Administration and Assessment in Virtual University ; Case Study of Database Practice

Tomoyuki Tanaka, Katsuya Tanaka, and Makoto Takizawa

Tokyo Denki University
E-mail {tana, katsu, taki}@takilab.k.dendai.ac.jp

There are three aspects on distance learning; awareness, assessment, and administration. Many papers discuss various topics of awareness like web-based training and video on demand. This paper discusses assessment and administration aspects of a distance learning. We present a system which supports automatic question answer of database practice as an example of the aspects.

Virtual Universityにおける データベース演習の遠隔教育システムの研究

田中 智之　田中 勝也　滝沢 誠

東京電機大学理工学部情報システム工学科
E-mail {tana, katsu, taki}@takilab.k.dendai.ac.jp

現在、多数の大学や企業により、遠隔教育システムの実現を目的とした研究開発が盛んに行われている。この遠隔教育システムを考えるときの重要な3つの視点として、awareness、assessment、administrationがある。これまでの遠隔教育システムの研究では、awarenessに着目し遠隔教育システムにおけるコンテンツの配布や意見交換の場をいかに提供するかが議論されている。一方、本論文では、遠隔教育のassessment、administrationに注目する。東京電機大学で行なわれているデータベース演習について、演習レポートを提出する学生の評価 (assessment)、結果の管理 (administration) を自動で行なうシステムの実装方式を述べる。

1 Introduction

According to the advance of network technologies, various kinds of computers, even consumer electricities in the world are interconnected in the Internet. In addition, mobile phones like IMT2000 and terminals like PDA are widely used. By using the Internet and mobile communication, users can access to every computer in the Internet anytime and anywhere. Furthermore, communication speed is getting faster like JGN [6]. By using the high-speed networks and data compression technologies like MPEG-4 [3], multimedia data like video and voice can be transmitted in continuous and realtime manner. At ICOIN-15 [1], a virtual live session was held between Beppu, CRL, Tokyo, London, and Madrid by using the JGN and high-speed network.

Distance learning is now one of the most important applications at universities, schools, and companies, which use high-speed Internet. There are various approaches to realizing the distance learning like web-based training (WBT) [4], [5]. Many universities are now starting to adopt the distance learning technologies [2].

There are views of the distance learning as pointed out by Macro project [2] ; *awareness*, *assessment*, and *administration* (*3A*). WBT is one of the awareness issues. There have been many discussions on the awareness issues [2], which are using the internet and multimedia technologies. In addition to the awareness issue, we need discussions on assessment and administration issues to realize the distance learning. The assessment issue means how to evaluate students, teachers, and courses. The administration issue shows how to administrate students, teachers, and courses. In this paper, we discuss assessment and administration aspects of the virtual university through a case study of a course database practice of Tokyo Denki University.

In section 2, we discuss difficulties in traditional courses. In section 3, we present a learning model. In section 4, we discuss design and implementation of our system.

2 Difficulties in Traditional Courses

In traditional universities, students enroll courses where professors give lectures to them. Students submit reports and take some number of examinations. The students finally get scores of the courses. This is a typical course.

There are the following difficulties in the traditional courses :

1. Students make copies of reports made by another student without doing their homework by themselves.
2. The students take the same examinations at the same time. The courses are synchronously succeeded. Every student listens to the same lecture and takes the same examination even if some students understand well and other students do not understand the lectures.
3. It is difficult to score reports submitted by many students.

In this paper, we discuss how to resolve these difficulties by using the internet and database systems technologies. Since it is difficult, possibly impossible to find a solution for general cases, we first take an example, say a course of database practice which sophomores and juniors take at our department of Tokyo Denki University.

3 Learning Model

We take as an example a *database practice* course where students study how to use SQL on a relational database system. In the course, a question is given to a student. Each question is written in Japanese and shows what to be derived from the database, e.g. find names of stations in Tokaido Shinkansen. The database includes tables showing data on JR trains, e.g. stations, lines, properties of stations like Ekiben, distance, and fare. The schema of the JR database is shown in Appendix.

The student makes up a SQL statement for a question and issues it to the JR database which is a relational database system. If the student could get a correct snapshot, the student submits the SQL statement and the snapshot to the instructor. The instructor checks if the SQL statement and the result are correct. If they are not correct, the instructor sends back an error report and then the student tries to get a correct SQL again.

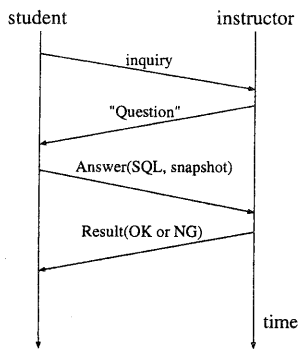

Figure 1: Interaction

There are various types of queries on the JR database. We classify SQLs to the following five classes :

1. Restriction.
2. Projection.
3. Join.
4. Group by.
5. View.

The first class means simple queries which use just restrictions on one table. The following is an example of question and SQL.

Question : find station names of JR lines in Tokyo.

SQL :
```
select name from station
where pref="Tokyo"
```

The second class shows queries which use projections on a table.

Question : find prefecture names and area of prefecture in Japan.

SQL :
```
select pref.name,pref.area
from pref
```

The third class indicates queries which use joins on multiple tables. The following is an example of query and its SQL.

Question : find station names of Yamanote line.

SQL :
```
select station.name
from station,stline,line
where station.stid=stlline.stid
and stline.lnid=line.lnid
and line.name="Yamanote-Line"
```

The fourth class shows queries which use joins of multiple tables and group-by. An example is shows as follows.

Question : find names of stations where the most number of Ekibens are sold in the JR.

SQL :
```
select station.name
from station,stlunch,lunch
where station.stid=stlunch.stid
and stlunch.luid=lunch.luid
group by station.stid
having count(lunch.luid)=
max(count(lunch.luid))
```

In order to prevent students from copying answers made by the other students, each student is given a different question from the others. For each question class, there are more number of questions than students. The system randomly takes one question in the class of the questions, which is not yet given to any student. Questions are stored in the metadatabase (MDB). A state of each question is also recorded, i.e. when and to whom the question is issued.

On receipt of a question, a student starts to write a SQL statement to derive data required by the question from the JR database. The student tries to get answers by issuing SQL statements to the JR database. Then, the student submits a pair of SQL statement and the result, i.e. snapshot derived from the database.

It is not easy for instructors to check if SQL is correct as well as the snapshot. The system automatically checks if the answers submitted by students are correct.

After a student submits some number of correct answers, the student moves to a one-level higher class. Figure 2 shows a state transition diagram where a circle shows a state indicating a question class i. At a state i, a student takes a question (Q) and then submits an answer. If the answer is correct (A), a counter c is incremented. If the answer is not correct (N), c is decremented. If the value of c gets larger than the predefined constant p, a state is transited to a state $i+1$, i.e. question class $i+1$. Here, $c = 0$. The more number of wrong answers the student makes, the larger value of the counter c the student loses. If the student gets smaller value of c than m, the student backs to the lower class i -1. The student studies for the question class i -1 again.

4 Implementation

4.1 System configuration

The system is implemented in a three-tier client-server *architecture* as shown in Figure 3. The system is composed of client, application server, and data server. The data server is installed by Sybase in the Sun Enterprise 450 Server. Application servers are distributed on Linux PC machines. Each student can use its own note PC

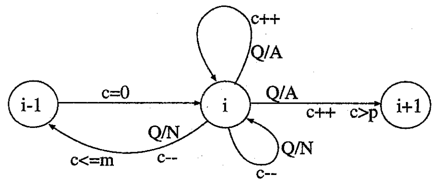

Figure 2: Transitions among classes

as a client. The servers are interconnected with 100base-T local area network.

There are two databases in the data server, JR database and metadatabase (MDB) in the data server. In the JR database, data on JR are stored. The schema of the JR database is shown in Appendix. In the metadatabase (MDB), the following tables are stored :

1. Questions table.
2. Answers table.
3. Students table.
4. Classes table.

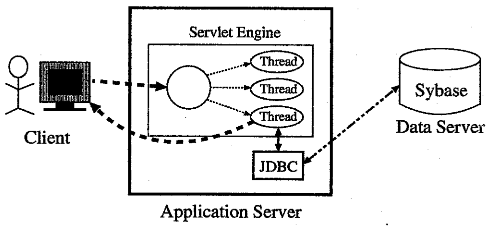

Figure 3: System configuration

The Questions table maintains questions which are written in Japanese. The schema of the Questions table is defined by the following SQL :

```
CREATE TABLE Questions(
  qno INT,
  lno INT NULL,
  query TEXT NULL,
  sql TEXT NULL,
  tname VARCHAR(7) NULL,
  atcount INT NULL,
  dis BIT,
  qname VARCHAR(50) NULL,
  qdate SMALLDATETIME NULL,
  rname VARCHAR(50) NULL,
  rdate SMALLDATETIME NULL,
  ord INT NULL)
```

Each question is identified by identifier *qno*. A question in Japanese is stored in attribute *query*. *tname* shows a name of table which is a correct snapshot of the question.

The Answers table stores a set of schemas of the snapshots for the questions. Snapshots derived by students are compared with the snapshot stored denoted by the Answer table. The schema is defined as follows :

```
CREATE TABLE Answers(
  qno INT,
  attno INT NULL,
  attname VARCHAR(8) NULL)
```

The Students table maintains data on each student, e.g. *sname*, identify *sno*, and record in the course, e.g. level of class. The schema is defined as following SQL :

```
CREATE TABLE Students(
  sno CHAR(7),
  sname VARCHAR(50),
  pass VARCHER(10) NULL,
  cqno INT NULL,
  clno INT NULL,
  lcn INT NULL,
  len INT NULL,
  tcn INT NULL,
  ten INT NULL,
  ustate INT NULL,
  flg INT NULL,
  que INT NULL)
```

The Classes table shows at which class each student is at present studying.

```
CREATE TABLE Classes(
  sno CHAR(7) NULL,
  luno INT NULL,
  ludate SMALLDATETIME NULL,
  deadline SMALLDATETIME NULL)
```

The system is implemented by using Java in an application server which is realized by Java servlet. Students use the system through the web in a client computer.

4.2 Overview of QA

Figure 4 shows an overview of the system behaviour. The system behaves as follows :

1. A student logs in the system with user identifier and password. The user identifier and password are searched for the Students table. If the student is an authenticated one, the student can start using the system.
2. A student sends a message "inquiry" to obtain a new question.
3. The class at which the student is studying is maintained in the Students table. The system randomly takes one question for the class which is not issued to any student. The question is marked "used." The system sends the question to the student.
4. The student receives the question. Then, the student tries to obtain SQL. The student issues SQL to the JR database. If the student thinks that the student makes up SQL, the student submits SQL with the snapshot.
5. The system receives the answer with SQL and the snapshot.
   a. First, the system issues SQL to the JR database. If the SQL statement could not be correctly performed, e.g. syntax error, the system sends back the error reply. Here, the counter c is decremented by 2, $c := c - 2$; If the SQL is performed on the JR database, the system moves up to the next step.
   b. Next, the snapshot R obtained by performing SQL is compared with the answer snapshot A stored in the Answers.
      (1). The application server AS issues the following SQL to the metadatabase (MDB) in order to obtain the number N_R of tuples in R :

SELECT COUNT(*)
FROM R

(2). Similarly, AS issues the following SQL to MDB and obtains the number N_A of tuples in A :

SELECT COUNT(*)
FROM A

(3). AS sends the following SQL to MDB :

SELECT COUNT(*)
FROM R,A
WHERE R.name=A.name

Here, name is a common attribute in R and A. AS obtains the number N_{BRA} of tuples included in both R and A.

(4). If N_R=N_A=N_{RA}, AS decides that R is correct. Otherwise, R is wrong.

c. If the answer is OK, the counter c is incremented by two, i.e. $c:= c +2$; Otherwise, $c:= c -1$ and the error reply is sent back to the student. If $c \geq p$, the student moves up to the succeeding class and $c:= 0$. If $c \leq m$, the student moves down to the previous class. The student has to study SQL at the previous class again.

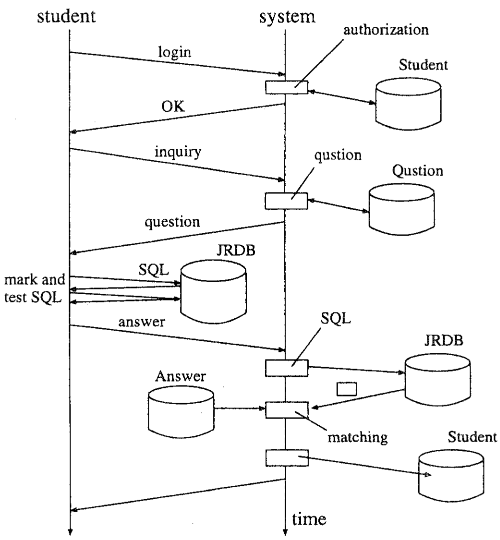

Figure 4: Overview

[Example] Let us show you an example of the system. First, a student starts at the following web page. The student "田中" inputs the user identifier and password.

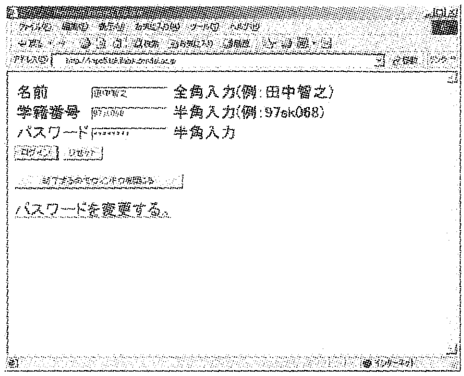

Figure 5: Login.

After the student is an authenticated one, the system gives a question for the current questions class. Then, the student makes SQL for the question in Japanese [Figure 6]. The SQL statement is submitted to the system. The SQL statement is automatically performed. If the result derived by the SQL statement is the same as the correct snapshot stored in the system, the system sends OK to the student. Otherwise, this system sends NG to the student.

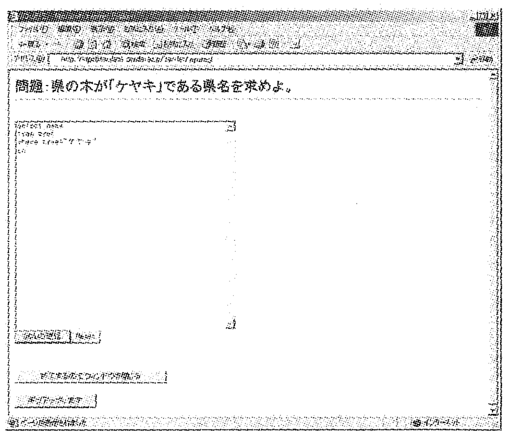

Figure 6: Input of SQL.

The student's accesses to the system are logged in the system [Figure 7]. The instructor obtains how each student uses the system. For example, a student who uses the system only just before the deadline is one who does must study well.

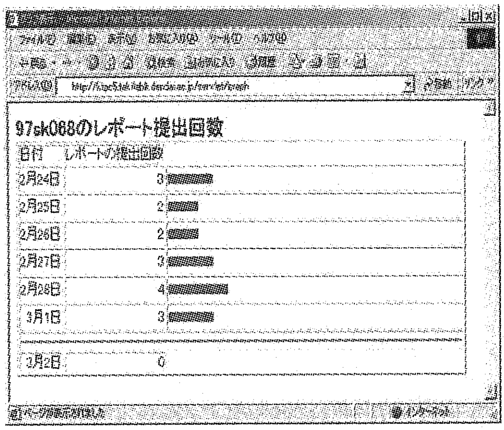

Figure 7: Status of report submission.

5 Concluding Remarks

This paper discussed the design and implementation of the database course. The system gives a unique question to each student. The student makes a SQL statement and submits it to the system. The system evaluates the SQL statement and then sends back the result to the student. By using the system, students study how to use SQL.

References

[1] Proc. of IEEE ICOIN-15, *http://www.takilab.k.dendai.ac.jp/conf/icoin15/*, 2001.

[2] Chang, S. K., Arndt, T., Guo, F. R., Levialdi, S. U.,Liu, A. C., Ma, J., Shih, T. and Tortora, G., "Macro University," *http://www.cs.pitt.edu/~chang/cpdis/macro-u.html*, 2000.

[3] MPEG Requirements Group, "MPEG-4 Requirements," ISO/IEC JTC1/SC29/WG11 N2321, 1998.

[4] SRC, eIT, *http://www.src-j.com/*

[5] Tommaso Leo, Marco Marini, Maurizio Maurizi, Maurizio Panti and Salvatore Valenti, "Web-based Teaching and Training on Motor Disability Assessment," *Proc. of 2000 Int'l Conf. on Information Society in the 21 Century: Emerging Technologies and New Challenges (IS2000)*, 516–524, 2000. IS2000,2000.

[6] JGN *http://www.jgn.tao.go.jp/*

Appendix : JRDB's schema

The schema of JRDB is shown as follows :

```
CREATE TABLE Questions(
  qno INT,
  lno INT NULL,
  query TEXT NULL,
  sql TEXT NULL,
  tname VARCHAR(7) NULL,
  atcount INT NULL,
  dis BIT,
  qname VARCHAR(50) NULL,
  qdate SMALLDATETIME NULL,
  rname VARCHAR(50) NULL,
  rdate SMALLDATETIME NULL,
  ord INT NULL)
CREATE TABLE Answers(
  qno INT,
  attno INT NULL,
  attname VARCHAR(8) NULL)
CREATE TABLE Students(
  sno CHAR(7),
  sname VARCHAR(50),
  pass VARCHER(10) NULL,
  cqno INT NULL,
  clno INT NULL,
  lcn INT NULL,
  len INT NULL,
  tcn INT NULL,
  ten INT NULL,
  ustate INT NULL,
  flg INT NULL,
  que INT NULL)
CREATE TABLE History(
  sno CHAR(7),
  qno INT,
  sqno INT NULL,
  reqdate SMALLDATETIME NULL,
  repdate SMALLDATETIME NULL,
  result BIT,
  repsql TEXT NULL)
CREATE TABLE Classes(
  sno CHAR(7) NULL,
  luno INT NULL,
  ludate SMALLDATETIME NULL,
  deadline SMALLDATETIME NULL)
CREATE TABLE Moveclass(
  lno INT,
  cmax INT,
  emax INT)
```

JRDB includes data on all stations and lines in JR. In addition, JRDB includes information of stations and lines of private lines in Kanto distinct. The size of JRDB is 80MB.