A Web-Based Classroom Environment for Enhanced Residential College Education

Kai Cheng, Limin Xiang, Toyohiko Hirota, and Ushijima Kazuo

Faculty of Information Science, Kyushu Sangyo University,
3-1 Matsukadai, 2-chome, Higashi-ku Fukuoka 813-8503, Japan
{chengk, xiang, hirota, ushijima}@is.kyusan-u.ac.jp

Abstract. In this paper, we describe a web based virtual classroom environment, called WTS (Web-based Teaching Support), to enhance classroom based residential teaching and learning. WTS implements an online view of physical classrooms with the "desks" and "seats". Teaching information associated with "classrooms" is managed by a database system. WTS also includes such capabilities as online attendance check, personalized notification for course participants, as well as a variety of tools to facilitate lecture preparation and post-lecture processing for teachers. We discuss the main design issues and the implementation details. We also evaluate its usefulness by analysis of real-world teaching data.

1 Introduction

College education is primarily conducted on the basis of classroom, a physical place where students and teachers get together and face-to-face interactions take place. Although online learning without face-to-face interactions is getting increasingly popular, residential college education will still be important and cannot be replaced in the foreseeable future. This is because the social aspects of college, e.g., learning in groups, informal faculty-student discussions, and so on, are too valuable and important [4]. Frequent faculty-student contact in and out of classes is the most important factor in student motivation and involvement. Faculty concern enhances students' intellectual commitment and encourages them to think about their own values and future plans [5].

One problem with the classroom-based education is the poor cost / performance record due to the lack of efficient way to manage detailed information for high quality in-class teaching. First, from the viewpoint of pedagogical psychology, it is important for teachers to bear in mind the names and faces for all students participating in their classes. However, this will be a heavy mental burden even for a middle-sized class with tens of students. Second, teachers usually want keep track of the changing attitude of students by observing how student attend a course. A possible way is to observe how a student takes seats in the classroom. This is much more cumbersome because it requires efficient way to gather and manage detailed information about the class participation for all students.

R.W.H. Lau et al. (Eds.): ICWL 2005, LNCS 3583, pp. 56 – 65, 2005.

To cope with these problems, we have been developing a web-based teaching support system, called WTS (Web-based Teaching Support). Instead of focusing on putting teaching contents online, our system is aimed to efficiently manage teaching information by implementing an analogy to physical classrooms with desks and seats. Information associated with courses, participants, teachers are managed in a database system. Students take attendance by simply registering their seat numbers. After registration, they can view an image of their own appearing on the classroom layout.

In the following sections we will describe in detail about the design and implementation of the system. In section 2, we give a brief overview of the system, outline its main features. In section 3, we describe in detail the issues in design implementation of each feature. Section 4 includes the design of the underlying database that supports implementation of the features described so far. We report some evaluation results in section 5. Section 6 includes the concluding remarks.

2 Overview of the Web-Based Classroom Environment

In this section, we give a brief overview of the web-based classroom environment, WTS. Further details will be given in the following sections. WTS supports an online view of a physical classroom with desks and seats and allows online attendance taking by entering seat numbers. It enhances a “real” classroom by management of detailed information about the associated teaching and learning activities, e.g., lecture notes, lecture schedule, course participation and test results.

Fig. 1. The Web view of a classroom (left) and a physical classroom (right) where all students personally hold a rental PC

The system has the following features:

1. *Password Protected Access*. Users (students or faculty members) accessing to the system should provide a valid user id together with a password. The authenticated id is used to enforce access control. Students are only allowed to access their own study records, while teachers can view and modify records of all stu-

dents in his/her class. User id is also used to retrieve associated information for personalization.

2. *Online Attendance Check*. After logging, a student can take attendance online simply by entering his seat number. System records the provided seat number and the associated information about the courses, IP addresses from which this attendance check is made. Attendance check is open for the students who registered to the course and during the time when the course is given.
3. *Seating Layout View*. The system enables an online view of classroom layout. An identification image is displayed as a representation of the student who is taking the specific seat. This feature makes it very convenient for instructors to tell who are sitting on that seat by name and face.
4. *Personal Notification*. Students can check participation status and test grades of their own. They can also ask questions and report problems about the courses and the system as well.
5. *Scheduled Publication Area*. Lecture notes and other resources are put online in accordance with the teaching schedule. For example, sometimes, lecture notes have to be made available after the lecture finished, whereas test questions should be put online for the specific time period.
6. *Aggregate and Statistical Analysis*. The system supports quick summary report and a variety of statistical analysis. This feature cuts the tedious and error-prone statistics work for instructors.

Despite of the above mentioned features, WTS also supports course scheduling and preparation.

3 Enhanced Classrooms with Teaching Information Management

In this section, we describe in detail the design and implementation of three features of importance in WTS. As WTS is a database based application, the implementation involves database operations. We postpone the description of database design to the next section.

As the system is written in PHP script language, in the following description, we use the form of $var, $array[$i] to represent scalar and array variables respectively as in most script languages, such as PHP, Perl, etc.

3.1 Course Scheduling

A course consists of a series of lectures with respect to related sections of course content. Course content is described in detail in syllabus. Lectures have to be scheduled correctly for the specified day of the week (DOW) and period of a day. To automate the course scheduling, the following data or parameters should be provided.

1. *$N*: number of lectures that constitute the course with an course id *$cid*;
2. $day1, $day2: initial and final dates of the term when a course is taught;
3. $holidays: holidays and other specified days that no lecture is given; Holidays are stored in database, allowing insert, update, and delete.

4. $dow, $period: the day of the week and the period for the course to be taught. Both $dow and $period for the course can be retrieved from database.

With these data and/or parameters, system can automatically generate a schedule for the course:

(1) Initialization. set $i=1; repeat (2)~(6) until $i > $N
(2) gets a $day on the $dow from [$day1, $day2] in order;
(3) check if $day is in $holidays;
(4) if $day is a holiday, goto (2) and repeat;
(5) otherwise, insert <$cid, $i, $day> into the schedule list
(6) $i++, goto (2) and repeat;

3.2 Online Attendance Check

Once a student has logged in to the system, he can check the courses that he can attend. In a specific time period, a student is allowed to attend at most one course. Suppose current time is $time, which is in the class period $period. $period and student id $sid are used to retrieve the courses information. If there are some courses to attend, the participation check screen appears and the student is asked to enter his seat number, $seatno. The $sid, $seatno, $time and the associated session data are then recorded in database. Session data is used to validate the operation, e.g., an IP address out of the valid scope will cause rejection to the database.

Once the attendance check finished, students can view their attendance status for this course since its beginning. This is much more convenient than a paper-based attendance check, whose results will be given in a few days and the results may be error-prone.

3.3 Classroom Layout

We now describe the implementation of classroom view. The feature requires a variety of information to be retrieved. First, the layout of the specified classroom should be retrieved from database. The layout is a list of the coordinates and the associated seat number for all seats in that classroom. Second, a list of the occupied seats, i.e. the coordinates and the student id for the one who took the corresponding seat, should also be provided. Finally, the parameters about the classroom have to be described as follows:

1. $layout: list of all seats, in the form of (row, col, pos)=>seatno
2. $seated: list of occupied seats, in the form of (row, col, pos)=>sid
3. $maxrow, $maxcol, $desksize: classroom parameters

Provided these data, we can draw the layout of a classroom as follows.

```
drawLayout($layout,$seating,$maxrow,$maxcol,$desksize){
  for ($i=1; $i<=$maxrow; $i++){ //row by row drawing
     beginNewRow()
     for ($j=1; $j<=$maxcol; $j++){
        beginNewDesk();
```

```
        for ($k=1; $k<=$desksize;$k++){
           beginNewSeat()
             if (empty($layout[$i][$j][$k]) continue;
           else $seatno = $layout[$i][$j][$k];
           if (empty($seated[$i][$j][$k])
              drawSeat($seatno,null);
           else{
              $sid = $seated[$i][$j][$k];
              drawSeat($seatno,$sid);
           }
           endNewSeat();
        }
        endNewDesk();
     }
     endNewRow();
  }
)
```

drawSeat($seatno, $sid) is a function that draws a seat with number $seatno on the desk If student id $sid is null, it draws a black seat. Otherwise an image provided by the student will be attached on the seat. Student id and name will also be printed there.

4 Database Design

The features described so far depend heavily on the database system. In this section, we present the database design issues. We begin with the design of a conceptual mode using ER diagram. We them give the details on how to transform the ER model to relational database scheme.

The conceptual model of the system is depicted in Fig. 2. The ER diagram consists of 6 entity sets (including 2 weak entity sets), and 4 relationship sets. A course consists of a series of lecture. **Lecture** is called a *weak entity set* with *owner* **Course** due to the lack of a natural way to identify it. Similarly, **Seat** is a weak entity set with owner **Classroom** since a seat can not exist independent of a classroom.

Course, **Student**, **Teacher** and **Classroom** are obviously entity sets. However, it is not easy to tell whether **Timetable** should be an entity or relationship. We take Timetable as an entity instead of relationship is for two reasons. Firstly, actually there is a timetable independently designed by registration office staff. Secondly and most importantly, other relationships, say **Study** and **Schedule** depend on an entity that identifies a unique lecture. Such entity is neither **Course**, since a same course may be given by more than one instructor, nor **Teacher**, as one instructor may hold more than one course.

Relationship **Study** describes how student takes courses. The multiplicity constraint M:N, means that a student can take one or more courses and a course can be taken by one or more students. Since a course can simultaneously be given by different instructors at different time, in different classrooms, students have to decide which one to take. Relationship **Schedule** describes the arrangement of dates that a course will be given. The multiplicity constraint is M:N. That is, each course can be given in several times and each content part can be taught by different instructors at different times. Finally, relationship **StudyRecord** describes details about learning activities of students over time. Students take attendance by entering their seat numbers, which will be stored in this **StudyRecord**.

Let R(A1, A2,…, An) represent a relation scheme, where R is relation name, and Ai(i=1,2,…,n) are attributes. Key attribute is indicated by underlining the corresponding attribute. We can transform the conceptual model in Fig. 2 to the following relational database scheme.

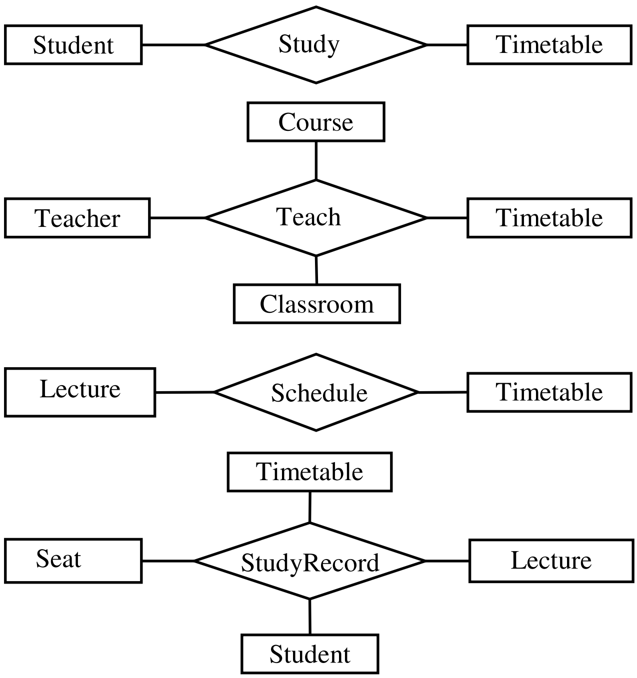

Fig. 2. Entity-Relationship (ER) Model

Course(cid, cname, credit, cyear, cterm)
Student(sid, sname, sex, dept, classid, sdate)
Teacher(fid, fname, sex, dept, office, tel)
Classroom(roomid, rname, capacity, maxrow, maxcol, desksize)
Seat(roomid, seatno, row, col, pos)
Lecture(cid, seq, title, detail)
Timetable(tid, fid, cid, classid, wday, period, roomid)
Study(sid, tid, sdate)
Schedule(tid,seq,sdate)
StudyRecord(sid, tid, seq, srdate, srtime, seatno, ipaddr, hostname, grade)

The detailed information for drawing the layout of a classroom is kept in relations **Classroom** and **Seat**. In **Classroom**, *maxrow* and *maxcol* define the rows and columns of desks in the classroom, while *desksize* indicates the number of seats in each desk, whose typical values are 1, 2, and 3. Seats in each classroom are listed in relation **Seat**. Each seat is associated with the coordinates <*row*, *col*, *pos*>, where *row* and *col* are coordinates of desk and *pos* is a number representing the position of a seat in the desk, e.g., 1 represents the first (leftmost) seat, 2 represents the second seat next to 1, and so on. Given the *roomid* and *seatno*, it is easy to locate seat in the classroom layout.

5 Evaluation Analysis

To evaluate the proposed system, we have analyzed data of the teaching activities accumulated in the past year. The goal is to (1) explore if there are any associations between the seating layout and the test grades. If any what the associations would like to be; (2) find the quantitative evidence to the system's effect. In this section, we present results about these preliminary analyses.

5.1 Deployment Environment

The system has been deployed in the faculty of information science. Our college was set up in 2002 and now there are about 600 enrolled students and 22 faculty members. As a newly-founded college, our faculty is enriched by leading educational infrastructure that can afford to support efficient education. The infrastructure includes:

1. *Personal hold rental PCs*. All students hold a rental notebook type PC once they enroll in the college.
2. *Network sockets for all seats.* Students can access to the Internet through network sockets on all seats of all classrooms.
3. *Lecture video recording*. All lectures given in our faculty are recorded on video. The video archives are online available and can be retrieved through web browsers.

The WTS system has supported two undergraduate courses, "database systems", and "data structures and algorithms" during the last year. The evaluation is based in data accumulated so far.

5.2 Associations Between Seating Layout and Test Grades

In general, students often taking a seat on front of the classroom seem to be those who are join the class aggressively. In contrast, students who often take seats on the rear half of a classroom tend to be inactive because they may be not quite interested in the course.

We analyze the accumulated study records, to see if this can be justified. To investigate the association between seating and test grades for course 'X' of class 'A', we issue the following SQL queries to the database server.

```
SELECT S.row, ROUND(AVG(R.grade),1)
FROM StudyRecord R, Timetable T,
     Classroom NATURAL JOIN Seat S
```

```
WHERE R.tid=T.tid AND S.roomid=T.roomid AND T.cid='X'
     AND S.seatno=R.seatno AND R.classid='A'
GROUP BY S.row
ORDER BY S.row
```

The results for class X of "database systems" and classes for Y, Z of "data structures and algorithms" are shown in Table 1. From the table, we can find that test results are partially associated with seating rows in that

Averagely, students on the first 1~2 rows are better than others. This is shown in Table 1 by bold typed numbers on the right-hand side of each column. Class X and Class Y have two rows with peculiar good grades while in Class Z, only first row with better grades. On the other hand, students seated behind seem not fall into the worst group. The worst groups (underlined in Table 1) are found on the front half near the central line. Table 1 seems to suggest that,

Table 1. Seating status and the grades

Class A		Class B		Class C	
Row	Avg. Grade	Row	Avg. Grade	Row	Avg. Grade
1	**77.5**	1	**83.0**	3	**78.0**
2	**77.0**	2	**85.5**	4	71.5
3	68.0	3	**77.5**	5	70.0
4	65.5	4	*61.5*	6	72.5
5	72.5	5	75.5	7	66.0
6	*64.0*	6	75.5	8	*61.0*
7	69.5	7	74.0	9	72.5
8	69.5	8	79.0	10	74.5
9	69.5	9	72.0	11	72.0
10	66.0	10	74.5	12	73.0
11	67.5	11	74.5	13	69.5

1. Students always taking seats on the first 1~2 rows are the most active learners and their test results are often much better than the rest. In this case, we can say that the association is obvious.
2. The worst group of students is most likely found on the first half of seats near the central row. This indicates that although they did their best and took part in the course quite actively, the results may still be not ideal.
3. Students seating on behind half of classroom form a group between the best and the worst.

5.2 The Quantitative Evidence to the Effects of WTS

Although we have obtained numerous informal feedbacks from both instructors and students during the deployment, without quantitative evidence to the effects of WTS, it is still not quite enough.

To deal with this problem, we have examined the change of absentees for two classes that began to use the system on the half way of the course "data structures and algorithms". As shown in Fig. 3, both class A and class B began to use the system from the lecture 6. The paper-based records for lecture 1 though lecture 5 were manually entered by the instructor after deployment of the system.

From Fig. 3, we can see that due to the deployment of WTS, the number of absentees actually drops down. There are an average 13 absentees in class A during lecture 1~5. However, after WTS is put in use, the average number of absentees' decreases to 9. Similar result can be found in class B: The average number of absentees for lecture 1~5 is 8, whereas the number for lecture 6~13 is down to 6. Therefore, we can conclude that *the deployment of the WTS has positive impact on the improvement of attendance rates.*

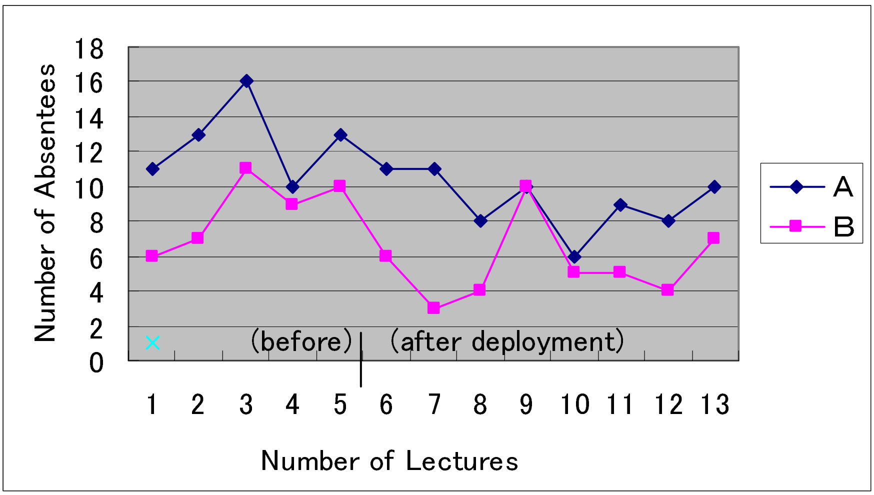

Fig. 3. Change of absentees before and after deployment of WTS: After WTS is put in use, the average number of absentee decreases in both class A and class B

6 Concluding Remarks

The Web provides a convenient platform for efficient content delivery at low cost while the database technology enables efficient management of large amount of data. Combination of the two gives a promising way to enhance traditional classroom based education. In this paper, we have described a web-based teaching support system, WTS. Instead of supporting learners for individual learning as in most E-learning systems, WTS was aimed to support teachers for efficient management of online teaching information. The online view of classroom layout was designed to facilitate teacher-student communications. By preliminary analyses, we have shown the usefulness and potential of the proposed system.

Our future work is to improve the flexibility and scalability of the current system for adoption in more courses. We are also going to investigate the possibility of combining features of E-learning features so as to support off-campus learning. A possible extension is to combine the legacy systems, e.g., lecture video recording system, with the proposed WTS system, so that students can take the "recorded" course if they missed the "live" one. Students can also use the system for review purpose.

References

1. Sheard, J., Lynch, J.: Challenges of web-based learning environments: Are we student-centred enuf? In: Second International Conference on Web-Based Learning (ICWL2003). (2003) 1–11
2. Madjarov, I., Boucelma, O., B´etari, A.: An agent- and service-oriented e-learning platform. In: Third International Conference on Web-Based Learning (ICWL2004). (2004) 27–34
3. Boschmann, E.: The Electronic Classroom: A Handbook for Education in the Electronic Environment. Information Today (1995)
4. Ullman, J.D.: Improving the efficiency of database-system teaching. In: SIGMOD Conference. (2003) 1–3
5. Chickering, A. W., & Gamson, Z. F. Seven Principles of Good Practice in Undergraduate Education. *AAHE Bulletin*, (1987) 39, 3-7.