

An Intelligent Tutoring System Based on EGL

Yan Xue^a, and Song Bo^b

Software College, Shenyang Normal University, Shenyang 110034, China

^ayanxue19900912@126.com, ^bsongbo63@aliyun.com

Keywords: EGL; Web 2.0; Intelligent Tutoring System; Architecture.

Abstract. With the development of Internet technology, intelligent tutoring system is the certain developing trend the human social digitized education. In this paper, aiming at the existing problems of intelligent tutoring system application architecture, in the analysis of the relationship between EGL and Web 2.0 technologies, an application architecture of intelligent tutoring system based on EGL and integrating Web 2.0 technology is proposed. Through the process of design and implementation of an intelligent tutoring system, 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. A design model for intelligent tutoring system was proposed.

Introduction

In general, the intelligent tutoring system (ITS) refers to the use of artificial intelligence technology that allows a computer to play the role of teachers implement individualized instruction to learners with different needs and different characteristics of the transfer of knowledge, and to provide a guide adaptive learning system [1]. Now with the popularity of the campus computer network, exploiting computer Network resources to build intelligent individualized teaching environment has become a key problem of modern information technology. Intelligent Tutoring System is a comprehensive topic involved artificial intelligence, computer science, pedagogy, psychology and behavioral science, and realize better teaching effects by giving intelligence to computer system. Its changes traditional teaching environment and pattern. It replace the way of traditional teaching pattern, and also can be better.

Intelligent tutoring system formed in the early 70s of the last century by JaimeCarbnoell, AllnaCollnis and colleagues to jointly developed SCHOLAR system for teaching in South America, it is considered to be the earliest intelligent tutoring systems [2]. Since the development of network technology, the development of intelligent tutoring system to bring new development. At present, many countries are conducting in – depth research of intelligent tutoring system, the main focus is the student model and teaching strategies model. Compared with the present stage of the intelligent tutoring system application framework, language used in this study to develop of the EGL. Using the features of a cross platform and across application of EGL, the architecture can develop Web 2.0 application applied to browser-side and Java application running on server-side only with EGL. To make developers do not need to master the two language – JavaScript and Java at the same time. The

key feature of the architecture – developers can focus on the business issues what code handle without caring for software technical details – is implemented by using existing platform and technology instead of replacing them and improve the real-time requirements and human- computer interaction experience of intelligent tutoring system effectively.

EGL and Web 2.0

EGL (Enterprise Generation Language) is a high-level language that lets developers create business software without requiring that they have a detailed knowledge of runtime technologies or that they be familiar with object-oriented programming [3]. EGL originally developed by IBM and now available as the EDT (EGL Development Tools). In the EDT environment, EGL is used in a development process that has defined steps, from coding a source to generating an output (Java, or JavaScript) to preparing and deploying that output. The first stage is to compile the EGL code and EGL compiler will scan and analyze the syntax and semantic of EGL source file, then generate the IR (intermediate representation). The second stage is code generation. The code generator read and parses the IR (eglxml file) and then generates corresponding target language. The third stage is to use target language compiler to compile into executable code. The Java compiler will compile the generated Java source files into class files to run in different platforms and the generated JavaScript code will be interpreted in the browser environment [4]. Open Source project under the Eclipse Public License (EPL), is a programming technology designed to meet the challenges of modern, multi-platform application development by providing a common language and programming model across languages, frameworks, and runtime platforms. The language borrows concepts familiar to anyone using statically typed languages like Java, COBOL, C, etc. However, it borrows the concept of stereotype from Unified Modeling Language (UML) that is not typically found in statically typed programming languages.

EDT support the development of Web 2.0 application and its deployment. 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 [5]. 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 [6].

Structure of intelligent tutoring system

Generally, the basic structure of the intelligent tutoring system is composed of three basic modules, such as knowledge base, student model and teacher model, as shown in Fig. 1.

Knowledge base. The knowledge base consists of two basic parts, namely, the composition of the knowledge system and the teaching material storehouse.. 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 [7, 8].

Student model. The student model can be thought of as an overlay on the knowledge base. It is considered as the core component of an ITS paying special attention to student's cognitive and affective states and their evolution as the learning process advances [9]. As the student works step-by-step through their problem solving process the system engages in a process called model tracing. Anytime the student model deviates from the domain model the system identifies, or flags, that an error has occurred.

Teacher model. The tutor model accepts information from the knowledge base and student models and makes choices about tutoring strategies and actions. At any point in the problem-solving process the learner may request guidance on what to do next, relative to their current location in the model [9]. In addition, the system recognizes when the learner has deviated from the production rules of the model and provides timely feedback for the learner, resulting in a shorter period of time to reach proficiency with the targeted skills. The tutor model may contain several hundred production rules that can be said to exist in one of two states, learned or unlearned. Every time a student successfully applies a rule to a problem, the system updates a probability estimate that the student has learned the rule.

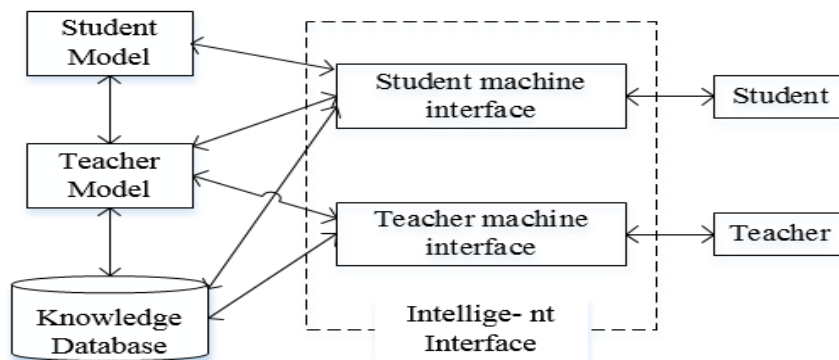


Fig. 1 Basic structure of intelligent tutoring system

Implementation of the intelligent teaching system

Rich UI. 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 [10]. 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 e-Learning 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 [11]. 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);
  ...
  end
end
end

```

Record. An EGL Record part defines a set of data elements. Create a sequence of fields that are processed as a unit [12]. A Record part defines such as a sequence. Intelligent tutoring system requires a login screen, and now we need to create a *user* record. The code is shown below:

```

record user
  username string;
  password string;
end

```

The part has a username and password-in this case, it is user's a list of field. Each field is a declaration that includes a name, data type, and semicolon. The part is a model of format and is used for declaring a record, which is a variable that's based on the Record part.

RUIHandler. The main component of a Rich UI application is a Rich UI handler part. These parts are generated into JavaScript. The code is shown below:

```

package client.client;
import org.eclipse.edt.rui.widgets.Box;
import org.eclipse.edt.rui.widgets.GridLayout;
...
import eglx.ui.rui.RUIHandler;
handler Student type RUIhandler{initialUI =[ui, gridlayout1], onConstructionFunction = start,
cssFile = "css/NewProject.css", title = ""}
  ui GridLayout{columns = 1, rows = 1, cellPadding = 4, children = [ BodyBox ],width =
"1000"};
  BodyBox Box{ layoutData = new GridLayoutData{ row = 1, column = 1 }, padding=8,
  children = [ Image ] };
  Image Image{
  src = "E:\\workspace\\pictures\\3.jpg";
  ...
  function start()
  end
  ...
end
end

```

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 [3].

Conclusions

The application architecture proposed in this paper is based improves the development efficiency of intelligent tutoring system. The architecture is simple, easy to use and across languages, frameworks and runtime platforms, it not only can avoid the repeated writing the similar logic of each domain module, also is conducive to the robustness, maintainability of the system, and flexibility to the changes of whole business needs. It will have an important guiding significance for the application and development of the intelligent tutoring system in the network education.

References

- [1] R. Freedman, What is an Intelligent Tutoring System?, Published in *Intelligence* 11(3), 2000, pp.15-16.
- [2] X. Q.Zhang, Research and design of data mining in Intelligent Tutoring System, Xiangtan Normal University, 2011.
- [3] B. Song, J.Gao, An e-Learning System Based on EGL and Web 2.0, *International Journal of Computer Theory and Engineering*, October 2014.
- [4] B. Song, J. Gao, The Enhance and Application of Collaborative Filtering in e-Learning System, *Lecture Notes in Computer Science* 8795, April 2015.
- [5] Nimisha, Goel, Seema, Maitrey, Sumita and Kanauzuya, Web Technologies (Web2.0), *International Journal of Computer Applications*, 1(6), pp. 11-12, 2010.
- [6] J. Gao, B. Song, A Personalized e-Learning System Based on EGL, *International Journal of Machine Learning and Computing*, Vol. 4, No. 5, October 2014.
- [7] H. S. Nwana, User modelling and user adapted interaction in an intelligent tutoring system, *User Modeling and User-Adapted Interaction* 1991, Volume 1, Issue 1, pp. 1-32.
- [8] M. E. Cook, Student Modelling in Intelligent Tutoring Systems, *Artificial Intelligence Review*, 1993.
- [9] Information on http://en.wikipedia.org/wiki/Intelligent_tutoring_system
- [10] IBM Corporation, EGL Language Reference Version7 Release 01, USA, 2007.
- [11] IBM Corporation, EGL Programmer's Guide Version7 Release 00, USA, 2007.
- [12] B. Margolis and Danny, Enterprise Web 2.0 with EGL, MC Press: USA, pp. 83-95, pp. 131-141, pp. 225-237, 2009.