about the part which applied the design pattern in system modeling.

3.4 Application of a design pattern

Each design pattern systematically names, explains, and evaluates an important and recurring design in object-oriented systems. Design patterns make it easier to reuse successful designs and architectures. Put simply, design patterns help a designer get a design “right” faster. In case the modeling of this system is carried out, a design pattern is applied as follows.

- (1) The problems on the modeling which should be solved will be taken up and recognized.
- (2) The purpose is clarified and the pattern that is considered to be required for a while is chosen.
- (3) The patterns which should be used have to be decided following what the situations are in which the design pattern can be applied, what examples of poor design that the pattern can address are, how you can recognize these situations.
- (4) Classes are constituted. And by verifying its effect to judge whether the selected pattern is correct.
- (5) The selected patterns are applied to design and implemented.
- (6) The pattern which should be applied to the next is chosen from the related figure among related patterns and design patterns. Then to return to (2) and repeat this procedure of the series.

3.5 Singleton pattern

Ensure a class only has one instance, and provide a global point of access to it. In the management department of users of this system a design pattern called singleton pattern which is one of generation patterns is used.

The singleton pattern is a pattern which guarantees that there is only one instance in a class, and conservativeness and its reusability increase by using this pattern. By using the singleton pattern for the user management class, this system is raising the conservativeness and reusability of user management.

We should use the singleton pattern when there must be exactly one instance of a class, and it must be accessible to clients from a well-known access point. And

when the sole instance should be extensible by subclassing, and clients should be able to use an extended instance without modifying their code. When it mounts the class by the singleton pattern in this system, the constructor which is the command which generates a class is processed as an error, and this class has the instance generation operation command which guarantees that there is only one instance of a class.

3.6 Command pattern

The design pattern used in the execution portion of a command is a command pattern which is one of behavior patterns. The command pattern is a design pattern with which the side which demands a command enabled it to perform the operation without the knowledge of the object which receives a command by various concreteness command classes inherited from the abstract command class defining the command. By using a command pattern, clarification, ordering, and execution of a command can be made to be able to become independent, cancellation and re-execution of a command can be performed, the system can be structured by advanced command etc. from a fundamental command.

The class figure and sequence figure of the command pattern are shown in Figs. 3 and 4.

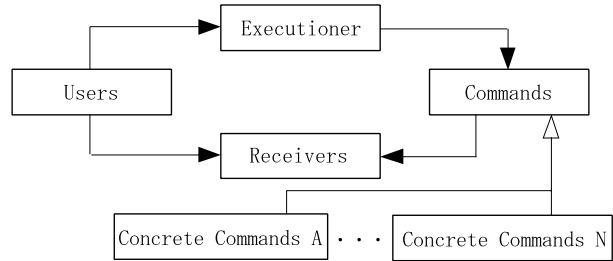


Figure 3: Command pattern class figure

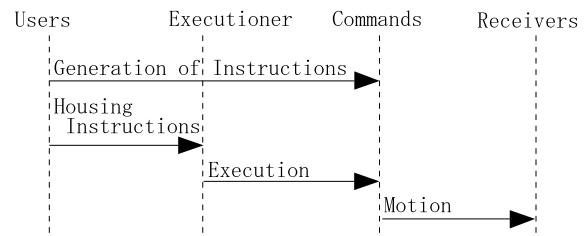


Figure 4: Command pattern sequence figure

Along with the flow of the above sequence figure, we explain the design pattern. The instance of the user class will generate the instance of the command class. When the instance of the command class is generated, it receives as an argument and the instance of the receptacle side class is passed. Next, the instance of the user class stores a command in the instance of an enforcement person class. The instance of the enforcement person class executes a command.

The central part of this system is constituted using this design pattern. By doing in this way, it can use in this system, without a user class being conscious of the database which a concreteness command class processes, the connection with other WWW servers, etc. Moreover, a system is also smoothly extensible by creating much concreteness commands if needed.

4. Implementation of the system

The technology used with implementation of the system in this section is described. The class figure of the whole system is shown in Fig. 5, and it is shown whether each class is arranged in it at what machine.

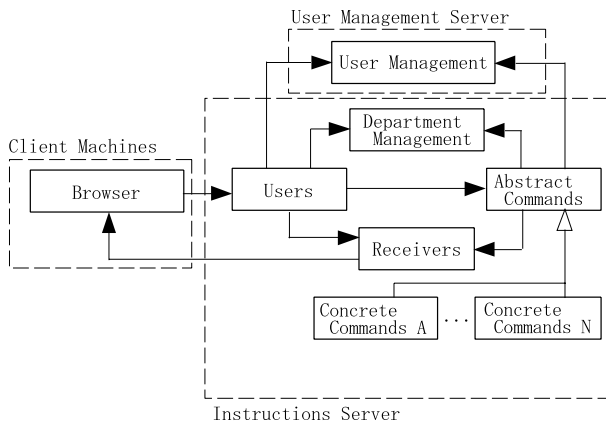


Figure 5: The whole implementation system class figure

It carries out as follows about implementation of each server in the system.

- (1) First, in order that the server for a command may build the service on a server supposing the WWW browser with a common client, Java Servlet is used and implemented. Java Servlet is a system which sends the document generated by the program module made by Java language to WWW server to be the machine of the side which offers WWW service. By this system, the execution result of the program module written in Java language can be sent to WWW client using HTTP (Hyper Text Transfer Protocol).
- (2) A user management server is implemented using RMI (Remote Method Invocation) server by using Java, or Java Servlet.
- (3) Communication of the server for commands and clients is performed using HTTP, and the user management server and the server for commands communicate using JRMP (Java Remote Method Protocol), or HTTP and XML according to implementation of a user management server.
- (4) In order that a concreteness command and a user management server may manage the data, a database management system is used, and it connects with the data using interface specification

JDBC (Java Database Connectivity) for accessing a database server from Java Servlet.

The technologies in which a server can demonstrate the effect and function most is chosen as implementation of each server in the system by this research, and the system is implemented using the technologies.

The technologies used for implementation of this system are Java language, JDBC (Java Database Connectivity), RMI (Remote Method Invocation), XML (Extensible Markup Language).

By this research, it considers making the most of a general thing and the thing standardized in case of implementation. By doing in this way, it is thought that the situation which is not desirable for many users is avoidable as much as possible.

5. Application of the system

5.1 The programming method

The examples of description in the case of programming this system first using Java language is shown. However, these are written for explanation of the method of programming, and since they are not perfect programs, they cannot be performed as they are.

Program.1 [UserManager.java] mounts a user management class. In Program.1 [UserManager.java], the handle to the one and only instance in the system is held to **theInstance**, and it is always made to return the handle to the same instance by **getInstance** method. Program.1 [UserManager.java] is shown as follows.

Program.1 [UserManager.java]

```

public class UserManager
{
    static UserManager theInstance = null;
    public static UserManager getInstance()
    {
        if (UserManager.theInstance == null)
        {
            UserManager.theInstance =
                new UserManager();
        }
        return UserManager.theInstance;
    }
    private UserManager()
    {
        // Instance generation processing
    }
}
  
```

Program.2 [EducationalSystem.java] mounts a user class (serve let). This class performs processing which serves as the entrance from a client, accepts user attestation processing and a demand of a user, and generates and performs a concreteness command class. Program.2 [EducationalSystem.java] is shown as follows.

Program.2 [EducationalSystem.java]

```

import javax.servlet.*;
import javax.servlet.http.*;
public class EducationalSystem
    extends HttpServlet
  
```

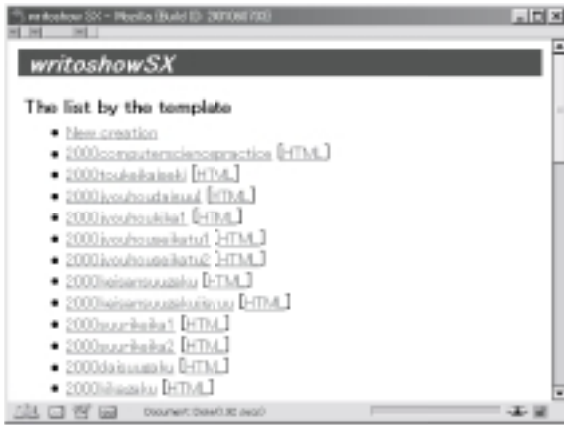


Figure 6: The main screen of the system

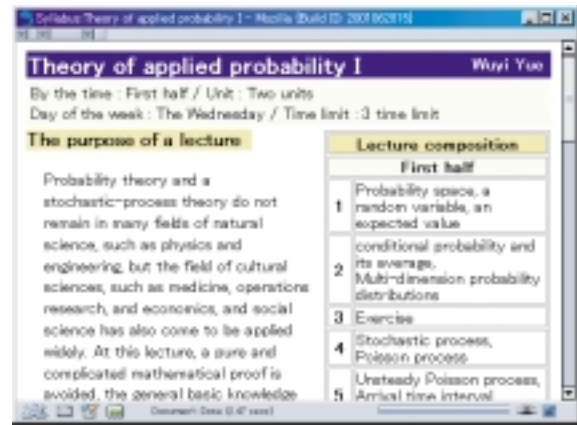


Figure 7: The mold of the document of a main screen

```

{
  service(HttpServletRequest request,
           HttpServletResponse response)
  throws ServletException, IOException
  {
    // User attestation processing etc.
    String cmdName =
    // The name of a command class is acquired.
    (ex. exampleConcreteCommand)
    EducationalCommand eCmd = Class.forName
    (cmdName).newInstance(response);
    eCmd.executeCommand(request);
  }
}

```

A department management class can be implemented like a user management class, and a receptacle side class is implemented using a javax.servlet.http.HttpServletResponse class.

5.2 The examples of application of the system

The creation of electronic teaching materials which used this system here, and the examples of application of the Internet public presentation are shown.

Fig. 6 is the main screen of the system. The menus (subject list etc.) currently displayed read the mold of the document of the main screen are shown in Fig. 6.

The mold of the document of a main screen is shown in Fig. 7. The item displayed on a main screen is changed by rewriting this file. The mold of a document is XML document according to the original rule of this system, and a conversion law is described according to XSLT (XSL Transformation) advised by World wide web consortium.

6. Conclusion

In this paper, the share of the teaching materials data system aiming at education and training support or a long distance communication system and the system for making the environment where share use can be performed were built by applying the design pattern inclination development for the reuse in object-orientation. The system was implemented by utilizing the program

module of Java language considered that each server in a system can demonstrate the effect of a server most, Java language, XML, etc. The system proposed by this research is unifying various functions, and it is easy to use it for a user, and it has the feature that the side which offers service can extend the service freely. Since a share of an information system can be realized and large intellectual communication is attained by not remaining within the campus but exhibiting educational environment on the Internet using the information system proposed by this research, the measure of the learning which stood on the large view by the students or the teachers is expected. Moreover, using the system by which modeling was carried out in this way brings integration nature to the educational environment using the Internet, and it is considered to lead to offering the educational environment which is easy to use for a user.

References

- [1] Japan Information Processing Development Center, "Informatization White Paper1999", Computer age company, 1999.
- [2] Shinji Heki, "Development of the Database System for Syllabus Internet Public Presentation", The Information Educational Method Research, Vol. 2, No. 1, pp.61-66, 1999.
- [3] E. Gamma, R. Helm, R. Johnson and J. Vlissides, "Design Patterns: Elements of Reusable Object-Oriented Software," Addison-Wesley, Reading, Massachusetts, 1995.
- [4] Shin Sahara, "Design Pattern: An Object Oriented Analysis and Design Capacity," Software Research Center, 1999.
- [5] Yuko Nakayama, et al., "The reuse of pattern soft development know-how 1st - 5th," Information Processing Society of Japan, Vol. 41, No. 1-5, 2000.
- [6] Tomoharu Asaumi, "Java Distribution Application Ultimate Distributed Environment Exceeding a Generation," Java Press, Vol.4, Technical Criticism Company, 1999.
- [7] Hiroshi Maruyama, Taketo Tamura, Naohiko Uramoto, "Web Application Development by XML and Java", Pearson Education, 1999.