

# SATA: A Web Based Scheduling Support System

Rajeswari Raju, Saiful Nizam Warris, and Hazlifah Mohd Rusli

**Abstract**—Developing a university course schedule is difficult. This is due to the limitations in the resources available. The process is made even harder with different faculties or departments having different ways of stating their schedule requirements. The person in charge of taking the schedule requirements and turning them into a proper course schedule is not only burden with the task of allocating the appropriate classes and time to lecturers and students, they also need to understand the schedule requirements. Therefore a scheduling support system named SATA is developed to assist ICRESS in the course scheduling process. SATA has been put to use for several semesters and the results have been encouraging. It won a bronze medal in the 2008 Invention, Innovation and Design competition (IID'08) and has been submitted to be patented in October 2008

**Keywords**—Course Scheduling, Scheduling Tool Aid.

## I. INTRODUCTION

IN organizations dealing with Education, the task of scheduling lecturer's timetable is a problem for both the administration personnel who create the schedules, and the lecturers who follow them. For the administrator, the process is a lot of manual work. This requires much studying of schedule requirements in order to find optimal fits for such metrics as room capacity, number of students, time slot, student availability, lecturer availability, concurrent lectures and etc.

Other than that, administrator too faces problems when in many cases such as: [1]

- (i) Turnover of academic staff.
- (a) Adding teaching staff: This problem can occur when teaching staff has just finished a study leave or new staff has joined. It can also be considered as a free staff.
- (b) Unplanned staff absences: This problem can occur due to retirement, illness, or emergence of other commitment of a teaching staff. It can also occur when teaching staffs gets a study leave.
- (ii) Dynamic of enrolments makes section or subject unavailable or inadequate.
- (a) Deleting sections or subjects. This problem occurs when there are not enough students to enroll in the subject. It can also be considered a free staff.

- (b) Adding new sections or subjects that previously are not being offered. This problem occurs in case of extra students or a new subject has been offered.
- (c) Adding/deleting activities (lectures, tutorials, seminars, etc.
- (d) Amending lecturers (swapped from one lecture to another, dropped from/ added to lecturers).
- (e) Grouping teaching activities.

Before SATA - Scheduling Support System using Web Based Application was in place, creating lecturers timetable was a manual process at University Technology MARA (UiTM), although UiTM uses ICRESS, a system used for scheduling purposes, the data that need to be input into ICRESS is still done manually. This becomes an issue when the administrator is not able to find the "best" schedule to fit needs of different faculties that come with different needs. The data comes in various formats, hand written, not consistent, and overlapping information.

SATA works with ICRESS to make the overall lecturer teaching schedule task hassle free, moreover the task need to be repeated every 5 months once, in semester basis. So with SATA, it is a helping hand.

The goal of SATA is to automate and improve the Course Scheduling System (ICRESS). SATA also helps to encounter the problems of overlapping schedule. It also provides reports on lecturers lecturing hours or known as ATA according to faculty and programs. With the existence of the reports the task to create timetable become more efficient. This indirectly help the organization to reduce cost, improve time management, and overcome lost of data. In short, SATA were build to help Timetable Schedule Committee to provide accurate information about their lecture hours, and related classes.

SATA is being fully utilized and the result is tremendous and very effective in helping the Timetable Management Team at UiTM Terengganu. SATA's usage is being popularized to UiTM Kelantan, the Business Faculty of UiTM Malaysia and UiTM Negeri Sembilan. SATA has a huge potential to be adopted by other UiTM branches and other institutes of higher learning as well.

## II. LITERATURE REVIEW

Timetabling is defined as the allocation, subject to constraints, of given resources to objects being placed in space-time in such a way as to satisfy as nearly as possible a set of desirable objectives. [2]

Managers, consultants and academics alike understand the variety of business and operational challenges facing enterprises today. Organizations are required to do more with less with regard to capital resources and people resources. The seemingly simple need to get more out of people has had a tremendous impact on employee workloads – and organizations are scrambling to address the fallout of these workloads. [3]

With all of the different tools available today, or perhaps because of that breadth of tools, schedule management in the enterprise is typically a mess. Why? Some of the reasons are simple, and have to do with human preferences (for both technology-based and manual scheduling tools). Some of it has to do with legacy or home grown scheduling systems and tools that are not scalable or have outlived their usefulness. And some of it has to do with the array of options that face enterprise workers every day. [3]

The following table illustrates just a few of the many methods people use to plan, record, and otherwise schedule their own time or the time and availability of other enterprise resources: [3]

Manual Scheduling	Group Messaging and Calendaring	Specialized Service Scheduling
<ul style="list-style-type: none"> <li>• Paper</li> <li>• Phone</li> <li>• E-Mail</li> <li>• Personal Information Manager (PLM's) e.g Palm Pilot, etc.</li> </ul>	<ul style="list-style-type: none"> <li>• Lotus Notes /Domino</li> <li>• Microsoft/ Outlook/ Exchange</li> <li>• Oracle Calendar/ Collaborative Suite</li> </ul>	<ul style="list-style-type: none"> <li>• Facilities schedulers</li> <li>• A/V bridge, gateway, and/or gatekeeper schedulers</li> <li>• Web conferencing service schedulers</li> </ul>

Fig.1 Schedule Methods

Across the organization, individuals, groups, and divisions frequently use different tools, with varying degrees of interoperability. This results in multiple scheduling methodologies that introduce inefficiencies. Schedule information is retained in a variety of locations and types of databases, without the ability to gather a comprehensive, unified view. Various tools may only support the scheduling of pieces of the overall infrastructure, not the complete set of elements that need to be scheduled. [3]

Most of the approaches that have been applied to solve the timetabling problems focus on the construction of the timetable as a static process. In real world, the timetabling

problems are dynamic and open problems since the initial timetable is not fixed and it is required to be changed as the constraints or assumptions, on which the timetable is based on, are changed or became invalid. [4]

Solving a real-world timetabling problem manually often requires a significant amount of time, sometimes several days or even weeks [5]

Constraints alter as unexpected events occur, such as adding or deleting resources which are subjects or lecturers, and as the changes of the user demands put on to the scheduling system.

During the last thirty years, many contributions related to the timetabling problem have appeared and it will probably continue with the same rate for years. This could be due to the fact that timetabling problems are often over-constrained, dynamic, and optimization criteria are hard to define. Different techniques have and are being applied to solve the static timetabling problem, including graph coloring [6], integer programming [7], simulated annealing [8], tabu search [9], genetic algorithms [10], and constraint logic programming [11]

In order to manage a rapid growth of academic activities in a university, an efficient and flexible timetabling must be developed. Fig.2 shows a typical university timetabling processes. Timetabling is thus an ongoing and continuous process. The problem of updating timetables in the most effective way, when the constraints or assumption on which they are based are changed or became invalid, is one that is receiving increasing attention amongst researchers and practitioners. [4]

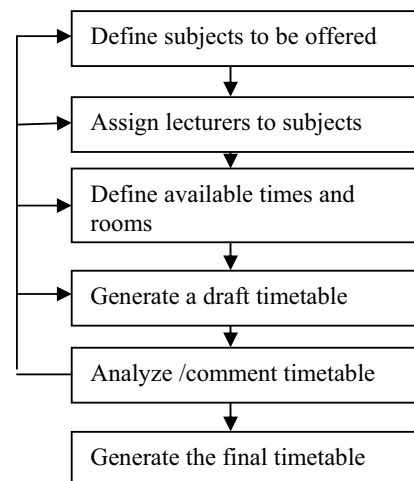


Fig. 2 A Typical University Timetabling Process

## III. UNIVERSITY TECHNOLOGY MARA (UiTM) SCHEDULING BUSINESS PROCESS

The UiTMT scheduling process starts with the timetable committee entering ATA into SATA. ATA (*Amanah Tugas Akademik*) describes the allocation of courses, student groups

and number of lectures to each lecturer. Fig.3 shows the timetable preparation workflow.

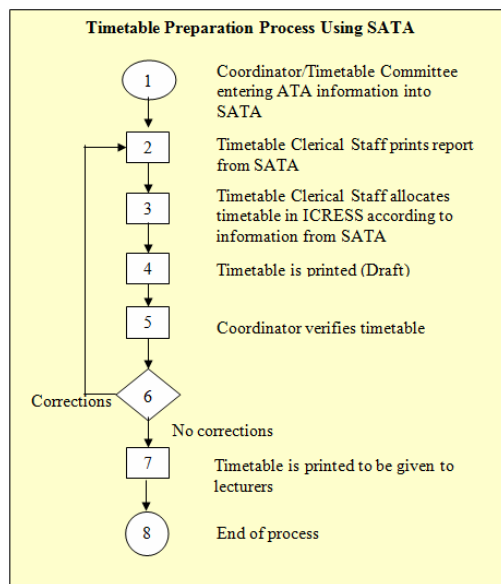


Fig. 3: Timetable Preparation Process Using SATA

Before SATA was used, the timetable preparation process was done by lectures from each faculty. The lecturers would first decide the ATA and then try to allocate the ATA in ICESS. The process of allocating the ATA was long and time consuming. The management wanted the clerical staff to take over this process. However, the clerical staff did not understand the ATA requirements of the various faculties. SATA was then developed to assist with this process. Using SATA, the timetable committee which consists of lecturers from each faculty or field of study would first define the ATA for lecturers from their own faculty, and then enter the information into SATA. SATA organizes the information into such a way that it is easy to be comprehended by the clerical staff. The clerical staff would then allocate the ATA using ICESS according to the information from SATA.

#### IV. DEVELOPMENT

SATA was developed using Active Server Pages (ASP) and XHTML. JavaScript and VBScript were used as the server side scripting and Microsoft Access as its database. The development tools used were Macromedia Dreamweaver MX 2004 and MySQLYog.

The initial version of SATA was developed in July 2005. This version was put to test during the timetable preparation of the December 2005 – May 2006 semester timetable. The trial run revealed a number of weaknesses and several changes and modifications were made to SATA. The second version of SATA was fully operational during the July 2006 – November 2006 semester timetable preparation. For the December 2007 – May 2008 semester timetable preparation, a newer version of SATA which was more user-friendly was used.

#### V. SATA

SATA users can be categorized into three categories which are the system administrator, timetable committee and the lecturers. SATA can be accessed by using a login id and a password provided by the system administrator as shown in Fig.4 below.

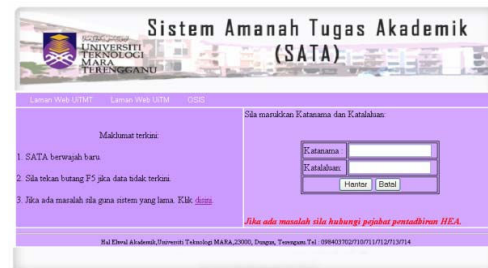


Fig. 4: System Log In

Different categories of users have different access level to the system. The system administrator has full access to all functions in SATA. The timetable committee is given the authority to add and update ATA information and to print out reports. Lecturers are only allowed to view their ATA and make changes to their personal information. After successful login into the system, SATA provides the users with the following functions:

- i) Data Modification
- ii) ATA Information Entry
- iii) System Administration
- iv) Report Printing

This is illustrated in Fig.5.



Fig. 5: Functions Provided by SATA

##### A. ATA Information Entry

The main function in SATA is the *ATA Information Entry*. This function can only be used by the system administrator and the timetable committee. To enter the ATA for a lecturer, the user will need to select the lecturer's name from a list. The system will then ask the user to enter the type of ATA for the lecturer as shown in Fig.6.



Fig. 6: Choosing ATA Type



Fig. 8: Grouped ATA without blocking

There are three types of ATA entry:

- i) Normal ATA,
- ii) Grouped ATA without blocking
- iii) Grouped ATA with blocking.

After selecting the type of ATA, the user will then need to enter information pertaining to the lecturer and the ATA associated with him. The information needed is the course code, program code, group code, room code, type of lecture, number of students for each lecture and contact hours. Different type of ATA will result in different screens used in order to enter the above information. Fig. 7, 8 and 9 illustrates the screens used for the different ATA types.

The ATA information entry for normal ATA is fairly easy and straightforward. The user simply selects the required course code, program code, group code, room code, type of lecture, number of students for each lecture and contact hours as illustrated in Fig.7.



Fig. 7: Normal ATA Information Entry Screen

Fig.8 shows the information entry screen for grouped ATA without blocking. What it means here is that the lecturer teaches several groups at a particular time. Here in Fig.7 it shows that the lecturer Ab. Aziz Ismail teaches course CTU151 for student groups ACDIAA, ACDIAB and ACDIAC of program AC110.

Fig.9 shows the information entry screen for grouped ATA with blocking. This kind of ATA is normally used for courses that have pre-requisites. For example, the course CSC125 is pre-requisite for course CSC175. Blocking course CSC125 with CS175 means that these two courses will run at the same time. For example, lecturer Ab. Aziz Ismail teaches course CSC175 for student group CSD2KA. At the same time, another lecturer teaches CSC125 for student group CSD1KA. Any student from group CSD2KA who failed CSC125 can re-sit the course while his other friends continue with CSC175.



Fig. 9: Grouped ATA with blocking

### B. Report Printing

After all the ATA for the lecturers have been entered into SATA, the timetable committee will then print out the information. SATA organizes the information in such a way that it is easier for the committee to comprehend the ATAs and enter it into ICESS. SATA allows two kinds of report to be printed out

- i) According to the field of study
- ii) According to the program.

Fig.10 and Fig.11 shows the report according to field of study and program respectively.

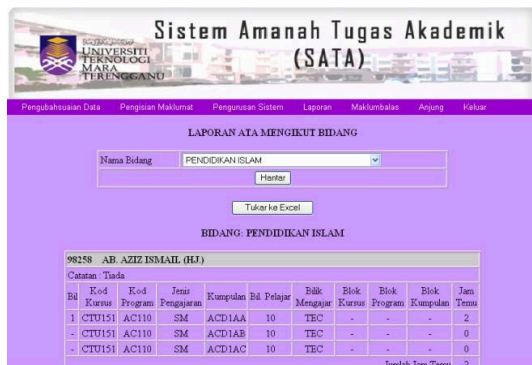


Fig.10: Report According to Field of Study



Fig.11: Report According to Program

VI. RESULTS

SATA has been used as a scheduling support system since July 2006 and the results have been encouraging. The use of SATA has resulted in the following:

- i) improved efficiency in the timetabling process,
- ii) increase in customer satisfaction,
- iii) savings in terms of operational costs, time and work process.

Before SATA existed, the timetabling process was carried out by 56 lecturers (4 lecturers each from 14 faculty or field of study). It took more than a month to complete the timetable. With the use of SATA, the timetabling process is done by only 6 clerical staffs and is completed within 2 weeks upon obtaining the ATA from the timetable committee.

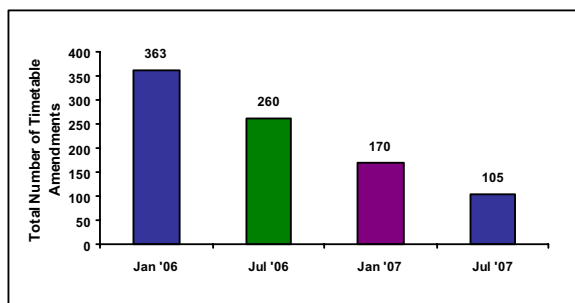


Fig.12: Number of Timetable Amendments

Fig.12 shows the total number of timetable amendments made to the timetable after it has been given to the lecturers. The graph shows a significant decline in the number of amendments made. This shows that SATA helps to reduce errors and misunderstandings during the process of creating timetables. Fewer amendments made to the timetable also means less paper is wasted in reprinting the corrected timetables. It also translates to less time spent on redoing the timetable.

The use of SATA has tremendously helped in the timetabling process and it has been put into UiTM Terengganu's Quality Operations Procedure. It has the potential to be commercialized and to be used by any institute of higher learning relating. It has also won a bronze medal in the 2008 Invention, Innovation and Design competition (IID'08) and has been submitted to be patented in October 2008. Future works on this system includes using AJAX technology to improve and enhance the user experience in using SATA.

REFERENCES

- [1] Cambazard, H., F. Demazeau, N. Jussien, and P. David, "Interactively Solving School Timetabling Problems Using Extensions of Constraint Programming". Practice and Theory of Automated Timetabling V. LNCS, Springer-Verlag, 190-207, 2004.
- [2] Md. Golam Yeasin, and Ahamad Tajudin Khader, "Local Search Algorithms for Examination Timetabling", National Conference on Programming Sciences (Atur'05), Kuala Lumpur, 1 December 2005.
- [3] Alan D. Greenberg, "The Business Case for Enterprise Conference Scheduling", for Wainhouse Research, March 2003. Retrieved December 16 2003, H.323 Forum, from <http://www.h323forum.org>
- [4] Hany Alashwal and Saaai Deris, "Dynamic Timetabling Using Reactive Constraint Agents", *Technology Journal*, 46(D), Faculty of Computer Science and Information Systems, Universiti Teknologi Malaysia, Skudai, Johor, Malaysia, 45-60, Jun 2007.
- [5] Tim, B. C. and J. H. Kingston, "The Complexity of Timetable Construction Problems", Proceedings of the 1st International Conference on Practice and Theory of Automated Timetabling (PATAT), LNCS 1153. Springer-Verlag. 283-295, 1995.
- [6] Miner, S., S. Elmohamed, and H. W. Yau, "Optimizing Timetabling Solutions Using Graph Coloring" NPAC REU Program, NPAC. Syracuse University. Syracuse, NY, 1995.
- [7] Dimopoulou, M. and P. Miliotis, "Theory and Methodology Implementation of a University Course and Examination Timetabling System", *European Journal of Operational Research*, 130: 202-213, 2001.
- [8] Melicio, F., P. Caldeira, and A. Rosa, "Solving the Timetabling Problem with Simulated Annealing", Proc. First Int. Conf. on Enterprise Information Systems, ICEIS' 99 - Setúbal, 272-279, 1999.
- [9] Schaerf, A., "Tabu Search Techniques for Large High-school Timetabling Problems", Proceedings of the Fourteenth National Conference on Artificial Intelligence, Portland, Oregon, 363-368, 1996.
- [10] Colomi, A., M. Dorigo, and V. Maniezzo, "Genetic Algorithms: A New Approach to the Time Table Problem", NATO-ASI School on Combinatorial Optimization, Ankara, Turkey, 1990.
- [11] Azevedo, F. and P. Barahona, "Timetabling in Constraint Logic Programming", Proceedings of 2<sup>nd</sup> World Congress on Expert Systems, Estoril, Portugal, 1994.