
DIAMANT

RUBEN GONZALEZ-RUBIO
DÉPARTEMENT DE GÉNIE ÉLECTRIQUE ET DE GÉNIE INFORMATIQUE
UNIVERSITÉ DE SHERBROOKE

1. INTRODUCTION

Diamant is a software system used to produce timetables at the Université de Sherbrooke, where it has been in use since 2001. Diamant allows the user to produce course and exams timetables. The timetable production can be done manually or automatically. This paper will detail the evolution of the system and its main features.

2. MOTIVATION

Producing timetables at the Université de Sherbrooke is done at the Faculty level. Each Faculty has its courses, instructors, students and rooms. The courses are offered for the whole year, there are three terms within the year: Winter (January to April), Summer (May to August) and Fall (September to December). A course timetable is prepared for each term in each Faculty. Each Faculty has a different way to build their timetables, some use curriculum-based and some use post enrolment based, another big difference is the slots in the timetable some are 3 hours some are 30 minutes. A exam timetable is prepared for the end of each term, in some cases a mid-term exam timetable is also prepared. The report [GR07] is a detailed account of how timetables are produced at Université de Sherbrooke.

It can be said that the main motivation to develop Diamant was to satisfy the needs of all Faculties.

Initially, Diamant was developed to replace another system called Saphir, because this system was too old to be updated. Saphir works in MS-DOS, and was programmed in a variant of Pascal. We chose a very conservative approach to creating this new program. It was first required to replace the existing system and then add new features. The program was set up so that Diamant would work like Saphir; using the same files and producing the same files. This process took about 18 months. Diamant has since been through several stages of evolution and is still in continuous progress towards further and better stages of development. Currently, the work is being done in a version adapted to produce timetables for the full term.

Taking in account all differences mentioned above, we believed that to build a system for each Faculty would be far too complex, especially for the maintenance of the program. Therefore, it was decided that only one system would be built, one that can be customized by changing the parameters for each Faculty. The development of the system was done using object oriented programming with Java and using the design patterns [GHJV95] along with

various practices coming from eXtreme Programming [Bec00] or [Mar03] to facilitate the evolution of Diamant.

3. EVOLUTION

When Saphir was operational, there was a central system where data concerning courses, instructors and students were stored. The information for room data was in a separate file. All data was transmitted in files. Therefore, Diamant was a stand alone program which read files coming from the central system, and then produced the timetables when conflicts within the scheduling were eliminated. The timetable was then transmitted to the central system in order to display the produced timetable information in a Web site.

There was no verification of the data when it is entered within the system, this means that the produced files can contain errors that could drive Saphir to crash. In Diamant it has been ensured that all of the data is verified and valid. Once this process has been completed, a Web system called DiamantWeb, that was created for the process of validating data, is used as a new interface where the data entry can be done. A timetable structure can be defined in DiamantWeb, and each Faculty can have a customized timetable, which, could include exams on Saturdays and Sundays. For example: A Faculty can decide that the exams must be scheduled in 5 or 6 day according to their needs.

We are currently working to produce timetables for a whole term. The process begins with a timetable structure where holidays are indicated in such a way so that no courses can be placed on those days. Some courses are not given the same day each week, or the same amount of time. DiamantWeb is easily accessible for many users, this is especially important for instructors so that they can indicate their availability and preferences concerning each course. In the past, this operation was completed using paper, whereby the instructor and the data were entered manually into the system by another person in charge of that particular operation.

The main aspect concerning this program is that DiamantWeb serves as a data entry point where data validation and preparation take place easily. Diamant is then used to produce timetables, and the user can eliminate conflicts manually or through an automatic build, as well as a combination of both methods.

4. DIAMANT MAIN FEATURES

Diamant can produce timetables manually or automatically for the three timetables types:

- exams [MMK⁺07].
- post enrolment [LPM07].
- curriculum based [DMS07].

This report was written to address the problems determined by the International Timetable Competition ITC 2007. The International Timetable Competition ITC 2007 reported that the current method of creating timetables for the various types were a problem and enumerated the ways that their solutions were evaluated, giving points or penalties in order to compare different solutions. The tracks were prepared to offer problems close to the real ones.

In this report the problems that are characterized by the ITC 2007 are used in order to indicate the possibilities of Diamant. Let us present how courses are organized at Université de Sherbrooke in order to justify why some details are different. In order to get a diploma, for example a bachelor in computer engineering a student needs to accumulate 120 credits of a well defined set of courses. Each course represent 3 credits. Normally the 3 credits are given to the student if he or she succeeds in the exam, and reaches the other requirements. Therefore, in order to succeed all exams and requirements, a student must perform different required activities during a term. A course is decomposed in differing events and these events must be scheduled in the weekly timetable. For example: The course Programming101 has 200 enrolled students and the classrooms that are available have room for 100 students, 50 students, and the computer laboratories have space for 25 respectively. This means that the events scheduled could be 2 lectures for groups of a 100 students, 4 groups of 50 students to make problem analysis and finally 8 groups in the computer laboratories to write and validate programs. In some cases the lectures are 3 hours but in others there are 2 hours + 1 hour lectures that are not on the same day. For the course Programming 101 there are 16 events to place in the timetable¹. Enrolled students in Programming 101 must take one final examination. Only one exam is scheduled for all Programming 101 students. The exam can take place in different rooms but all exams start at the same hour. Exam rooms can be shared by student sets taking two different exams.

The instructors are assigned during timetable process build. No instructor is able to be in two different places at the same time. Normally there is one instructor who is responsible for the grading of the exams, but he or she can be associated only to certain events, other instructors or assistants are assigned to the remaining events.

In the case of exams, the presence of the instructor responsible for the course is not required. Survey exams are done by assistants.

4.1. Exam timetable. In Diamant we can define a timetable composed by n days, each day can have a number of periods with a defined length. The length of the period could be between 5 minutes and 12 hours. University exams are all three hours in length. For this problem the user needs a set of exams, a set of students and a set of rooms. The set of students contains for each student all exams that he or she will take².

Hard Constraints:

- No student can sit more than one examination at the same time. This is taken into account.
- The capacity of individual rooms cannot be exceeded. The rooms will be filled as much as possible.
- Period lengths are not violated. In our example the exams and periods have the same length. Changing this can be easy because the only thing to add is the length of each exam.

¹There are 2 lectures groups, each has 2 lectures, 4 analysis groups and 8 laboratory groups.

²All courses where the student is enrolled

- Satisfaction of period related constraints e.g. ExamA after ExamB. This is not taken in account this kind of constraint but users can do so manually.
- Satisfaction of room related constraints e.g. ExamA must be in Room1. This could be done at any time. If a user sets two exams in the same room a conflict will be indicated.

The constraints that are taken into account are exams that are in a row and exams set in a day.

4.2. Post enrolment timetable. As in exam timetable, in Diamant we can define a timetable where events can be placed. The rooms are classified in categories, this means that the user can specify a room as "classroom" or "laboratory" then the system can take care of the capacity and features of the room. Normally, all slots are equivalent, but there some priorities to need to be decided upon if events should be placed in one particular slot. There are no precedence requirements.

Hard Constraints:

- No student should be required to attend more than one event at the same time. This is respected.
- In each case the room should be big enough for all the attending students and should satisfy all the features required by the event. The first part is respected and partially the second.
- Only one room is put into each room in a time slot. This is taken into account.
- Events should only be assigned to time slots that are pre-defined available for those events. There is no list like that.
- Where specified, events should be scheduled to occur in correct order during the week. There is no list like that.
- An extra requirement that we implement is that no instructor should be required to attend more than one event at the same time. The instructors have an availability that is respected.

We do not take care of specified soft constraints.

4.3. Curriculum based timetable. In Diamant we have an auxiliary program to take care of Curriculum based timetables. The auxiliary program generates virtual student sets, respecting the predictions. Furthermore, with these virtual student sets it is possible to take in account sets of events that must be programmed in different slots, because some virtual students follow the same set of courses.

Diamant takes care of all hard constraints for curriculum based timetables. We do not take care of specified soft constraints

5. CONCLUSION

The system Diamant has been presented and as detailed the system can take care of different types of timetable production. When the user is preparing to build a timetable he or she prepares the data for a specific type of timetable as outlined above, with the associated parameter the system works for this specific type of timetable. Users may enjoy

the facility of the system because they have already learned only one type of system with few commands. We do not fulfil all hard and soft constraints for all types of timetables because they are not requested by Faculties at the Université de Sherbrooke

REFERENCES

- [Bec00] K. Beck. *Extreme Programming Explained*. Addison-Wesley, 2000.
- [DMS07] L. Di Gaspero, B. McCollum, and A. Schaefer. The second international timetabling competition (itc-2007): Curriculum-based course timetabling(track 3). Technical Report QUB/IEEE/Tech/ITC2007/CurriculumCTT, Queen's University, Belfast, 2007.
- [GHJV95] E. Gamma, R. Helm, R. Johnson, and J. M. Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, 1995.
- [GR07] R. Gonzalez-Rubio. La production et la consultation d'horaires dans une université. Technical report, Université de Sherbrooke, 2007.
- [LPM07] R. Lewis, B. Paechter, and B. McCollum. Post enrolment based course timetabling: A description of the problem model used for track two of the second international timetabling competition. Technical Report ISSN: 1750-6658, Cardiff University, Wales, 2007.
- [Mar03] R. C. Martin. *Agile Software Development*. Prentice-Hall, 2003.
- [MMK⁺07] B. McCollum, P. McMullan, E. K. Burke, A. J. Parkes, and R. Qu. The second international timetabling competition: Examination timetable track. Technical Report QUB/IEEE/Tech/ITC2007/Exam, Queen's University, Belfast, 2007.