Open-Source Academic Timetabling System: FET Implementation at TATI University College

S. Fahmy, I. Ruhaida, N. Akhyari, I, Nazri

Faculty of Computer, Media & Technology, TATI University College, 24000 Terengganu, MALAYSIA.

Fakulti Teknologi Maklumat dan Sains Komputer, University College Bestari, 22100 Bandar Permasai, MALAYSIA.

ARTICLE INFO
Article history:
Received 20 November 2013
Received in revised form 24 January 2014
Accepted 29 January 2014
Available online 5 April 2014

Keywords:
Open-Source Application, FET, Timetabling, Malaysia.

ABSTRACT
Scheduling is a classic problem faced by many organizations around the world and has sparked much interest from industry and academia alike due to the uniqueness of each problem and the list of possible solutions. This work is highly motivated by the lack of computer system to support the preparation of academic timetable at the Faculty of Computer, Media and Technology Management, TATI University College. The current process (manual) is time-consuming, error-prone and results in an uneven distribution of workloads for both lecturers and student groups. This paper proposes the use of FET, an open-source computer application, to automatically generate academic timetable. The implementation involves four phases: Data Collection, Data Input, Activity Creation and Timetable Generation. Results indicate that allocation of time/venue is more efficient and it is also noted that the distribution of hours is more even amongst lecturers and student groups. It is hoped that this work has minimized, if not eliminate, the scheduling problem faced by TATI University College in particular and other academic institutions in general.


INTRODUCTION

The utilization of Information and Communication Technology (ICT) in organizations has made the task of scheduling more manageable and efficient. Scheduling is an important process for academic institutions such as school, college and university in order to manage organizational resources effectively. However, due to the high number of constraints that are involved, many institutions face a challenging task before an academic timetable can be finalized.

This work is highly motivated by the lack of computer system to support the preparation of academic timetable at the Faculty of Computer, Media and Technology Management, TATI University College (FCMT)(Fahmy et al., 2012). Based on initial observations, this problem leads to several issues:

i. The current process is too time-consuming and tedious in nature. Time spent for this process could be better spent for other academic activities such as lecture and laboratory sessions.

ii. The (manually prepared) timetable has several drawbacks such as overlapping classes and double-booking of venues. When these occur, a considerable amount of time is needed to rectify the situation.

iii. Uneven or unjust distribution of classloads for both lecturer and student groups, resulting in unnecessary cramped schedule for some of lecturers and students.

Since intake is projected to increase on a yearly basis, there is a need for this process to be automated in order to save time and minimizing (if not eliminating) human error.As such, this work aims to identify available computer applications in the market and to demonstrate the feasibility of the application to support the preparation of academic timetable at FCMT. The application should enable the automatic generation of timetable at the beginning of each academic semester.

2. Computerized Timetabling Applications:

This section reviews several timetabling systems in order to identify the most appropriate application for FCMT. This section starts with a definition of timetabling, followed by research efforts in the field, classification
of timetabling applications and finally an overview of FET.

Definition of Timetabling:

Timetabling can be defined as “the problem of scheduling a set of events (exams, lecturers, classes) to a specific time slots” (Ross et al., 1996). Wren (1996) describes timetabling as “the problem of placing certain resources, subject to constraints, into a limited number of time slots”. Scheduling is also seen as “is the process of assigning tasks to a set of resources”.

The authors would like to point out that there are contradicting views of the terms ‘timetabling’ and ‘scheduling’ found in literature. For example, Wren emphasized that scheduling aims “to minimize the total cost of resources used”, while timetabling “tries to achieve the desirable objectives as nearly as possible” (Wren, 1996). Carter points out that timetabling decides “upon the time when events will take place”, but does not usually involve “the allocation of resources in the way that scheduling often does” (Carter and Johnson, 2001).

We believe that contrasting views on the meanings of timetabling and scheduling is immaterial in this work, and definitions by Ross (1996) and Wren (1996) will be used to refer to timetabling in this paper. Timetabling problems exist in a variety of domains including education (allocation of class-venue), healthcare institutions (allocation of patient-doctor), transportation (allocation of routes) and sports (allocation of match-venues).

Research Efforts in Computerized Timetabling:

Works in the field of computerized timetabling include Schaerf and Di Gaspero (2001), who discuss’ measurability’ and ‘reproducibility’ in university timetabling including practices that contribute to the improvement of these two aspects. Different issues in computer-aided applications including problems, requirements, data representations and mathematical models are discussed by Bardadym (1996). Solution methods from the 1960s to the 1990s were overviewed covering heuristics, meta-heuristics and algorithmic tools for integration in decision support systems. Results indicate that meta-heuristics and interactive timetabling are seen as the new wave of computer-aided timetabling systems.

Classification of Timetabling Applications:

Timetabling applications can be broadly categorized into three namely Commercial, Freeware and Open-Source. Commercial applications are 'software that are designed and developed for sale to the general public'. Timetabling applications in this category include Wise (2012), TT Maker, Edval (2012), Event map, Framing, and Keith Johnson’s Timetabler. Freeware is ‘a piece of software that can be downloaded, passed around, and distributed without any initial payment’. Freeware is usually copyrighted and examples include minor program updates and games. List of timetabling software in this category include Lantivy (2012), SchoolTimetable.edu, and Mimosa Small School Edition (2012). The term Open-Source Software has been used interchangeably with freeware. The main difference between the two is open-source ‘grants access to the source code’. In addition, the distribution terms of open-source software includes free redistribution; not specific to a product; and not restricting other software; which are usually not the terms of freeware. List of Open-Source timetabling software include FET (2012), Virtual Paragon, Droolsolver and Timefinder (2012).

FET:

FET is an open-source application to support the automatic generation of timetable for an academic institution such as school or university. It is licensed under GNU/GPL and uses an efficient heuristic algorithm based on recursive swapping of activities to solve difficult timetables. It is able to consider all groups (classes/courses/students) and able to display reports in XHTML format. FET can be localized to numerous languages including English, Arabic, German, Malay and Dutch. It includes an automatic generation algorithm, allowing semi-automating or manual allocation of classes. It also has a flexible modular XML format for the input file, allowing editing with an XML editor or other applications. Generated timetable can be exported into XML and HTML formats. Students’ structures are flexible and organized into set of years, groups and subgroups. FET allows overlapping of years and groups. The algorithm swaps activities recursively to make space for new activities. A complicated timetable can be usually solved within the hour while only a few minutes is needed for simpler timetables (in some cases, a matter of seconds). FET can be freely used, copied, modified and redistributed. It is platform independent, allowing execution on GNU/Linux, Windows and Mac. In summary, FET is an excellent application that can easily compete with some of the well-known commercial timetabling applications.

3. Framework:

This section presents the generic framework for the implementation of FET. There are four phases involved: Data Collection, Data Input, Activity Creation and Timetable Generation (Fig. 1).
**Fig. 1: FET Implementation Framework**

**Data Collection:**
In the first phase, necessary data for the preparation of academic timetable are obtained from various entities such as the University and Faculty. These information include venues (locations of lectures/ laboratories/ tutorial/ other academic session will take place including lecture halls, labs, workshops, etc); subjects (list of subjects offered in a given semester); and grouping (list of students grouped into a particular category such as academic programs, intake, etc).

**Data Input:**
The next phase, data is stored into the FET system/ database. Additional information required in this process includes details of lecturer/ instructor; academic hours per day; and academic days per week.

**Activity Creation:**
Next, a set of activities are created amongst different student groups/ lecturers and venues. Rules are also created in this phase for example a particular venue for a session or a blackout period for lecturers/ student groups.

**Timetable Generation:**
Finally, the timetable is generated and locked. The generation process is successful if there is no error reported by FET. If error is detected, execution is halted and the activities and/ or data must be revised. Typical errors include too few hours/ days allocated for an activity, demand for a particular venue, or too little hours allocated for a group of lecturers/ students.

4. **FET Implementation:**
This section presents the implementation of FET to test its feasibility as a tool to support automatic timetable generation at FCMT. Data were collected from the Academic Affairs Department (list of students), Property Development Department (list of venues for academic sessions), Faculty (list of lecturers and subjects offered). Scope of implementation is 8 academic programs, 34 academic venues, 69 subjects, 32 student groups and 31 lecturers. All lecturers, programs and students are from FCMT.

Fig. 2a and 2b illustrate data input process for lecturers and subjects respectively. Data input took approximately 7 days involving 2 officers. Fig. 2a illustrates the process of adding a new lecturer into the database. List of lecturers can be created, updated, removed and sorted in the system.
Fig. 2a: Data Input in FET (Lecturer)

Fig. 2b. illustrates the process of adding a new subject into the database. Similar to Lecturer (Fig. 2a.) subjects can be created, updated, removed and sorted.

Fig. 2b: Data Input in FET (Subject)

Fig. 3a and 3b illustrate the activity creation process involving data configuration for lecturers, student groups, subjects and academic hours (per week). This process required an additional 3 days and was headed by the Head of Academic Departments (HoDs).

Fig. 3a. illustrates the process of creating a new activity for ‘Syahrul Fahmy’ (lecturer), who teaches BNS 2123 Visual Programming (subject) to 2 BCBM and 1 BCNS (student groups). Class hours can also be set in this process, and if the class are consecutive or split.
Fig. 3a: New Activity Creation in FET

Fig. 3b. illustrates the process of updating/ removing an activity. Activities can be filtered according to lecturer, student group, subject or activity.

Fig. 3b: Activity Information in FET

Once data for all programs have been configured, the timetable was generated and reviewed by the Dean and HoDs. Few activities were revised before an accepted timetable was generated. The generation took less than a day. Fig. 4 illustrates the timetable generated by FET listing subject, lecturer, venue and time.

Fig. 4. illustrates the timetable generated for a particular student group. For example, this group has BCS1323 Computer Organization lecture by Syahrul Fahmy at L18 on Sunday, 8.00–11.00 am; BCS1313 Information Technology and System lecture by Zainuddin at L17 on Monday, 11.00–1.00 pm; and BCS1213 Programming Techniques lecture by Suriati at L12 on Wednesday, 8.00–10.00 am.
Fig. 4: Timetable Generated by FET

The FET-generated timetable was compared to the manually prepared timetable for Semester 1 and 2, intake 2012. Results indicated that not only the hours are distributed more evenly, but also the allocation of venues is more efficient using FET.

Conclusion and Future Work:

This paper has proposed the use of FET, an open-source computer system, to automatically generate academic timetable at the Faculty of Computer, Media and Technology Management, TATI University College. The implementation was carried out in four phases: Data Collection, Data Input, Activity Creation and Timetable Generation. Results show that allocation of time/venue is more efficient than the manually prepared timetable. It is also noted that the distribution of hours is more even amongst lecturers and student groups. It is hoped that this work has minimized, if not eliminate, the scheduling problem faced by TATI University College in particular and other academic institutions in general.

Future work includes the implementation of FET throughout the university, involving more academic programs, lecturers and venues into the system.

REFERENCES


