

Intelligent Tutoring System of Primary Mathematics Based on EGL and Web 2.0

Bo Song^{*}, Xue Yan

College of Software, Shenyang Normal University, Liaoning Shenyang, China

Email address:

songbo63@aliyun.com (Bo Song), yanxue19900912@126.com (Xue Yan)

^{*}Corresponding author

To cite this article:

Bo Song, Xue Yan. Intelligent Tutoring System of Primary Mathematics Based on EGL and Web 2.0. *Software Engineering*.

Vol. 4, No. 2, 2016, pp. 34-40. doi: 10.11648/j.se.20160402.15

Received: April 1, 2016; **Accepted:** May 7, 2016; **Published:** May 9, 2016

Abstract: In this paper, an application architecture of intelligent tutoring system of primary mathematics based on EGL (Enterprise Generation Language) and integrating Web 2.0 technology is proposed. Through the process of design and implementation of intelligent tutoring system of primary mathematics, the key feature of the architecture is demonstrated – developers can focus on the business issues what code handle without caring for software technical details. The architecture is simple, easy to use and across languages, frameworks and runtime platforms. In addition, it can reduce the cost during the development stage of application and effectively improve the real-time requirements and human–computer interaction experience of intelligent tutoring system of primary mathematics.

Keywords: EGL, Web 2.0, Intelligent Tutoring System, Architecture

1. Introduction

In general, the intelligent tutoring system (ITS) refers to use of artificial intelligence technology that allows a computer to play the role of teachers implement personalized instruction to learners with different needs and different characteristics of the transfer of knowledge, and to provide a guide adaptive learning system [1]. With the development of Web 2.0 technology, the rich client in RIA (Rich Internet Application) is rapidly replacing the thin client in B/S [2]. Because the intelligent tutoring system can provide richer end-user experience, it has been adopted by more and more developers based on RIA architecture. For intelligent tutoring system, the main benefit of Ajax is a greatly improved user experience. Although JavaScript and DHTML - the technical foundations of Ajax - have been available for years, most programmers ignored them because they were difficult to master. Although most of the Ajax frameworks available today simplify development work, you still need a good grasp of the technology stack. So, if you're planning to use Ajax to improve only your application's user experience - if you're not also using it as a strategic advantage for your business - it may be unwise to spend a lot of money and time

on the technology [3]. Aiming at the above-mentioned problems, an application architecture of intelligent tutoring system based on EGL (Enterprise Generation Language) and Web 2.0 technology is proposed in this paper. Developers can focus on the business issues what code handle without caring for software technical details. The architecture is simple, easy to use and across languages, frameworks and runtime platforms. EGL code can be compiled into COBOL, Java and JavaScript, which means that it can be deployed to so many heterogeneous environments. Figure 1 is the relationship between EGL and Java, COBOL and JavaScript [4].

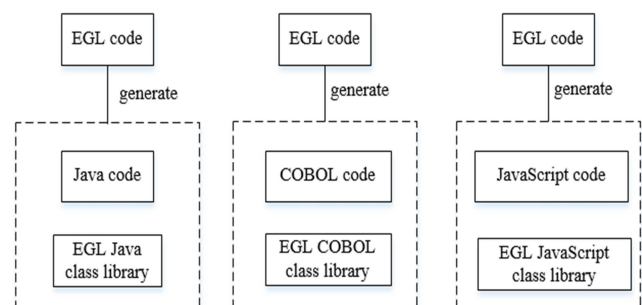


Figure 1. Relationship between EGL and Java, COBOL and JavaScript.

EGL's Web 2.0 development support includes the foreground that the income of the interface development, including support for the background of the service call and business logic to achieve. Front desk interface applications and business logic processing can share the same data structure definition, so the use of a EGL language can complete the front-end to the back-end of the program development, do not need to be proficient in two languages. Effectively reduce the software development cycle and development costs, but also effectively enhance the intelligent tutoring system of human-computer interaction experience and real-time demand.

2. Research Status of Intelligent Tutoring System

Intelligent tutoring system changes the traditional teaching mode and teaching environment, interaction, diversity, flexibility, increase the amount of knowledge, they changed their method of acquiring knowledge. This paper is expected to discuss the principle of the technology of data-mining and its application in the intelligent tutoring system. On the basis of this, the new structure, basic strategy and method are put forward. Intelligent tutoring system was formed in the early 70's of last century, by Jaime Carboell, Allna Collnis and colleagues developed SCHOLAR system, for the region of teaching, it is considered to be the earliest intelligent tutoring system [5]. Because of the development of network technology, to the development of intelligent tutoring system brings new development space. The National Science Foundation will (NFS) research project on its learning and intelligent systems, totaling \$22.5 million dollars in funding. United States of its research work mainly in the army and the University, some famous universities such as MIT, Stanford, California, Johns-Hopkins, Carnegie Mellon, Memphis, are in the research field of intelligent tutoring system made fruitful results, and developed through practice has proved more successful prototype system. University of Memphis on its project, tutor system, the 15 year long study, the problems in students by hints and tips from the strategy to make proper response [6].

At present, many countries are making deeper study of intelligent tutoring system, the main focus is on mining the student model and teaching strategy model for students, the log data to construct a personalized, as well as the application of social and emotional intelligence in the system, so as to further improve the intelligence of the system, make better use of the ITS "excellent teacher" this characteristic. In order to promote students' self-regulated learning: on the one hand, intelligent tutoring system provides a large number of sample resources; on the other hand, intelligent tutoring system creates an interactive problem solving environment for the way of learning by doing. Through the diagnosis and real-time evaluation of the operation process, and the results are analyzed, the error reason of the intelligent tutoring system is put forward, until students reach the goal of problem solving.

Observation, imitation, and then to solve the problem of the process, the students have a good solution to solve the problem of migration. Personalized intelligent tutoring system is the effective way to carry out personalized study [7]. It takes cognitive science as its theories foundation. Meanwhile, the teaching system of the integrated use of educational psychology, educational technology, computer science and other products, which can play the leading role of teachers, reflecting the students' cognitive ability.

3. Basic Architecture of Intelligent Tutoring System

Generally, the basic structure of the intelligent tutoring system is composed of three basic modules, such as the knowledge Database, the student model and teacher model, as shown in Figure 2 [8].

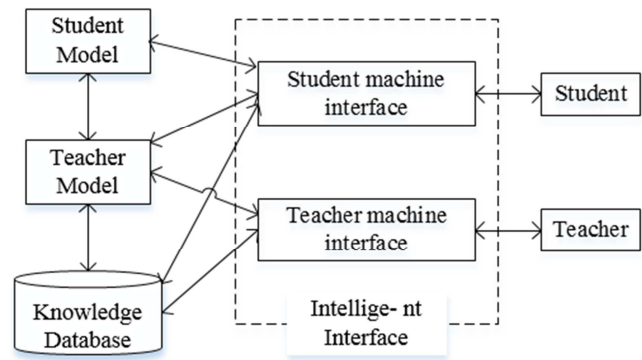


Figure 2. Basic Structure of Intelligent Tutoring System.

3.1. Knowledge Database

The knowledge database consists of two basic parts, namely, the composition of the knowledge system and the composition of the teaching material library [9]. The knowledge system is a collection of knowledge points, including the structure and the relationship between knowledge, a knowledge system is a part of subject knowledge is decomposed into a series of knowledge points, and then gives each knowledge point of an attribute, attribute knowledge includes knowledge point number, importance degree, master degree, prerequisite knowledge requirements the main contents of teaching materials and so on.

3.2. Student Model

In the face of a successful teaching system in many different characters of the learners, to truly teach students in accordance with their own personalized teaching ability, the system must understand the current learning objects, learning cognitive characteristics of various objects in the data structure is known as the student model. The main function of the student model is to record the student ID, name, gender, and other information, at the same time, it is more important to reflect the level of students' knowledge and learning ability, and provide the basis for the intelligent teaching. The student

model is based on the interaction between the students and the system formed, can be modified according to the students' learning dynamic, the system through the student model can be personalized teaching [10].

3.3. Teacher Model

Teacher model can be based on the content of the student model and knowledge of the content of intelligent decision-making and intelligent navigation, collect the students' response, analysis and processing, to judge the students' performance, according to students' different choice of teaching content, to help students analyze the causes of errors, to determine the students' most need to learn knowledge, to provide students with personalized guidance and appropriate supplementary materials.

4. EGL and Web 2.0

Compared with the present stage of the intelligent tutoring system application framework, language used in this study to develop the EGL is a fourth generation programming language, it can let developers attention to code processing business problems, and has the characteristics of cross platform, cross language and cross application. In this research, using EGL, we can develop the Web 2.0 application and Java application running on server side only using EGL language. The application framework is able to make full use of mature technology platform and rather than replace them, effectively reduces the software development cycle and development costs at the same time, is effective to enhance the intelligent tutoring system "experience of human-computer interaction and real-time requirements" [11], [12], [13].

EDT supports the development and deployment of Web 2.0 applications. Terminal user access the Web page (include HTML and JavaScript) generated by EGL code and the browser is responsible for download them to client-side. On the client-side, JavaScript generated by EGL code will interpreter in the browser and demonstrate the corresponding interface. Then JavaScript code generated by corresponding EGL statement is responsible for calling the Web Service or REST Service deployed on the server. Java EE container on the server-side is responsible for receiving the request from client-side and returning it to the browser, and then JavaScript application generated by EGL on the client-side will demonstrate it to the terminal user [14].

As is shown in the compilation process of EGL, EGL itself does not run directly and it will be compiled and executed by the compiler of target language in building and generating target language on corresponding platform [15]. And in the process of language generation, several of different platforms and existing technology can be integrated and made full use of. For example, in the browser-side, EDT made full use of Dojo framework to support the development of Web 2.0 and JavaScript generated by EGL encapsulated the Dojo; in the server-side, Java code generated by EGL encapsulated database access using JDBC. EGL is not to replace existing technology and not to unify to develop new language and it is

to maximize the use of mature technology as well as the supplement and extension of existing technology [16].

EGL language is architected to reflect patterns that are common to different kinds of business software and hides many details that are platform specific. EGL also helps a company retain developers who are knowledgeable in business processes, even if those developers lack the time needed to stay current with technical change. And the relative simplicity of the language helps traditional developers become accustomed to the latest technologies [17].

5. Implementation of Intelligent Tutoring System of Primary Mathematics

This system is the EGL page layout, students will enter the student after landing module, elementary school mathematics intelligent tutoring system of student module as shown in figure 3.



Figure 3. Student model.

5.1. Rich UI Interface

In the Rich UI design of intelligent tutoring system, the basic design idea is to refresh RUI widget as a unit. That is, the whole page is divided into several widgets and each widget varies independently. When retrieving data by calling backend service, EDT need to refresh the front page and the grain size of refreshed widgets should be as small as possible, so that it can reduce the throughput and response time. To create an EGL Rich UI application, a RUI handler is need primarily. The handler holds the EGL logic to add widgets to an initial DOM tree and to respond to events such as a user's click of a button. Primarily, an EGL RUI handler named Student (or Teacher or Manager) is created as a whole to call other handlers that can be designed as the sub modules of the intelligent tutoring system by user's click of buttons. The buttons should be added an event named *showcall* as is shown below:

```
onClick ::=showcall;
```

Then we can configure the code so that the event handler responds to the event that is internal to the code. And the function *showcall* runs as soon as the user clicks the buttons. Such an event might be receipt of a message that was returned from a service. In the Student (or Teacher or Manager), we can write the event handler and call other handlers by using a case

statement. The code is shown below:

```
function showcall(event Event in)
  button DojoButton=event.widget;
  ContentBox.children = [ ];
  case (button.text)
    when("Personal Information")
      ContentBox.appendChild(new StInformation{}.ui);
    when("Online Learning")
      ContentBox.appendChild(new Learn{}.ui);
    when("OnlineTest")
      ContentBox.appendChild(new Test{}.ui);
    when("Job Management")
      ContentBox.appendChild(new StWork{}.ui);
    when("Result inquiry")
      ContentBox.appendChild(new Inquiry{}.ui);
  end
end
end
```

The button widgets integrated the Dojo widgets and were placed into Box and GridLayout widget of the page. The page is divided into several sections by Layout widgets and the Box named *ContentBox* belongs to Layout widget. The sub modules will be the children of *ContentBox* and compose the single application by embedding multiple RUI handlers. However, by saying “embedded handlers” we do not mean to say that we physically embed one handler in another. Instead, one handler – an EGL section of the current user interface - declares a functional variant for accessing the second processor and components.

A reasonable practice is to use embedded handlers for service invocation and for other business processing that lacks a user interface. If the embedded handler has an on-construction function, the function runs when the declaration for the related variable runs. In this paper, the core module of the Student module is online learning, which is included in the handler named *Learn {}*. It encapsulates the references of some RUI widgets such as buttons including events, dataGrid displaying the information of the intelligent tutoring system from database. These events can be executed by calling service from business service layer. Each service calling will construct a callback function instance and retrieve the references of the needed RUI widgets. By the RUI, service calling layer can be responsible for calling the service provided by the Business service layer on the server-side and then return the results to client logic layer and client presentation. That is to say, service calling layer decouple the front logic from the backend logic and use call statement to call the created *MyService* of each service. The access of Backend service becomes simpler and the code becomes easier to maintain. Then Take function *readFromTable* for example to demonstrate the implementation principle of service calling layer. The widgets of client-side include UI controls and each widget can indicate screen events such as “onClick” to call the code of service layer.

```
function readFromTable (event Event in)
```

```
  call MyService.getAllLessons( )
    using dedicatedServiceBinding
    returning to mycallback
    onException serviceExceptionHandler;
end
```

The function *readFromTable* is responsible for retrieving data and displaying the data. As is show in the code of the function, the screen event “onClick” will call the service of *MyService* using *dedicatedServiceBinding*. Resource Binding is one of the outstanding characteristics of EGL language. This simply means it is a description about how to connect to the database and how to invoke the service. We can maintain the binding in the deployment descriptor files of EGL and the binding can be seen the extending of the application logic. When we develop and deploy the application, the deployment descriptor files will provide specific details of connection and calling service. If errors occur in the process of calling service, exception handler *serviceExceptionHandler* will be called. After the screen event “onClick” retrieve the data from the database by service calling layer, and then it will use the function *mycallback* to process it. In this case, the DataGrid widget named *myLesson* is responsible for displaying the data. The code of *mycallback* is shown as below:

```
function mycallback (retResult Lesson[] in)
  myLesson = retResult;
  myLesson_ui.data = myLesson as any [];
end
```

We can access a function or property in an embedded widget by extending the dot syntax. For example, the above-mentioned statement retrieves the displayed data of the DataGrid widget named *myLesson*. In the Learning handler, users can not only retrieve the information of lessons from database, but also add, delete or edit the lessons into the RUI. These events can also be realized by buttons and they can be displayed by function *showDialog* and also be hidid by function *hideDialog*.

```
function showDialog (event Event in)
  Dialogcontent.children = [info,buttonBar];
  dialog.showDialog();
end
function hideDialog (event Event in)
  dialog.hideDialog();
end
```

5.2. Business Service Layer

Services can include new logic and can expose the data returned from other services and from called programs. EGL language offers end-to-end processing: developers can write the user interface, the service logic, and, if necessary, new backend programs.

A RUI application invokes services asynchronously, which means that the user can still interact with the user interface while the RUI application is waiting for the service to respond. However, if the user needs the

information to continue a task, we can disable widgets and present a simple animation until the service responds. The runtime technology ensures that the invocation occurs as soon as a message arrives from the service. The process for invoking a service often requires an EGL interface part, which describes the data that can pass between the application and service. The Interface part tells the names of the service operations and, for a given operation, the kinds of data that the application exchanges with the service [16].

In many cases, before invoking the service, we need declare the variable based on the interface part in the client-side handler. Here is a declaration of *MyService*:

```
MyService SQLDataSource?;
dedicatedServiceBinding HTTPProxy;
```

The service variable declaration specifies the location which identifies the protocol that formats a message at the start of transmission and unformatted the message at the end of transmission. Web Service is a facility to let developers create logic that receives or sends messages over HTTP. The called service can be available by binding the database connection. This information can all be included in a Web Services Description Language (WSDL) file. A service contains public functions that can be accessed from other code and can include private functions and global variables, but those functions and variables are solely for use by functions that are within the service. At run time, the service is stateless, which means that the internal logic never relies on data from a previous invocation.

Any technique for working with a service is a variation on what we have shown here: create a variable, bind it to a service client binding, and access a function by way of the variable. Remote Services are the standard way to communicate with the Web application server from the user's browser. This is done using a standard Ajax, essentially an HTTP POST. The server-side code for that Ajax call is found within the generated JavaScript application. This is code that executes a Web application server and this is where we tie user actions at the browser into business logic and then for data persistence storage.

5.3. Database Access Layer

The basic idea of a relational database is that data is stored in persistent tables. Each table column represents a discrete unit of data and each row represents a collection of such data and is equivalent to a file record. An EGL record can be the basic of a variable used as the source or target of an I/O operation [16]. We can interact with a relational database as follow: define a record whose stereotype is *SQLRecord*, create a variable based on that record and use the variable as an I/O object in different data-access statements. Usually, one or more columns in a database table can be primary keys, which mean that the values in those columns are unique to a given row. For example, here is a Record about lessons of the intelligent tutoring system:

```
record Lesson type Entity {
  @table{name = "USER.LESSON"}}
  lesson_id int { @id, @GeneratedValue,
  @Column{name="lesson_id"}};
  lessonname string(100) {
  @Column{name="lessonname"}};
  lessonintro string(255){
  @Column{name="lessoninfo"}};
end
```

The *lesson_id* column is an identity column, which means that the database will place a unique value into that column whenever the user creates a record. Each value is one more than the last. To generate the Java code that is appropriate for the SQL operation, the EDT Java generator uses the Table, ID, Generated Value, and Column annotations.

Access to relational database is by way of SQL. To retrieve a row, we can assign a value to the record field associated with the key column and then issue a get SQL statement to read data from the table in database. The code of function *getAllLessons* using get SQL statement is shown below:

```
function getAllLessons () returns (Lesson [])
  Lessons Lesson [];
  try get lessons from MyService with #sql {select
  lesson_id, lessonname, lessonintro from LESSON};
  onException (ex SQLException)
  end
  return (Lessons);
end
```

The following code declares a related record variable in the client-side handler. And then we can use the variable to access the database [18].

```
myLesson Lesson [ ];
```

In addition, before we generate EGL code, we should configure a build descriptor, which is a build part that guides the generation process and references other definitions. The resource associations can associate the logical file name with a physical file on each target platform where we intend to run the code.

6. Conclusion

In this paper, it is simulated that the client-side submits request to the Web server-side in the EDT development environment to propose application architecture of intelligent tutoring system based on EGL and Web 2.0. The testing scenario is information query in the handler of client-side, users can create and connect the database connection and retrieve the data in database through access the service. The aggregate average response time (AART) of the system is shown as Figure 4.

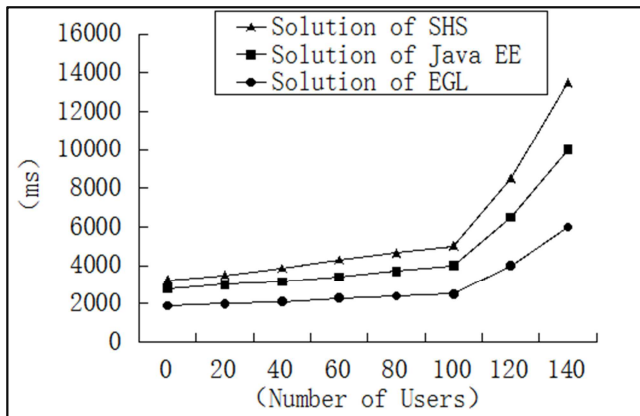


Figure 4. AART curve.

Analysis of AART curve in Figure 4, compared with the intelligent tutoring system solution based on open source framework—Struts, Hibernate and Spring (SHS) [19], the system response time of the solution of Java EE increases in accordance with the linear way until it reaches about 140 users [20]. After that time, the increase of the curve becomes more intense. By analyzing the change trend of the two curves, we can draw the conclusion: the maximum number of users of the solution of Java EE presents is 100 and the application is suitable for intelligent tutoring system. However, the solution can significantly reduce the cost and complexity of intelligent tutoring system and improve the intelligent tutoring system interactive experience and real time requirements. Because EGL language can be executed after generating Java code, the changes of the curve of the solution of EGL are similar to Java EE's and only its speed is slightly slow. It implements the loose coupling between business logic and access control, that is, choices related to a user interface, and choices related to the data in persistent, are handled separately. EGL is general because the technologies are so varied, but we will follow up by nothing how the separation applies to EGL Rich UI. In the development of EGL Rich UI application, the MVC pattern is often used to realize above-mentioned features. Therefore, the primary benefit of the application architecture of intelligent tutoring system based on EGL is simplicity.

In this paper, an application architecture of intelligent tutoring system of primary mathematics based on EGL (Enterprise Generation Language) and integrating Web 2.0 technology is realized. Compared with the solution of intelligent tutoring system based on Java technology, the EGL of cross platform and cross application characteristics, use only the EGL development application in the browser application of Web 2.0. and the server side of the Java application running browser applications, and do not need to master the two languages, JavaScript and Java. In addition, in the development process also can make full use of mature existing platforms and technologies and software resources, rather than replace them, so as to achieve the business of key functional code of the developer concerned about the program, and has not been affected by the specific software technology issues. In view of the problems existed in the

current teaching system and modern distance education teaching environment of intelligent tutoring system demand, analyzes the research direction and development trend of the system, is proposed and implemented a generic, building the future of elementary school mathematics in an intelligent tutoring system has guiding significance, has a very broad application-prospects.

References

- [1] J. Reva Freedman, What is an Intelligent Tutoring System? Published in *Intelligence* 11(3), 2000, pp. 15-16.
- [2] J. Nimisha, Goel, Seema, Maitrey, Sumita and Kanauzuya, Web Technologies (Web 2.0), *International Journal of Computer Applications*, 1(6), 2010, pp. 11-12.
- [3] J. Bo Song and Jie Liu, Implementation of J2EE Data Persistence Tier with Top Link, *Microelectronics & Computer*, 23(8), 2006, pp. 132-135.
- [4] J. Huafa Jin and Zhanghong Chen, *Introduce to Enterprise Generation Language*, Programmer, 2009, pp. 91.
- [5] J. Hyacinth, S., Nwana, User modelling and user adapted interaction in an intelligent tutoring system, *User Modeling and User-Adapted Interaction*, 1991, pp. 1-32.
- [6] D. Zhang Xiaoqing, Research and design of data mining in Intelligent Tutoring System, Xiangtan University, 2011, pp. 3-4.
- [7] J. Niu Xiaoyan, Design and implementation of intelligent system based on Agent, *Agriculture Network Information*, 2016, pp. 58-59.
- [8] J. M, Elsom-Cook, Student Modelling in Intelligent Tutoring Systems, *Artificial Intelligence Review*, 1993, pp. 25-27.
- [9] J. Joseph, Psotka, Sharon, Mutter, *Intelligent Tutoring Systems, Lessons Learned.*, Lawrence Erlbaum Associates, 1988, pp. 58.
- [10] J. Shute, Psotka, *Intelligent Tutoring Systems, Past, Present, and Future*, Human resources directorate manpower and personnel research division, 1994, pp. 2-52.
- [11] J. Bo Song and Jie Gao, The Enhance and Application of Collaborative Filtering in e-Learning System, *Lecture Notes in Computer Science*, 2015, pp. 245-247.
- [12] J. Bo Song and Jie Gao, An e-Learning System Based on EGL and Web 2.0, *International Journal of Computer and Engineering* Vol. 7, No. 2, April 2015, pp. 162-163.
- [13] J. Jie Gao and Bo Song, A Personalized e-Learning System Based on EGL, *International Journal of Machine Learning and Computing*, Vol. 4, No. 5, October 2014, pp. 411-412.
- [14] J. Nimisha Goel and Seema Maitrey, Sumita, Kanauzuya., Web Technologies (Web 2.0), *International Journal of Computer Applications* 1(6), 2010, pp. 11-12.
- [15] J. Matthew A Russell, *DOJO: The Definitive Guide*, O'Reilly Media, USA, 2009.
- [16] J. B Margolis and Danny Allan, *Enterprise Web 2.0 with EGL*. MC Press, USA, 2009, pp. 83-95, pp. 131-141, pp. 225-237.

- [17] J. Hayden Lindsey and Ben Margolis, IBM Rational Business Developer with EGL, MC Press, USA, 2008, pp. 187-189.
- [18] M. B Margolis and Danny, Enterprise Web 2.0 with EGL, MC Press, USA, 2009, pp. 83-95, pp. 131-141, pp. 225-237.
- [19] D. Bo Song and Jing Zhao, Research on Network Teaching System Based on Open Source Framework, IEEE Ninth International Conference on Hybrid Intelligent Systems, 2009, pp. 28-32.
- [20] J. Bo Song and Y Zhang, Implementation on Network Teaching System Based on Java EE Architecture, IEEE Second International Conference on Information Technology and Computer Science, 2010, pp. 354-357.