Journal of Southeast University (English Edition) Dec. 2001 Vol.17 No.2 ISSN 1003—7985

Synchronous Multi-Objects Individual Marking Model*

Shen Jun** Gu Guanqun

(Department of Computer Science and Engineering, Southeast University, Nanjing 210096, China)

Abstract: For individual marking in coming distance education, a novel model that is based on component object model and agent is proposed. It supports multi-objects-oriented online synchronous individual marking on operating test. Some related implementation technologies are also discussed.

Key words: agent, component object model, distance education, individual teaching and marking

With the rapid development of computer technology and network technology, the distance education is an attractive hot issue of research. The distance education can break the limitation of time and space, and bring huge market potential and social benefits. It will breathe new energy into the reform and the popularization of higher education and establish the foundations of lifelong education. As the evaluation way of teaching quality, marking is an efficient method. On the traditional standard marking, there are many skilled marking technologies. But, on the operational test, there are many key technologies that must be solved[1].

In this paper, an agent-based online synchronous distance marking architecture is presented. It supports distributed marking and individual marking. It can execute online mark on many different testees that are tested on different items and output corresponding report cards. Based on the component object model, we can track the behaviors of every testee and get their operating results of every step and execute corresponding marking. By the cooperating of multi-agents, the marking status of every testee is transmitted to server in order to do statistic, generate report and save related data to database. At the same time, on different testee, according to the accepted marking status, the server agent selects next test item from database, transmits the test item to corresponding client agent. Then the client agent continues the next marking procedure.

1 Architecture of Marking System

The architecture of marking system adopts prevalent 3-tiers structure: representation, business logic and data server. The representation layer bases on dynamic Web, and the business logic layer bases on COM +, the data server layer supports prevalent many kinds of RDBMS and other data sources, for example, SQL Server, ORACLE, File System, etc. It is presented as Fig.1[2].

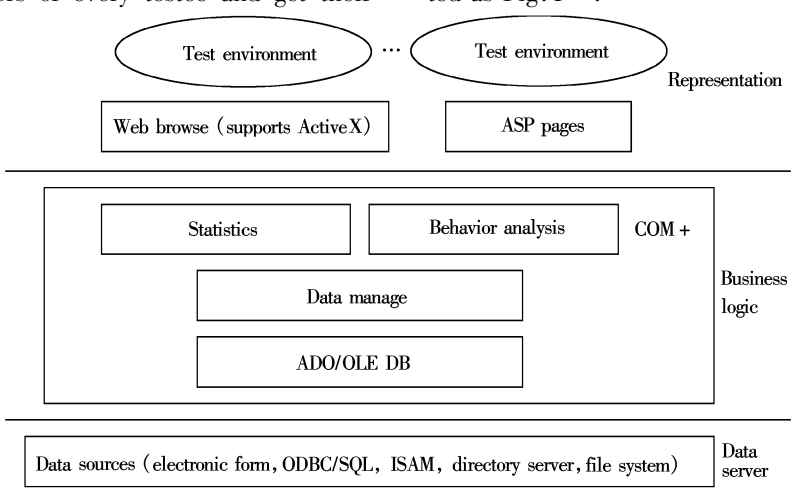

Fig.1 The architecture of marking system

Received 2001 - 04 - 29.

* The project supported by the project of ministry of China about key fundamental research on application (6209003001).

** Born in 1963, male, associate professor.

The Web browse that supports ActiveX controls and ASP pages constitutes the representation layer, which includes two agents that are downloaded from server after testee logins and is accepted. One is charge of communication with the agent in business logic layer in server end, and the other is charge of interaction with testee and executing marking. The task of the former includes marshaling, de-marshaling, and interaction with the latter. The task of the latter includes the representation of test items that may be multimedia data such as texts, graphics, audio, video and so on, the tracking of the behaviors of testee, the marking, and interaction with the former.

In business logic layer, four agents that conform the specifications of COM + cooperate each other in order to complete the tasks such as interaction with client ends, statistic, data server, testee's behavior analysis, and the generation of next test item, etc. The co-relations of four agents are presented in Fig.2.

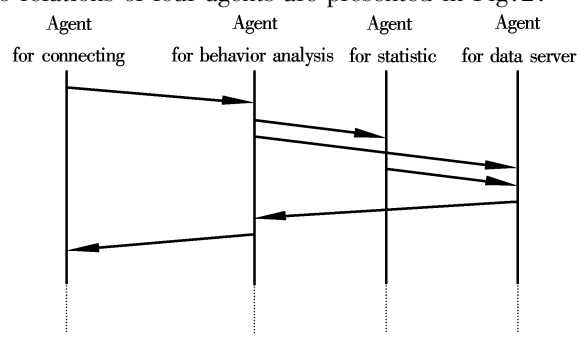

Fig.2 The interaction of agents

2 Protocol between Server Agent and Client Agent

The application protocol for marking includes two parts that respectively satisfy the up chain and down chain. In the up chain, it consists of three sections: the ID of testee, the ID of test items, and the information about operating status and marking results. On the contrary, it includes the ID, the type, the marking rules and the contents of next test item, etc. Both two kinds of application protocols are described with XML[3,4]. The related elements and attributes are presented as follows.

```
〈! -- for up chain -- 〉
〈?xml version = "1.0"?〉
〈!DOCTYPE Marklist
  SYSTEM "http://www.seu.edu.cn/Marklist.dtd"〉
〈Marklist〉
    〈Testee〉
      〈TID〉…〈/TID〉
      〈TName〉…〈/TName〉
      〈TSite〉…〈/TSite〉
  〈/Testee〉
  〈TestItem 〉
      〈ItemID〉…〈/ItemID〉
      〈ItemType〉…〈/ItemType〉
  〈/TestItem 〉
  〈MarkState 〉
    〈Answer〉…〈/Answer〉
    〈Marking〉…〈/Marking〉
  〈/MarkState 〉
〈/Marklist〉
  〈! -- for down chain -- 〉
  〈?xml version = "1.0"?〉
〈!DOCTYPE Marklist
  SYSTEM "http://www.seu.edu.cn/Marklist.dtd"〉
  〈Marklist〉
    〈Testee〉
        〈TID〉…〈/TID〉
        〈TName〉…〈/TName〉
        〈TSite〉…〈/TSite〉
    〈/Testee〉
    〈TestItem 〉
        〈ItemID〉…〈/ItemID〉
        〈ItemType〉…〈/ItemType〉
        〈Item〉…〈/Item〉
  〈/TestItem 〉
  〈MarkRule 〉
      〈StdAnswer〉…〈/StdAnswer〉
      〈Marks〉…〈/Marks〉
  〈/MarkRule 〉
〈/Marklist〉
```

3 Some Related Implementing Methods

3.1 Invoking and releasing of the test environment

With the evolution of software development model, all software products are based on the component object model. Thus, according to the needs of the test, the invoking and releasing of test environment really mean the invoking and releasing of related component objects. For example, in the test of the course about the application foundation of computer, we will invoke and release the Microsoft Word, Microsoft Excel and others. But, based on the different specification of component object model, i.e. COM +, CORBA or EJB, we must take different methods in the implementation.

On the COM + specification, the invoking and releasing of a component object means object automation. The client program, an agent, invokes the test environment by its automation interface, and releases the object after test. For example, the test environment, Microsoft Word, is invoked and released according to the steps as follows (described with Visual BASIC)[5].

```
Set MyWord = GetObject(, "Word.Application")
```

```
Myword.Visible = True
Myword.Documents.Open "C:\test\test.doc"
    …          //( the operating of testee )
    …
    …          // (the obtaining and marking of operating results)
ActiveDocument.SaveAs FileName: = "c:\test\Text1.rtf",
FileFormat: = wdFormatRTF
Myword.Quit
Set Myword = Nothing
```

When it is described with Visual C++, the steps are presented as follows:

```
CLSID clsid;
IclassFactory * pClf;
IUnknown    * pUnknown;
CoInitialize(NULL)          //initial COM
CLSIDFromProgID("Word.Application", &clsid);
                    //Get the component's clsid from the progid
CoGetClassObject (clsid, ,NULL,IID_IclassFactory,(void **)
&pClf);
pClf -> CreateInstance(NULL,IID_IUnknown,(void **)&Un-
known);
    …          // ( the operating of testee )
    …
    …          // (the obtaining and marking of operating results)
pClf -> Release();
pUnknown -> Release();
CoUninitialize()
```

3.2 Obtaining and processing of the operating results

After the test environment is invoked, client agent will represent the current test item on the screen according to the accepted data package and wait the completion of operating of testee. If the current test item is multimedia data, client agent invokes related media player in the client machine or works on the predicted method. Then the agent obtains the results of current test step from the test environment by interaction with the test environment object, and executes online marking according to the predicted marking rules. At last, the results and status of marking, such as testee's operating trails, are marshaled by application protocol and sent to server agent.

The results of operating depend on the corresponding attribute values of test environment object. Every test item has its needs to attributes of test environment object. For example, in the test of the course about the application foundation of computer, on the format of font, the following attributes of test environment object, Microsoft word, are concerned.

```
Font.Name, Font.Color,
Font.Bold, Font.BoldBi, Font.BoldRun, Font.Italic,
Font.Undelline, Font.FormattedText,
etc.
```

3.3 Establishment of the marking rule

The establishment of the marking rule is a key. In order to enhance convenience of operating, modern software often has many kinds of operating method on an operating step. Therefore, the rule of marking must include all kinds of operating methods of an operating step, and assign different power value to every operating method according to the standard of test. On every test item, there are some defined rules. The establishment of the marking rule must consider the demands of tester. The model of marking is usually the function of operating method, operating sequences, error frequency, power value and demands of test, etc. It is defined by $f(a_1, a_2, a_3, a_4, \cdots)$ that represents the marking rules. For instance, it is simply defined as follows in current version.

$$f = (a_1 \times p_1 + a_2 \times p_2) \times (1 - a_3/M)$$

where a_1 indicates the selected method; p_1 is related power value; and a_2 indicates the selected operating sequences; p_2 is related power value; M is the maximum allowable value of illegal operation related to current step; and a_3 is the practical value of the illegal operation of current step.

According to the different demands of test, many kinds of marking models will be established.

4 Prototype Implementation

Based on the idea described above, a prototype system, MOIMS, is implemented. In order to test the system, we construct a test environment in laboratory. Five kinds of operating test items were used as test pattern and ten test items were included in a pattern. The result of test is satisfied. It efficiently implements the synchronous individual marking. But, there are some weaknesses as follows: 1) The syntax of application protocol is not enough to satisfy all kinds of operating items; 2) On the client end, we don't consider the multi-version problem of test environment; 3) With the connect number increase, the load of server must be greatly increased. It will delay the response time. So multi-thread technology must be adopted in server end.

5 Conclusion

On the online mark of interactive operating, this paper presents a new design idea, and discusses some efficient implementation methods. The idea and its implementation are not only adapted to online indi-

vidual marking, but also adapted to online individual teaching. It breaks the bottleneck of individual marking and individual teaching of interactive operating aspect. It inevitably promotes the development of network education, and pours new energy into distance open teaching.

References

1 J. Shen, G.J. Gu, Research on architecture of interactive network teaching (in Chinese), *CSIT*'2000, Nanjing, pp. 137 – 139, Mar.2000

2 G. Eddon, and H. Eddon, *Inside COM + base services*, Microsoft Press, America, 1999

3 Michael Morrison, *XML unleashed*, Sams Publishing, America, 2000

4 W3C, Extensible markup language (XML) 1.0 specification. *W3C Recommendation*, February 10, 1998, http://www.w3.org/TR/REC-XML

5 A. Rofail, and Y. Shohoud, *Mastering COM and COM +*, SYBEX Inc., America, 1999

同步多主体个性化评测模型

沈 军 顾冠群

(东南大学计算机科学与工程系, 南京 210096)

摘 要 针对远程教育中的个性化评测,提出了一种新型的评测模型,该模型建立在组件对象模型和 Agent 基础上,支持多主体的操作型试题在线同步个性评测.同时论述了相关的实现技术.

关键词 Agent,组件对象模型,远程教育,个性化教学及评测

中图分类号 TP18