Idealist Timetabling

1157177

Abstract

The traditional way of timetabling regards user as a realist, who has fix future plan and tolerates any imperfection of the computed timetable which generates with certain algorithm. However, this timetabling style ignores the changing environment, which affects the plan of user at anytime anyway. Therefore, idealist timetabling approach which adopted a user-computer cooperation mode is proposed to better consider this factor. Idealist timetabling allows user to alter their plan anytime, even after the result has been generated. There is no final outcome under this style of timetable since the timetable can always be improved. This paper realize idealist timetabling by constructing a model using Empirical Modelling tool. The model situates time slot registration for third year student oral presentation. It allows teacher to update their schedule and student always gets an up-to-date schedule combining the timetable of their supervisor, assessor and moderator, plus the current selected room. This model illustrates the characteristic of idealist timetabling, and also explores the possibility of human computing.

1 Introduction

Timetabling is a rather mature technology in the field of computing. There exist various applications in the commercial market to deal with different kinds of timetabling problem. However, the approach of these applications is surprisingly unified, that is to find an optimal solution using certain algorithm, or in other words, automatic timetabling. This approach ensures to give a solution, provided that user set all the constraints preliminary, and the solution generated has little or even zero flexibility. These applications ignore the fact that the environment is always changing and affect the decision of user at the same time, which leads to amendment in generated solution.

Instead of taking the realist perspective, timetabling can be more reality-reflective by adopting idealist approach. The idea is to involve user throughout the whole process. The advantage of user-computer cooperation mode is user can always change the constraints according to their own situation. There is no final solution, but the result always reflects the reality and capable to improve.

This paper together with an idealist timetabling model developed using Empirical Modelling tool illustrate how such timetabling approach can be achieved.

2 Realist Timetabling and Idealist Timetabling

2.1 Realist Timetabling

Most of the currently available timetabling application regard user as a realist. A realist knows exactly the future plan and will not make any further changes. In this case, timetabling application serves as an automatic machine. It requires user to input every constraint in detail, and generate a timetable with certain underlying algorithm. If the output timetable is not perfect, a realist simply accepts it and has no intention to alter the timetable [1]. Schedule Expert® is a very good example of realist timetabling application. It is used to schedule final exams of students. In order to generate the timetable, it takes nine steps beforehand, which most of them are to set up constraints [2].

2.2 Idealist Timetabling

On the contrary, idealist timetabling treat every user as an idealist. An idealist always desires a perfect timetable. They need the flexibility of changing the timetable anyway anytime according to the ongoing situation, even after the timetable has been generated. Therefore, there is no final timetable in the mind of idealist. Instead of fully automatic machine, idealist timetabling application works in user-computer cooperation mode, which the user take charge of most details in timetabling. The next section talks about the detail of idealist timetabling model.

3 Idealist Timetabling Model

In order to illustrate idealist timetabling approach, an idealist timetabling model is developed using Empirical Modelling tool, EDEN interpreter.

3.1 Background

The situation of the model is set to be oral presentation time slot allocation for third year students. There are a list of teacher and a list of students, plus two rooms for the oral presentation. Each student has one supervisor, one assessor and one moderator, which mean each student has to present for three separate teachers at the same time. Each teacher has their own timetable to indicate if they are available, not available or potentially not available, and this timetable is always subject to change. Similarly, each classroom has its own available time. Each student has to register a time slot in one of the classroom that their supervisor, assessor and moderator are not unavailable. Since everything is allowed to change, the registered time slot of a particular student may become unavailable, and they have to register another timeslot.

Noted that the situation is not necessary to be oral presentation time slot allocation but anything else, this is one of the characteristic of idealist timetabling.

3.2 Structure

This model implements timetable in a colourful way. Figure 1 illustrates the interface of the idealist timetabling model. It is divided into two main components, namely list region on the left and schedule region on the right. The list region contains the list of teacher, the list of classroom and the list of student. The schedule region contains the schedule of a particular teacher, classroom or student. User can click on a member of teacher and classroom list to view their own schedule, and the selected member slot becomes yellow. The user can then click on the schedule to mark unavailable and potentially unavailable time slot, which is represented by red and orange colour respectively. User can also click on one of the student to view the schedule which combined the schedule of the selected classroom and the schedules of the three teachers of that student, the schedule also indicate the occupied time slot for the selected classroom in brown colour. If the time slot is not unavailable, the user can click on the slot to register for oral presentation at that time, the registered slot is indicated in blue colour with the selected classroom print on it. Figure 2 illustrates the ongoing timetabling process.

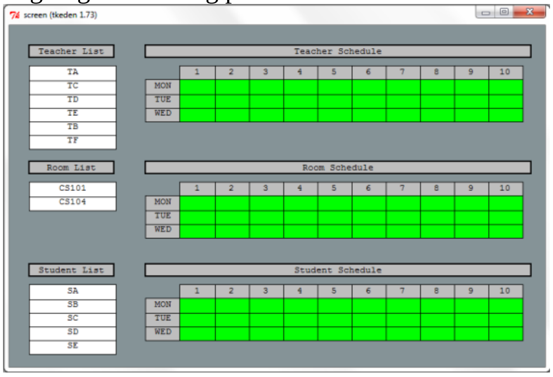

Fig 1: Interface of Idealist Timetabling Model

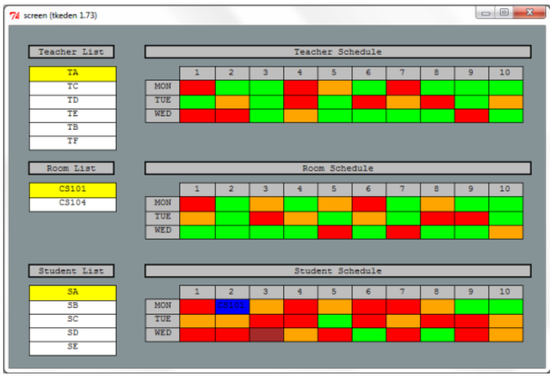

Fig 2: Snap of Ongoing Timetabling Process

3.3 Dependency

In this model, the observables are the member slot in list region and the time slot in schedule region. These observables change depends on the action of user.

The first dependency is student-teacher relationship. EDDI notation is used in the model, each entry in data file record the supervisor, assessor and moderator of a student. The model is implemented to generate list of teachers, list of classrooms and list of students automatically from all entries of the data file.

The size of schedule depends on the number of weekdays and number of time session each day. If user changes these two variables, the model generates schedule with correct number of time slot automatically.

In each time slot scout window, a variable *colour_o_r_c* is used, where *o* is the offset of this timeslot, 1 represents teacher schedule, 2 represents room schedule and 3 represents student schedule, *r* is the row number and *c* is the column number. Each time user click on a time slot, the *colour* variable changes, and the scout window refresh to show the current colour of that time slot.

The timetable is shared by each of the list member, which means *colour_o_r_c* only represents current value and will be vanished if user select another list member. Under this circumstance, timetable storage of each list member is simply a job of recording colour. A list *clist_o_k* is used to store the current colour of each time slot, where *o* is the offset and *k* represents a member of the list. The value of *colour_o_r_c* is store in nth position of *clist_o_k*, where n is equal to the number of column times r minus 1 plus c. With this implementation, when user click on a member of teachers or rooms, every value of *col-*

our are rewritten according to the value of *clist* of that particular member, and the timetable of that member is therefore restored.

The timetable of students is however different. It needs to reflect the schedule of three teachers and the selected room. Therefore, the values of *colour* are also stored in a list *tempclist_o_k*. For a particular student, the *tempclist* of this student's supervisor, assessor and moderator are added up and store in *tempclist* of this student. The *tempclist* of currently selected classroom is also added to the *tempclist* of this student. The rule of colour adding is red dominates all colour, and orange dominates green. Therefore, *tempclist* of this student becomes *clist* if registration is not yet happened.

If user click on an available time slot on the timetable of a particular student, that slot turns to blue colour, and the value of string *selectroom_0[n]* of that time slot scout window updates to the name of selected classroom, where n is the clicked position of the timetable. The selected classroom name is also stored in the list *selectroom_k*, where *k* represents a member of students. The purpose of this list is to store the registered time slot of a student. The list is then cross check with *tempclist* of the same student, if the colour of the registered time slot is red according to *templist*, the registration will be cancelled, else the *clist* colour of this time slot will be rewritten to blue.

All registered time slots of the selected room are shown in brown colour on the timetable of each student, except the one who register it. This is done by checking all *selectroom* lists except the one of current selected student, if the time slot is registered, the *clist* colour of this time slot will be turned into brown. ing.

4 Conclusion

Idealist timetabling is a different way to schedule a timetable. It provides the flexibility that realist timetabling lacks of. This paper demonstrates the possibility of idealist timetabling with help of a model developed with Empirical Modelling tool, which implemented with some basic idealist timetabling features, like allowing user to update their timetable anytime according to the changing environment and to reflect these changes with an updated timetable. The model not only illustrates the functionality of idealist timetabling, but also shows the possibility of human computing, which is likely to be the future style of computing.

References

[1] W.M.Beynon, A. Ward, S. Maad, A. Wong, S. Rasmequan, and S. Russ. The Temposcope: a Computer Instrument for the Idealist Timetabler. Proceedings of the 3rd international conference on the practice and Theory of Automated Timetabling, Konstanz, Germany, August 16-18, 2000

[2] Schedule Expert® Website: http://www.scheduleexpert.com/index.htm