

A Novel Approach To Enhance a University Information System

Afif Jadalla ALmghawish, PhD
 Dept. of Computer Science
 AL-Zaytoonah University of Jordan
 Amman, Jordan

Alexandre F. Ossyka, PhD
 Dept. of Computer Science
 AL-Zaytoonah University of Jordan
 Amman, Jordan

Issa S. Ottoum , PhD
 Dept. of Computer Networking
 AL-Zaytoonah University of Jordan
 Amman, Jordan

Abstract— A new approach is presented for helping a student to go ahead in his\her progress to reach its final aim “Obtaining a Bachelor Degree”. This approach based on an algorithm that enforces and enables a student to follow the curriculum (Advising Plan) when he\she wants to register subjects for a next semester. The proposed algorithm relies upon many factors, such as specifying the subject(s) level, distribution subjects on time table for five days a week, and others. The proposed algorithm will enhance the efficiency of the timetable for both students and administration. It gives a student space for choosing the subjects which in turn enforces a student to go ahead with respect to his study plan that will affect his success in reaching a final goal. For administration it is a good idea to make a student commits his self to respect his suggested curriculum.

Keywords- Curriculum; Timetable; Course Level; Course Distribution; University Information System (UIS).

I. INTRODUCTION.

To achieve a goal of enhancement a University Information System (UIS), some raw data must be existed:

1.1. Study plan

A study plan contains all courses that a student must be passed to obtain his\her Bachelor degree. The study plan is comprised of the following blocks:

1. University Block which contains university courses: mandatory and elected.
2. Faculty Block that contains faculty courses: mandatory and elected.
3. Specialty (Department) Block that contains core courses: mandatory, elected, and supported.
4. Free Block that allows a student to choose a course from any other blocks of his\her study plan. (The chosen course is of course from elected blocks).

The overall structure of a study plan can be viewed as depicted in this figure:

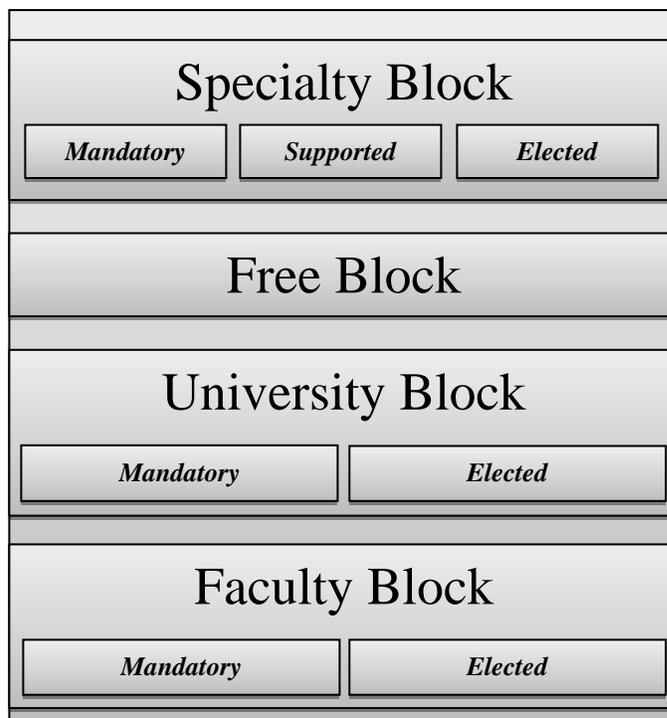


Figure 1. A structure of study plan.

1.2. Curriculum (Advising plan)

A curriculum is a plan that directs and helps students to choose their courses for every semester through the studying process. This curriculum is constructed based on study plan information and the level of all courses contained in the study plan. So, courses are distributed in some order relies on academic year and semester.

The following figure shows the curriculum structure of the fourth academic year study plan:

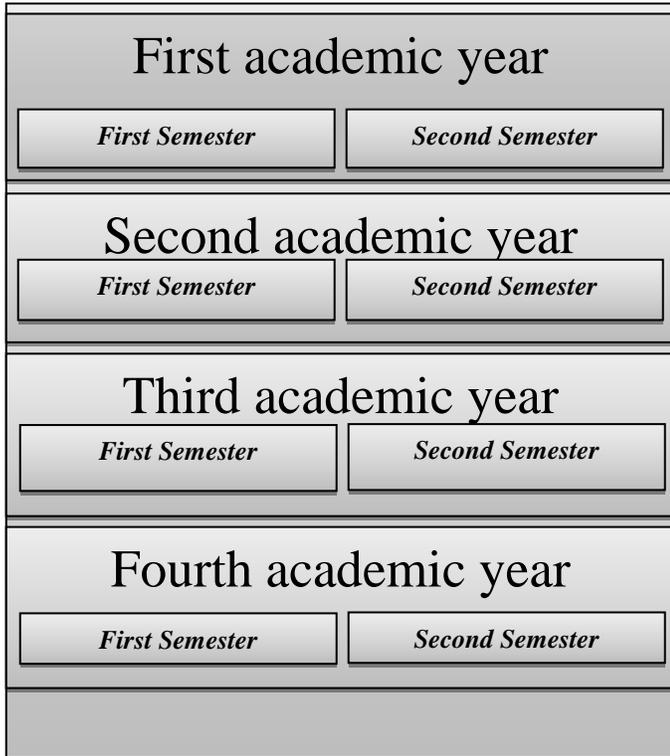


Figure2. A curriculum structure of a study plan.

1.3. Database

A database is a collection of related data for the under studied University Information System which contains data about students, courses, class rooms, and other related data. The files of the database are shown in figure 3:

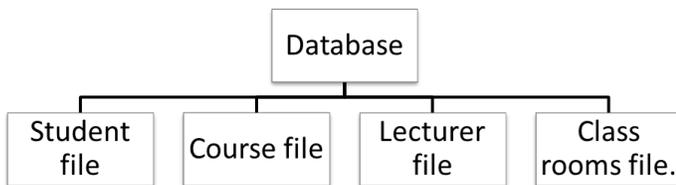


Figure3. A related database.

II. RELATED WORKS

The academic success of most universities based upon many factors, one of them is the modern UIS they have. In general UIS contains huge data that concerns student needs such as, study plans, university specialties, timetable, and other

related things. It enables student to register courses for next semester, knows university news, and so on.[1,2,3,4]. UIS also collects valid applications of students who are attracted to the university from all over the world. [5, 6, 7].

As a part of UIS, the timetabling problem has been the main subject of many researchers.

The constructing of a timetable as an actual problem has been addressed in many research works with different approaches and methods. Francisco Azevedo, and Pedro Barahona [15] proposed constraint programming technique for this problem. Another approach to deal with such a problem called a procedure FORECAST was studied in a research work done by Afif Mghawish [16]. Zhang I. and Lau S. In their paper [9] focused on developing a constraint satisfaction problem model for a university timetabling problem. A solution of a constraint satisfaction problem is a consistent assignment of all variables to values in such a way that all constraints are satisfied, Henz and Wurtz [19] propose the use of consecutive disjunction and reified constrains for course timetabling in Oz.Schaerd in his paper[18] classified timetabling into School Timing, in which there cannot be overlaps between courses having common students.

III. THE PROPOSED ALGORITHM

The enhancement of the University Information System relies upon many factors:

1. Specifying the level of courses in study plan, by defining for every course its prerequisite course(s).
2. Distribution of courses on time table in such a way that allows students to register at least twelve credit hours in a week (five days a week).
3. Restrictions that must be taken into account:
 - o The maximum number of allowed credit hours in a week must not exceed eighteen credit hours (18 credit hours).
 - o The number of proposed subjects in each time slot must not exceed the number of available class rooms.
 - o Courses must be distributed by two week times:
 1. Some courses will propose to be scheduled at three days: Sunday, Tuesday, and Thursday, (week time1).
 2. Other courses will propose to be scheduled at the rest two days: Monday and Wednesday, (week time2).
 - 3. Regardless of a week time, every course must have three credit hours a week.
 - 4. In different week times, there must be different courses.
 - 5. Each course must have no more than one section in general; but there are cases for

which a need for more than one section is required to a course.

The proposed algorithm will take care of all the factors mentioned above and some other details that make it more consistent, reliable, efficient, and proper for both student and administration.

The main steps of the proposed algorithm are as followed:

Step1. Follow this general rule when applying the next two steps.

“A course for each academic year and each specified semester must be opened at that semester not at other”.

For example, “011” the course(s) that related to this category of level division must be opened in first academic year at first semester, and so on for other categories of level divisions (see table1).

Of course, there are some exceptions that may be appeared (some of them are mentioned above in this paper). These exceptions must be treated by regarding every case separately from others.

Step2. Divide the courses in the study plan into proper required levels by academic year and semesters that comprise that academic year.

Firstly, the code ”0xy” is used to denote academic year and its related semester, where” x“ denotes the academic year and ”y” denotes the related semester.

Secondly, the use of the code “0xy” helps in obtaining the level division of courses as illustrated below:

First academic year

“01”- means, level1 for first academic year;

“011”- means, first semester of first academic year;

Second academic year

“012”- means, second semester of first academic year;

“02”- means, level1 for second academic year;

“021”- means, first semester of second academic year;

“022”- means, second semester of second academic year;

Third academic year

“03”- means, level1 for third academic year;

“031”- means, first semester of third academic year;

“032”- means, second semester of third academic year;

Fourth academic year

“04”- means, level1 for fourth academic year;

“041”- means, first semester of fourth academic year;

“042”- means, second semester of fourth academic year;

The obtained results of division courses in study plan into appropriate levels are summarized in table1.

TABLE1. LEVELS OF COURSES IN STUDY PLAN.

| Courses of first semester | Courses of second semester |
|---------------------------|----------------------------|
| 011 | 012 |
| 011 | 012 |
| 011 | 012 |
| | |
| 021 | 022 |
| 021 | 022 |
| | |
| 031 | 032 |
| 031 | 032 |
| | |
| 041 | 0421 |
| 041 | 0421 |
| | |

Step3. Arrange all courses by their semester level regardless of academic year to facilitate distribution in timetable, as shown in this table:

TABLE2. ARRANGEMENT COURSES LEVELS BY SEMESTER

| Academic Year / Semester | 01 | | 02 | | 03 | | 04 | |
|---|-------------|-----|---------|-----|---------|-----|---------|-----|
| | 011 | 012 | 021 | 022 | 031 | 032 | 041 | 042 |
| Courses | | | | | | | | |
| Course1 | √ | √ | | | | | | |
| Course2 | | | √ | | | | | |
| Course3 | | | | | √ | | | |
| Course4 | | | | | | | √ | √ |
| | | | | | | | | |
| Course(n) | | | | | | √ | | |
| Number of courses for every semester | n1 | n2 | m1 | m2 | q1 | q2 | s1 | s2 |
| Number of courses for an academic year | n=n1+n2 | | m=m1+m2 | | q=q1+q2 | | s=s1+s2 | |
| Total number of courses in a study plan | $T=n+m+q+s$ | | | | | | | |

To get more evident and clarity of this step in a proposed algorithm, the data in the above table2 must be illustrated. This can be done by naming some courses within the study plan according to the course level as shown in the below table3.

TABLE3. SOME COURSES CORRESPONDING TO THEIR LEVELS.

| Course level | Name of courses | Course level | Name of courses |
|--------------|---|--------------|---|
| 011 | <ul style="list-style-type: none"> • Calculas1 • Computer concepts • Discrete structure | 012 | <ul style="list-style-type: none"> • Calculas2 • Logic design • Programming language1 |
| 021 | <ul style="list-style-type: none"> • Linear Algebra • Programming language2 • Web technology | 022 | <ul style="list-style-type: none"> • Numerical analysis1 • Data structure • Advanced programming language |
| 031 | <ul style="list-style-type: none"> • Numerical analysis2 • Database1 • Computer architecture | 032 | <ul style="list-style-type: none"> • Linear differential equations • System analysis and design • Advance computer network |
| 041 | <ul style="list-style-type: none"> • Topology • Database2 • Computer security | 042 | <ul style="list-style-type: none"> • Algorithms design and analysis • Graduation project |

Step4. Distribute courses on a time table based on the following information:

- Levels of courses. This information is captured from table1.
- A proper time for students and lecturers. This point partially depends on the number of buses reserved for each time slot, and other objective circumstances.
- Different courses at different days(week time) i.e. courses at Sunday, Tuesday, Thursday; and other courses at Monday, Wednesday as mentioned before in this paper
- One section per one course unless there are other requirements.
- The number of credit hours allowed for a student to register at a semester is in the range of “12<=credit hours<=18”.
- The number of courses at each time slot must not exceed the number of available class rooms.
- Courses that have levels “011”, “021”, “031”, “041”must be opened in the first semester of every academic year, and courses that have levels “012”, “022”, “032”, “042”, must be opened in the second semester of each academic year (see table2, and table3).

This algorithm has the following advantages:

1. Simple to understand.

2. Give a balance among student desiring and administration restrictions.
3. Help students to get their goal (obtaining Bachelor degree) by putting them on a right way.
4. Increasing the performance of department, faculty, and lastly the university.
5. The more important one is its simplicity to implement.

IV. ANALYSIS AND IMPLEMENTATION

The proposed algorithm was applied to make a timetable for a faculty of Science and Information Technology at Alzaytoonah University of Jordan.

The implementation of this algorithm was done as follows:

1. Distribution the courses by their levels for six departments comprised the faculty of Science and Information Technology(Mathematics, Computer Science, Computer Information System, Software Engineering, Multimedia Systems, and Computer Networks).
2. Courses belong to any department in the faculty, so the levels of courses (011, 012, 021, 022, 031, 032, 041, and 042) may express them. For example,
3. 011- May be a course of mathematic, computer science, computer information systems, etc. And so on, for other defined levels.

TABLE4. DISTRIBUTION OF COURSES BY THEIR LEVELS ON WEEK TIME1

| Time \ Classrooms | 8-9 | 9-10 | 10-11 | 11-12 | 12-13 | 13-14 |
|-------------------|-----|------|-------|-------|-------|-------|
| Classroom(1) | 011 | 021 | 011 | 011 | 021 | 011 |
| Classroom(2) | 031 | 041 | 031 | 031 | 024 | 021 |
| Classroom(3) | | 021 | 021 | 031 | 024 | 031 |
| | | | | | | |
| Classroom(n) | 011 | 041 | 041 | 011 | 021 | 041 |

TABLE5.DISTRIBUTION OF COURSES BY THEIR LEVELS ON WEEK TIME2.

| Time \ Classrooms | 8-9:30 | 9:30-11 | 11-12:30 | 12:30-2 | 2-3:30 |
|-------------------|--------|---------|----------|---------|--------|
| Classroom(1) | 021 | 011 | 021 | 011 | 011 |
| Classroom(2) | 041 | 031 | 041 | 021 | 031 |
| Classroom(3) | 021 | 011 | 021 | 031 | 011 |
| | | | | | |
| Classroom(n) | 041 | 031 | 041 | 041 | 031 |

Analysing data in tables 4 and 5, we can show that:

-At each slot time the courses are of different levels and distributed in such a way that permit students to register at least 12 credit hours.

- The overlaps between courses having common students are cancelled

- Students whose precisely follow the curriculum cannot be register 3 consecutive courses due to a proper distribution of courses. This increases a student capability for more understanding his\her courses, because there must be a break of 1 hour (week time1) or 1:30 hours (week time2).

-This way is more preferable for administration, because it restricts a space that allows a student to choose his\her courses to be registered.

- The given option is just not a unique one, there are other options to distribute courses on a timetable, but this one is more efficient for both student and administration.

V. CONCLUSIONS

As known, one of the important goals of the University Information System is to help students in managing the registration process when they choose their courses for the next semester. The proposed algorithm gives students whose follow precisely the curriculum (advising plan) a right way in their progress to reach the final aim of the university studies “Bachelor Degree”, due to the imposed limitations , and chosen distribution option.

REFERENCES

- [1] [http:// www.bu.edu/reg](http://www.bu.edu/reg) (UIS of Boston University, USA)
- [2] <http://www.oibs.metu.edu.tr> (UIS of Middle East Technical University, Turkey)
- [3] <http://www.utdallas.edu> (UIS of University of Texas, USA)
- [4] <http://www.acs.utah.edu/student> (UIS of University of Utah, USA)
- [5] <http://www.sisweb.uccavis.edu> (UIS of University of California)
- [6] <http://www.registar.mit.edu> (UIS of MIT, USA)
- [7] <http://www.epgy.stanford.edu> (UIS of Stanford University, USA)
- [8] Wenhao Li, (2011) .A Scheduling Algorithm for the Distributed Student Registration System in Transaction-Intensive Environment. International Journal of Distance Education Technologies, 9(1), January – March, p.p. 72 - 85.
- [9] Zhang, L., Lau, S. (2005). Constructing university timetable using constraint satisfaction programming approach. Proceedings of the International Conference on Computational Intelligence for Modeling , Control and Automation and International Conference on Intelligent Agents , Web Technologies and Internet Commerce Vol-2 (CIMCA - IAWTIC'06), November 28 - 30 , p.55 - 60.
- [10] Ho Sheau Fen, ET. al. (2009) University Course Timetable Planning using Hybrid Particle Swam Optimization. GEC'09 Proceedings of the First ACM / SIGEVO Summit on Generic and Evolutionary Computation, NY, USA, p.p. 239 - 246.
- [11] Beck J.C., Davenport A.J., and Fox M.S. (1988) The ODO Project Towards a Unified Basis for Constraint-Directed Scheduling. International Journal of Scheduling, 1, p.p. 89 - 125.
- [12] <http://www.ttable.com> (software package TTABLE)
- [13] <http://timetabler.com/tt4windows> (software package TimeTabler4).
- [14] Ossyka, A. Mghawish, A. Ottom, E. (2005) Computer-Aided Students Registration System. The Second International Conference on Information Technology ICIT-2005 (p.p. 291 – 297), Amman, Jordan.
- [15] Francisco Azevedo and Pedro Barahona. Timetabling in constraint logic programming. In Proceedings of 2nd World Congress on Expert Systems, Estoril, Portugal, Jan 1994.
- [16] Developing New Features for a University Information System Dr. Afif J. Almgawish Journal “ Science Series Data Report ”, Vol.4, and No.12. Dec 2012, P.71 - 85.
- [17] Supporting Student Information System Validity. Afif J. ALmghawish, Alexandre F. Ossyka, European Journal of Scientific Research Vol.99 , No.2 April 2013.
- [18] Andrea Schaerf. A survey of automated timetabling. Artificial Intelligence Review, 13(2):87 - 127, 1999.
- [19] Martin Henz and Jörg Würtz. Using Oz for college timetabling. In Proceedings of the 1995 International Conference on the Practice and Theory of Automated Timetabling, Edinburg Scotland, Aug. 1995.