2014 4th International Conference on Education, Research and Innovation
IPEDR vol.81 (2014) © (2014) IACSIT Press, Singapore
DOI: 10.7763/IPEDR.2014.V81.25

Web-Based Class Scheduling for a Collaborative Preparation of Block-Based Schedules

Khavee Agustus W. Botangen[1+]

[1] Central Luzon State University, Science City of Munoz, Nueva Ecija, Philippines

Abstract. This work aimed to develop, implement, and evaluate a web-based scheduling application for a collaborative preparation of class schedules at the Central Luzon State University. The development integrated an algorithm for automated plotting and conflict-checking of the different scheduling entities based on their availability considering different constraints and preferences. The application was deployed using a free Virtual Private Network software, Hamachi. Four academic units in the University collaboratively created their class schedules using the scheduling application. The occurrences of problems relative to the scheduling of classes in the academic units that used the application were compared with the five academic units that did not use it. In the comparisons made, there is a significant effect of the use of the application to the occurrences of the observed scheduling problems.

Keywords: scheduling; class-scheduling; collaborative scheduling; web-based scheduling

1. Introduction

Academic institutions and universities often find difficulties in scheduling classes [1]. This difficult task is devoted with hefty amount of time, human, and material resources [2]. Several factors such as diverse student groups, time allotments, courses, rooms, teachers, and scheduling discretions are being considered in the task. For institutions composed of several academic units, the complexity is worsened by a decentralized preparation of schedule. In this approach, several uncontrolled scheduling elements should be taken into account such as schedules of subjects to be taken offered by other units, the use of resources such as rooms or facilities located in other units, schedule preferences and constraints implemented by other units, all of which are affecting schedule preparation processes. The above mentioned scenario holds true for Central Luzon State University. The unit registrars spend too much time scrambling and fitting schedules, mindful of considering every known factor. They need several time-consuming face to face consultations among themselves to settle anticipated conflicts. Individual consultations between registrars pose difficulty because of the distances between the units they belong. They need to travel from one location to another that makes scheduling both physically and mentally arduous task. These situations contribute to the lengthy schedule preparation and the frequent delay in the submission of final schedules to the in-charge of registration. Moreover, manual checking does not guarantee a conflict-free schedule. In most occasions, several problems arise because of scheduling mistakes. Schedule conflicts affect the normal flow of processes during registration. Classes are delayed until conflicts are rectified. The worst scenario brought about by these scheduling lapses is when an affected registered-student is forced to drop or change a course/subject enrolled due to change in the schedule.

2. Objectives and Methodologies

The main objective of this work was to develop, implement, and evaluate a web-based scheduling application for a collaborative preparation of class schedules at the Central Luzon State University. The

[+] Corresponding author. Tel.: + 63 44 4565267.
E-mail address: xjasonk2000@yahoo.com.

development phase of the application integrated an algorithm for automated plotting and conflict-checking of the different scheduling entities based on their availability considering different constraints and preferences. In addition, common formats and comprehensive graphical representations of schedule reports were created. In the implementation phase, the application was deployed in a server and was accessed through a Virtual Private Network software, Hamachi. Four academic unit registrars used the deployed application to prepare their respective schedules. The evaluation phase used three scheduling concerns to be observed and be gathered from the academic units' registrars: i) the number of consultations made with a particular unit registrar; ii) the number of schedule conflicts identified on schedules related with courses offered at a particular academic unit; and iii) the estimated time used to prepare schedules. The comparisons of observation data were made between those that used and those that did not use the application.

3. Results and Discussions

3.1. Application development

The application is designed primarily to emulate the scheduling processes in the university. In addition, several scheduling applications implemented in other universities were considered. The SlotManager [3] describes similar major components. It has three components, the interface, the database, and the algorithm engine. Its interface handles user data management and report management. Its database is relational and contains the details of created schedules. The algorithm engine detects block-schedule conflicts, instructor-schedule conflicts, and provides available resources for creating schedules. Similarly, eClasSked [4] which provides a decision support capability, had also adapted this model. The input information module is a database that stores all information relating to courses, classrooms, and instructors. The user interface module and optimization module are both for system control and modeling mechanisms that produce the scheduling solutions. The report generation module presents the solution results as management reports, including reports on course schedules, the use of classrooms, and schedules for individual instructors.

For this application, each major process is handled by a separate module (Fig. 1). The Data Management Module handles inputting, editing, and deleting data. This feature makes sure that all of the needed data of the scheduling entities are entered and fixed to ensure the integrity of the inputs. The initial step of the actual scheduling is handled by the Course Assignment Module. This module is used to assign the courses to be enrolled by a certain block. The Scheduling Module facilitates the management of available time, room, and faculty to fit a certain course assigned to a block. This module is responsible to create a reliable, conflict-free schedule considering all the set attributes and variables. Several options of creating schedule are featured by this module. The manual-scheduling option needs a user to fit a desired schedule for the course. The auto-scheduling option lets the algorithm of the system create schedule for the course. The copy-previous-schedule option creates a schedule imitating the most recent previous schedule made for the same course for the past academic term. The Result Storage Module stores the created schedule in the database. The View Schedule Module lets the system display to the user scheduling results and system generated information in the most comprehensible way.

Privileges and restrictions, for accessing every module, is dependent on the type of user (Fig. 2). The University schedule administrator manages and controls the overall use of the application. The unit registrars will be the main users having privileges and restrictions based on the college or unit they represent. A unit registrar can manage and control data and scheduling entities associated with his own unit. The schedule assignment of a block or a course can be managed by a unit registrar representing the same unit or college as those mentioned entities. Students, faculty members, and other guest users or the public in general, may use the application for viewing or printing reports or schedule information.

3.2. Scheduling behavior

The scheduling process which is attributed to the Scheduling Module of the application follows the principle of the Greedy Algorithm [5], [6]. This algorithm selects an option by choosing what is most available. There are three sequential sub-processes of the Scheduling Module (Fig. 3). To create a schedule, these processes are repeatedly executed in a one-dimensional forward pattern without backtracking. The first process looks for a time-slot available in the block that can accommodate the assigned course. The first available option that fits is selected. The second process finds a room that is available at the selected time. It also follows that the first available room that can be assigned in the selected time is chosen. The third process looks for an appropriate faculty to handle the given course. In the search for a faculty, the first qualified option that fits is selected.

The behavior of the algorithm limits the decision-making of the entire scheduling process to create the most optimal choice. Whenever at least one process is unsuccessful, there's no way to rearrange other created schedules to create availability. This is when the user needs to intervene with the process to interactively arrange the schedules.

3.3. Implementation and evaluation

Using the application, the initial stage of schedule preparation is data input. In this stage, details about the scheduling entities are entered. Most scheduling preferences and constraints depend on these inputted parameters. Accordingly, the application offers a module for easy management of these data. Users use the module to manually check entries before plotting schedules. However, in the implementation of this work, problems occurred because of inconsistent or redundant data inputted by the unit registrars. Physically the same rooms or subjects were given different names in every college. The same problem was encountered in the scheduling implementation at Purdue University [7].

Table 1 shows the experiences of the unit registrars of different academic units in the University in preparing schedules for a particular term. It details the number of occurrences of three (3) scheduling concerns that were observed involving each academic unit: i) the number of consultations made with a particular unit registrar; ii) the number of schedule conflicts identified on schedules related with courses offered at a particular academic unit; and iii) the estimated time used to prepare schedules. These scheduling concerns were primarily accounted in this work with the basis that in each academic unit, there are created schedules that involves entities such as courses and rooms managed by other units. It follows that if a course assigned to a block is offered by other units, the faculty to handle or room to use cannot be fully controlled by just one registrar. The approval of other concerned registrars is needed.

In the data presented (Table 1), the total numbers of created schedules of each academic unit were noted (a). In addition, the number of schedules created involving courses offered at academic unit X and schedules using rooms under the supervision of X were accounted (b). The table also shows the number of occurrences of scheduling problems relative to the schedules that involve academic unit X (c, d, and e). In the comparisons made between the academic units that used the application for collaborative scheduling (H, I, J, K) and academic units that did not use it (L, M, ..., P), there is a significant effect of the use of the application to the occurrences of the observed scheduling problems. Firstly, in the frequency of consultations made between the registrar of X and the registrars of the academic units that did not use the application, the average ratio of the number of consultations that occurred to the number of created schedules is (0.19). This is significantly a higher consultation occurrence than (0.05) which is for the academic units that used the application. Secondly, the occurrence of schedule conflicts is higher in the academic units that did not use the application. The average ratio of the number of schedule conflicts that occurred with respect to the number of schedules created is (0.05), which shows a higher conflict-occurrence compared to (0.01) for the academic units that used the application. However, in the third observation, the average ratio of the estimated number of days consumed by unit registrars in preparing schedules to the total number of created schedules in the academic units that used the application is (0.012). This is ironically higher compared to (0.010) which is for the academic units that did not use the application. The users' first time use of the System may have affected this result. Thus, it is subject to change once familiarity on the use of the application is established.

There were several other factors that were not considered in the evaluation of this work. Some of these factors are the physical distances between the unit registrars; the number of blocks and courses scheduled in each unit; the number of unit registrars to be consulted for a particular schedule; prioritizations of which unit's schedule should be plotted first; the number of scheduling entities involved in each schedule; the time spent per day in doing the scheduling task, and the discretionary options of the unit registrars. These factors may affect in any way the probability of occurrence of the stated problems.

3.4. Significance of this work

The output of this work which is the web-based class scheduling application will be used as a tool for preparing schedules for a particular term in the different academic units of the university. Through it, the task of the different unit registrars in preparing schedules will become lighter, easier, and efficient. The application will be able identify and prevent schedule conflicts eliminating common problems in the scheduling process. It may be used to produce an on-time, reliable, conflict-free, and efficient schedule taking into account the different entities attached such as block, room, course and faculty. The schedule generated will be used as a valuable data in the registration thus minimizing registration problems frequently encountered due to scheduling errors. Moreover, delays of classes caused by erroneous scheduling can be avoided with the application's assurance of handling and checking these schedule related problems. With the possible reports that can be generated from the application, information can be produced to locate unused rooms to maximize their use. Some rooms in other academic units are underutilized because of their small population. On the contrary, other academic units with large populations lack rooms to hold classes. Through the application, unit registrars can easily identify these vacant rooms for a possible equally distributed room utilization. The application will also consider the maximum allowed number of students in a room to fit which block is to accommodate. Faculty workload can be easily prepared because the application includes the scheduling of faculty members to handle a course. The problem on uneven distribution of teaching loads can be easily addressed. At the same time, the need for additional faculty to handle excess teaching loads can be determined and addressed earlier.

4. Tables and Figures

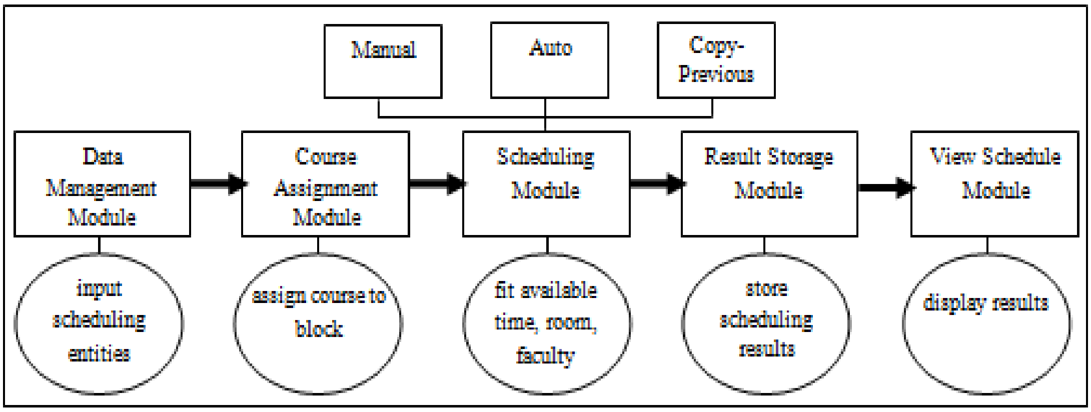


Fig. 1: A diagram showing the modules of the scheduling application.

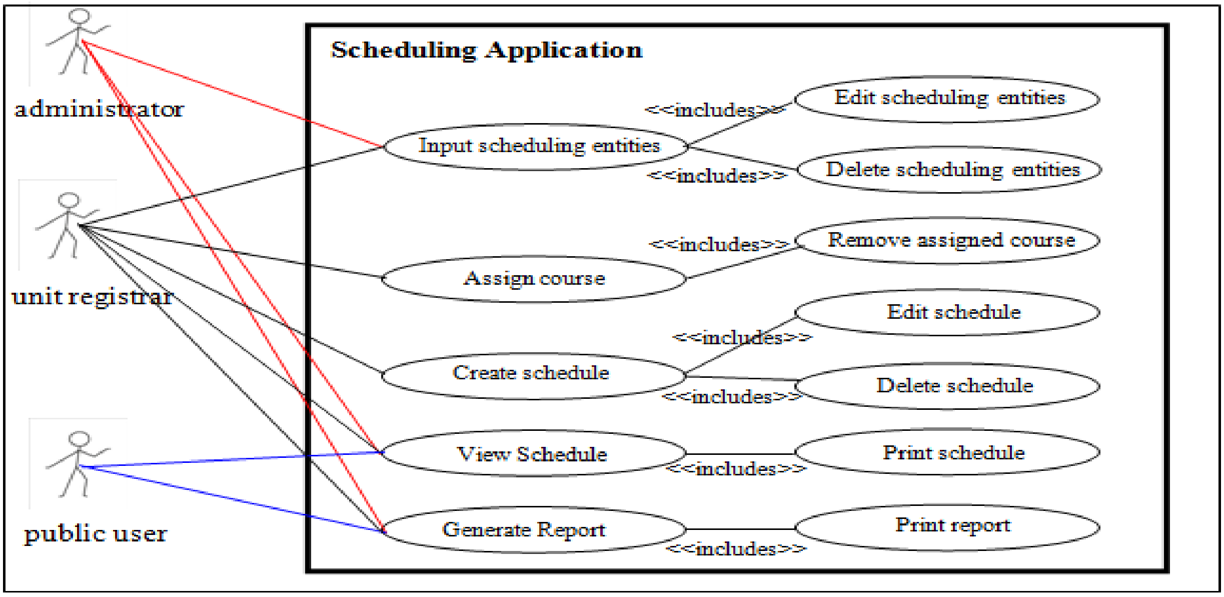


Fig. 2: A Use Case Diagram for the scheduling application showing the users and processes they can interact with.

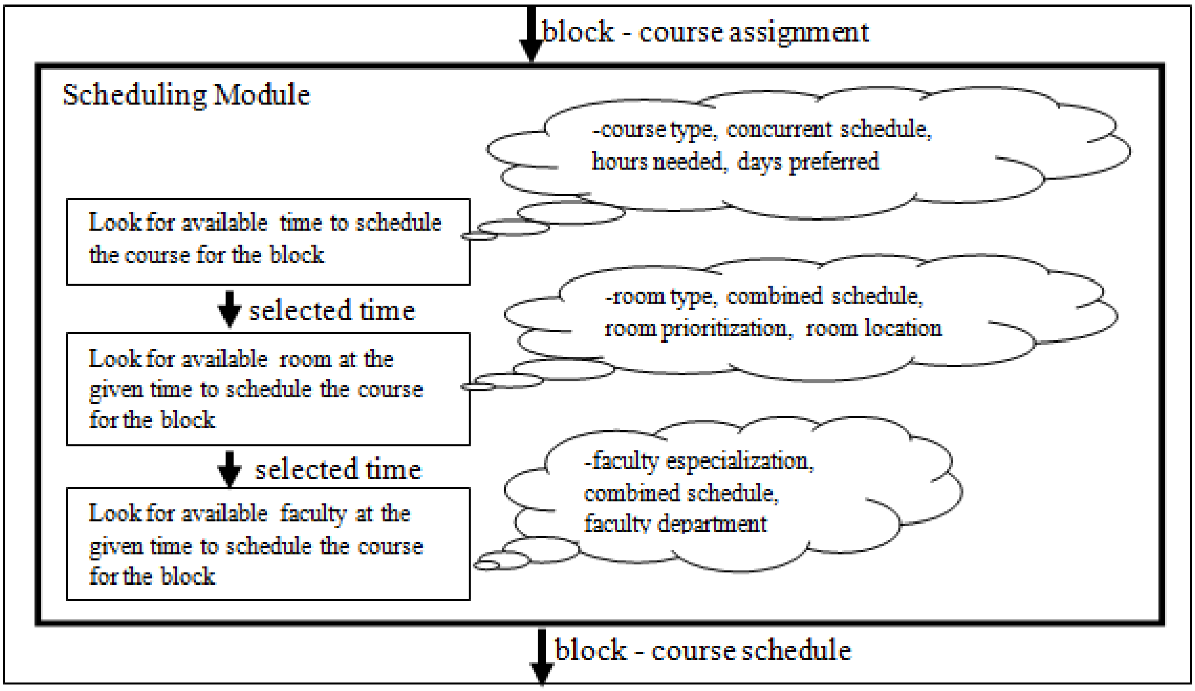
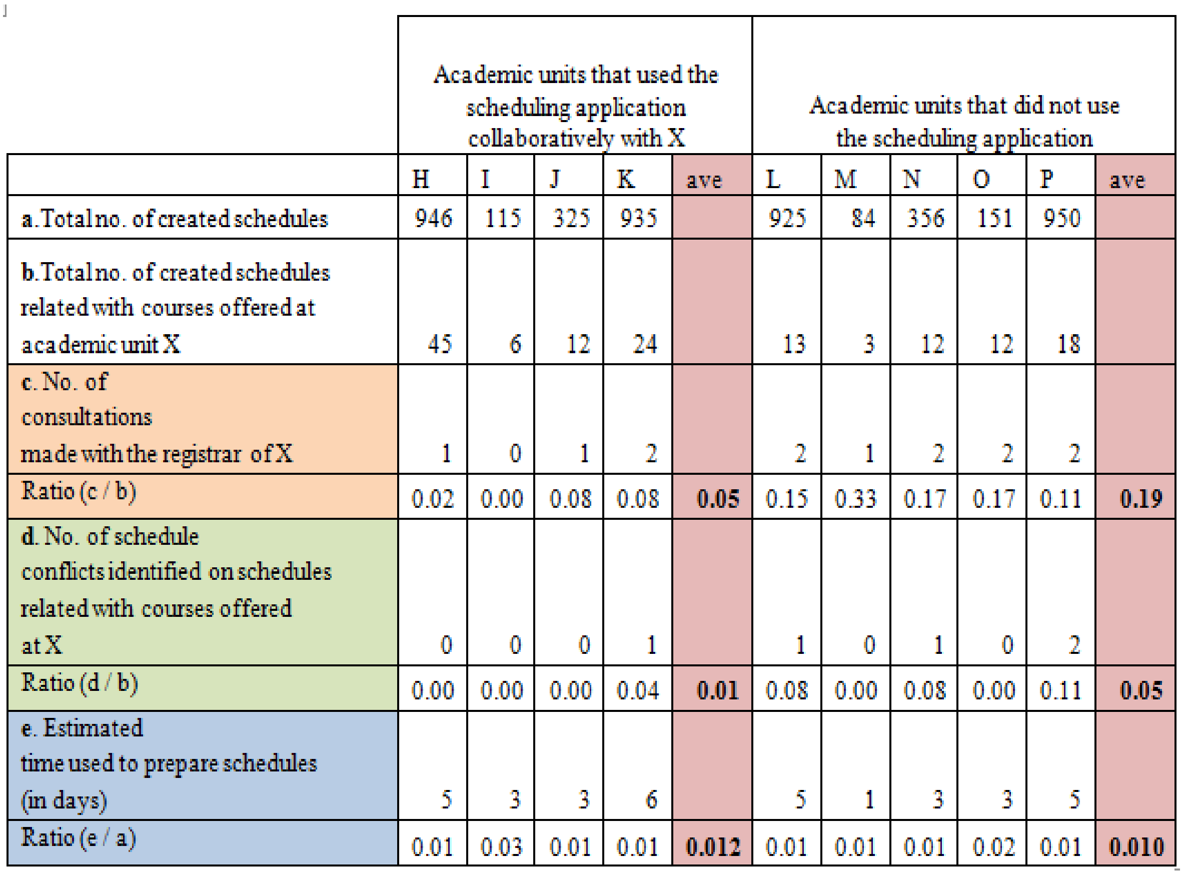

block - course assignment
Scheduling Module
Look for available time to schedule the course for the block
-course type, concurrent schedule, hours needed, days preferred
selected time
Look for available room at the given time to schedule the course for the block
-room type, combined schedule, room prioritization, room location
selected time
Look for available faculty at the given time to schedule the course for the block
-faculty especialization, combined schedule, faculty department
block - course schedule

Fig. 3: A figure showing the processes in the Scheduling Module of the application.

Table 1. Evaluation Data of the Scheduling Application



	Academic units that used the scheduling application collaboratively with X					Academic units that did not use the scheduling application					
	H	I	J	K	ave	L	M	N	O	P	ave
a. Total no. of created schedules	946	115	325	935		925	84	356	151	950	
b. Total no. of created schedules related with courses offered at academic unit X	45	6	12	24		13	3	12	12	18	
c. No. of consultations made with the registrar of X	1	0	1	2		2	1	2	2	2	
Ratio (c / b)	0.02	0.00	0.08	0.08	**0.05**	0.15	0.33	0.17	0.17	0.11	**0.19**
d. No. of schedule conflicts identified on schedules related with courses offered at X	0	0	0	1		1	0	1	0	2	
Ratio (d / b)	0.00	0.00	0.00	0.04	**0.01**	0.08	0.00	0.08	0.00	0.11	**0.05**
e. Estimated time used to prepare schedules (in days)	5	3	3	6		5	1	3	3	5	
Ratio (e / a)	0.01	0.03	0.01	0.01	**0.012**	0.01	0.01	0.01	0.02	0.01	**0.010**

5. References

[1] Gunawan A., Ng K.M., Poh K.L. Solving the Teacher Assignment-Course Scheduling Problem by a Hybrid Algorithm. (2007). Retrieved from http://www.waset.org/journals/waset/v33/v33-50.pdf

[2] R.A. Valdes, E. Crespo, J.M. Tamarit. Design and Implementation of a Course Scheduling System using Tabu

Search. European Journal of Operational Research. 2002, vol. 153.

[3] L. R. Foulds, D.G. Jonhson. SlotManager: A microcomputer-based decision support system for university timetabling. Decision Support Systems. 2000, 27(4) 367–381.

[4] Jaime Miranda. eClasSkeduler: A Course Scheduling System for the Executive Education Unit at the Universidad de Chile. Interfaces. 2010, Vol. 40, No. 3, pp. 196–207.

[5] T. M. Murali. Greedy Algorithm Lecture. (2009). Retrieved from http://courses.cs.vt.edu/~cs5114/spring2009/lectures/lecture04-greedy-scheduling.pdf

[6] Jeff Erickson. Greedy Algorithm Lecture. (2007). Retrieved from http://www.cs.uiuc.edu/class/sp07/cs473g/lectures/06-greedy.pdf

[7] Keith Murray, Tomas Muller, Hana Rudova. Modeling and Solution of a Complex University Course Timetabling Problem. (2006). Retrieved from http://wendang.baidu.com/view/060d78f8770bf78a65295429.html