SOFTWARE ENGINEERING APPROACH FOR A TIMETABLE GENERATOR

Albert Chai Meng Fatt, Chia Wee Kee, Lee Chee Heong, Ng How Seng, Karen Ng Sor Har,
Puah Suet Ni, Alvis Yeo Kok Yong, Mark Yeo Soon Hock, and Edmond C. Prakash
School of Computer Engineering
Nanyang Technological University
Singapore – 639 798
Asprakash@ntu.edu.sg

Abstract: The objective of this research and development project is to create an online application. The application allows user to generate available timetable combinations for his subjects, with the purpose of letting a student have the ease and flexibility in choosing a schedule which best match his/her preferences. We report our experiences and the step-by-step analysis, design and implementation technique using a commercial CASE tool Rational Rose and the UML notation.

I. PROJECT MAIN REQUIREMENTS

Several timetabling systems has been used by NTU for academic claendar generation and planning for an optimum timetable without any clash. One such system in use today is the STARS[1] system at NTU. However, the existing system has not been analyzed properly, runs only on legacy systems, does not provide a mechanism for software maintenance, cannot be easily understood, and very difficult to plugin new algorithms. Hence this new software project using Software Engineering principles has been initiated. The main requirements of this software project are listed below:

- To authenticate the user using the online system.
- Display the subjects that the user is allowed to take.
- Allow the user to select the subjects.
- Generate the possible timetable combination.
- Track which combination is selected by the user.
- Print out the timetable.

Additional requirements include:

- Allow administrators to perform database management online.
- Allow for add drop of subjects.
- Track total academic units been registered.
- Determine number of vacancies for a particular timetable combination.

We use the Rational Rose CASE tool[2] and the UML notation[3,4] to capture the artefacts of our software system.

II. SOFTWARE SPECIFICATIONS

A. Initial design specifications

The timetable system receives username and password from the student via an online form. The username and password are received and validated by the system software. Base on the username and course, the system software generates the subjects that the user can register. An online form displays the subjects and allows the user to select them. Upon submission, the system will check if there are conflicts between the subject allocation and generate all the combination of timetable to the user. The system software displays the combination available to the user. The user will be allowed to select the combinations that he wants and submits the request to the system software. The system software will update the database to register the subjects that the user have chosen and allow the user to print the selected timetable. Administrator's responsibility is to maintain the database. The lecturers are able to view the size of an individual class.

B. First refinement of design specifications

The timetable system *receives* a username and password from the student. The username and password are *validated* by the system. Based on the username, the system is able to know which course the user is from, and *generates* the subjects which the user can register. An online form *displays* the subjects available to the user, and *allows* the user to *select* the subjects he wants to register for the semester. Upon submission of the form, the system first *resolves* the total AUs for the subjects he registered. The system *resolves* any conflicting time slots between subjects. Last, the system will *generate* all the possible combinations of timetable with the registered subjects. The system *displays* the combinations to the user and *allows* him to choose a combination. Upon submission of the combination request, the system *updates* the databases to *register* the combination chosen. The system *allows* the user to print the confirmed timetable. The administrator is to *maintain* the databases. The lecturers are able to *view* the size of respective class.

C. The Potential List

After refining the software specification, we isolated the potential classes and perform a grammatical parse to isolate the nouns and noun phrases. After isolation, each noun is categories and decision is made to decide whether to include the noun in the list or reject the noun.

0-7803-6355-8/00/$10.00©2000 IEEE

Potential List	Type	Confirm / Reject
timetablesystem	**things remembered or structure**	**confirmed (system itself)**
username	things remembered	rejected (criteria 1 & 2 apply but 3 fails)
password	things remembered	rejected (criteria 1 & 2 apply but 3 fails)
student	role or external entity	rejected (criteria 1 & 2 fail)
course	things remembered	rejected (criteria 1 & 2 apply but 3 fails)
subjects	**things remembered**	**confirmed (criteria 1, 2 & 3 apply)**
online form	**things remembered**	**confirmed (criteria 1, 2 & 3 apply)**
total AUs	things remembered	rejected (criteria 1 & 2 apply but 3 fails)
combinations	**things remembered or structure**	**confirmed (criteria 1, 2 & 3 apply)**
combination request	occurrence remembered	rejected (criteria 1 & 2 apply but 3 fails)
databases	**things remembered or structure**	**confirmed (criteria 1, 2 & 3 apply)**
confirmed timetable	**things remembered or structure**	**confirmed (criteria 1, 2 & 3 apply)**
administrator	role or external entity	rejected (criteria 1 & 2 fails)
lecturer	role or external entity	rejected (criteria 1 & 2 fails)

D. DFD for timetable generator

The diagram below show both the context and data flow diagram for the timetable generator module of the software.

Context Diagram

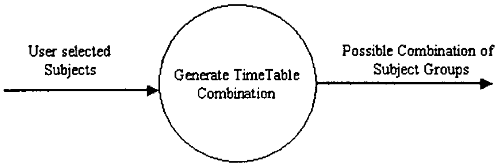


Figure 1. Context Diagram for the Time Table Software

Level 1 Data Flow Diagram

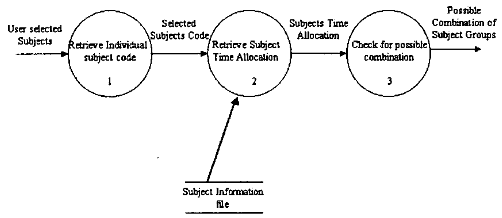


Figure 2. Level-1 Data Flow Diagram indicates the refinement of processes

E. Use Case Diagram

Based on the program specification, this is the use case diagram that describes the program modules and interaction. The following diagram is created using Rational Rose software.

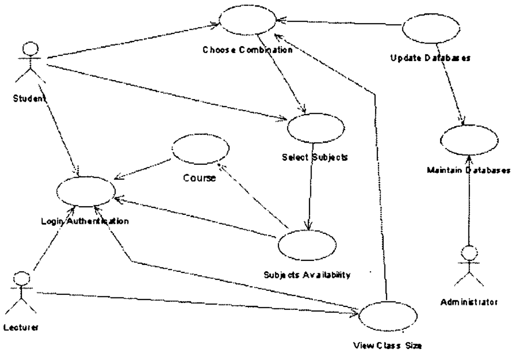


Figure 3. Use Case Diagram for the Time Table Software

F. Class Diagram

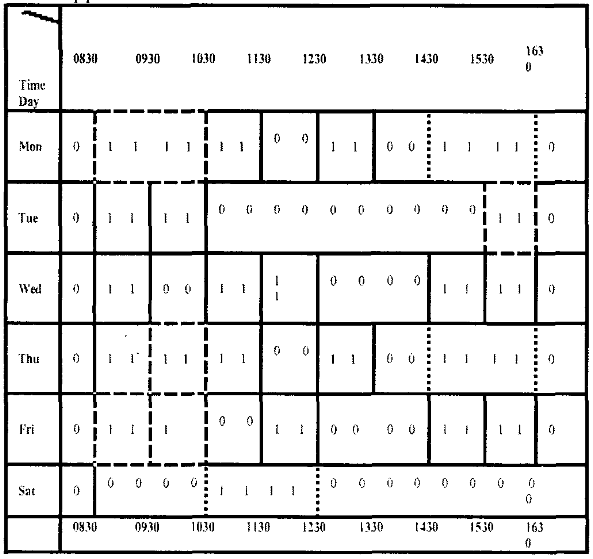
Base on the program specification, this is the class diagram that describes the program modules and interaction. The following diagram is created using rational rose software.

III. THE PROTOTYPE MODEL

We have adopted the Prototype Model for our project development. The sequence of events and the durations are illustrated in the diagram. The main reason in adopting this model is the uncertainty of algorithm implementation in the subjects combination. We thought of getting a workable algorithm before further improvisation on it. Moreover, this project deals a lot with client and server communication. We want to get the communication channel set up properly before transacting over this channel. Last but not least, we wish to give the user a chance to evaluate the prototype and make refinements accordingly.

IV. THE ALGORITHM

Instead of comparing every possible combination for detecting clash in the timetable using some efficent methods. One approach is described below.



Time Day	0830		0930		1030		1130		1230		1330		1430		1530		163 0	
Mon	0	1	1	1	1	1	1	0	0	1	1	0	0	1	1	1	1	0
Tue	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0
Wed	0	1	1	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0
Thu	0	1	1	1	1	1	1	0	0	1	1	0	0	1	1	1	1	0
Fri	0	1	1	1		0	0	1	1	0	0	0	0	1	1	1	1	0
Sat	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0 0	
	0830		0930		1030		1130		1230		1330		1430		1530		163 0	

Step 1.Assign Binary Code: A sample of our timetable is translated into binary data below. Each bit represents a time slot of 30 minutes. The binary representation covers from 8 a.m. to 10 p.m., 14 hours per day. In all, there are 28 binary bits representing each day.

Mon	0111	1110	0110	0111	1000	0000	0000
Tue	0111	1000	0000	0001	1000	0000	0000
Wed	0110	0111	1000	0111	1000	0000	0000
Thur	0111	1110	0110	0111	1000	0000	0000
Fri	0111	1001	1000	0111	1000	0000	0000
Sat	0000	0111	1000	0000	0000	0000	0000
Time	0800 - 1000	1000 – 1200	1200 – 1400	1400 – 1600	1600 – 1800	1800 – 2000	2000 - 2200

Step 2. Convert to HEX:
Monday
0111 1110 0110 0111 1000 0000 0000
=
Monday
7 E 6 7 8 0 0

Step 3. Comparison of schedule whether clashes, use **AND** function:
A Monday schedule of a one hour HRM from 1030 to 1130. Binary code would be:
0000 0110 0000 0000 0000 0000 0000
Current Monday time table schedule:
0111 1110 0110 0111 1000 0000 0000

Step 4. Check AND results: Performed **AND** function:
0000 [redacted] 0000 0000 0000 0000 0000
Step 5. Result (should be zero if no clashes): Clashes!

V. DATA DICTIONARY

The data dictionary is shown below as the list of tables for use in our application. The data fields are organized using techniques of normalization. More information on normalization techniques will be available in the final refinement. This database design is then implemented using Access 2000.

A. Table Name : tblStudent
Description: This table stores the student information

Fieldname	Data type	Description
Login (Key)	String	Login name same as matrix no
Password	String	Password
Name	String	Student name
Address	String	Student address
Course	String	Course the student is taking
Year	String	Year of the student
ICNo	String	IC No of the student
Sex	String	Sex of the student

B. Table Name : tblSubjectTaken
Description: This table stores the subject that a student has taken and clear

Fieldname	Data type	Description
Login	String	Login name
Subject Code	String	Subject code
NoAUs	Short Integer	Academic Units

C. Table Name : tblSubjectInfo
Description: This table stores the subjects information

Fieldname	Data type	Description
SubjectCode (Key)	String	Subject Code
SubjectName	String	Subject Description
NoAUs	Short Integer	Academic Units
PreRequisite	String	Pre-Requisite
CoRequisite	String	Co-Requisite
Course	String	Organized by which course
Description	String	General Description of subject
IsGE	Boolean	Is this subject GE?
NA	String	Departments prohibit to take this subject
ExamDate	String	Date of Examination
ExamTime	String	Time of Examination

D. Table Name : tblGroupInfo
Description: This table stores the groups information

Fieldname	Data type	Description
GroupCode(Key)	String	Group Code
GroupSize	Short integer	Size of Group
SubjectCode	String	Subject related to this group
Vacancies	Short Integer	No of vacancies left for this group

E. Table Name : tblSubjectAlloc
Description: This table store information about the schedules of the groups

Fieldname	Data type	Description
Subject Code	String	Subject Code
Type	String	Tut,Lec,Lab
Day	Short Integer	1-6
Period	Short Integer	1-20 Starting with 0800H and increment at 0.5 hours until 2100H
Venue	String	Location
GroupName	String	Group Code
StartWeek	String	Even, Odd, or All

F. Table Name : tblSubjectRegister
Description: This table stores the subjects that the user has currently register

Fieldname	Data type	Description
Login	String	Login name
Subject Code	String	Subject code
NoAUs	Short Integer	Academic Units

VI. RESULTS

We have completed a prototype of our time table system for our customer to review. We have shown our customer our prototype. The program flow is shown to the customer and they have verified that it is what they want. The customer also has requested for additional features. Currently, based on our documentation, we have started the implementation of the final product.

VII. CONCLUSION

We have followed a systematic approach for development of a large software system. The software system is currently being subject to further refinement. The next stage after detailed design is the final implementation, testing, quality analysis and finally deployment.

VIII. REFERENCES

1. NTU STARS Online Registration System Manual, 1995
2. Rational Rose, http://www.rational.com
3. Martin Fowler and Kendall Scott, UML Distilled: Applying the standard object modeling language, Addison Wesley, 1999.
4. Terry Quatrani, Visual Modeling with Rational Rose 2000 and UML, Addison Wesley, 2000