A Decision Support System Improves Course Timetabling at the University of Waikato

L.R. Foulds
Department of Management Systems,
University of Waikato
New Zealand
lfoulds@waikato.ac.nz

D.G. Johnson
The Business School
Loughborough University of Technology
Loughborough
Great Britain
d.g.johnson@lboro.ac.uk

Abstract

In this paper we outline recent research into the complex and difficult task of university course timetabling, which involves both subjective and objective information and expert judgement. We go on describe the development and use of a decision support system, called SlotManager, for assigning courses to timeslots and rooms within a New Zealand university. SlotManager is designed to augment the expertise of an experienced timetabler, usually in a situation where timetables evolve from one year to the next, rather than being created, *ab initio*, on each occasion. We report on the successful implementation of the system, as operated by the Waikato Management School. The system is written in Microsoft Access, to run on a personal computer, and provides an effective and user-friendly addition to the university timetabler's toolkit.

1. Introduction

All educational institutions need to timetable the various teaching and related activities that they undertake, and universities are no exception to this. Over the last 25 years, many approaches and models have been proposed for dealing with a variety of problems related to timetabling, which have been largely specific to particular problem situations. A number of review articles and annotated bibliographies have surveyed course timetabling generally; for example Schmidt and Strohlein [10], de Werra [14], and White [15]. More recently, Kingston [6] lists 1056 references. The main problems dealt with are: allocating teachers to courses (Hilton [3] and Lawrie [7]), assigning courses to timeslots and rooms (Sampson, Freland, and Weiss [9], and Stallaert [12]), allocating students to different sections of a course (Wood and Whitaker [16]), and scheduling end-of-course examinations (Carter, Laporte, and Chinneck [2] and Johnson [4]). In the general area of course (rather than examination) timetabling, most approaches attempt to create timetables or assignments semi-automatically using particular models or heuristics (Badri, Davis, Davis and Hollingsworth [1]). However some authors seek to provide relevant decision support for an experienced timetabler (Mathaisal and Comm [8] and Johnson [5]).

2. Course Timetabling at the University of Waikato

The University of Waikato (UW) is a central government-funded university of approximately12,000 students, located in Hamilton, New Zealand. It comprises seven schools of study: management, law, arts and social sciences, computing and mathematical sciences, education, Maori and Pacific development, and science and technology. The timetabling process at UW used to begin in May of each year with a coordinated request from each school for its timetabling needs for the following teaching year. These requests were sent to a central university timetabler who finalized them by checking for conflicts between schools, and resolving such conflicts by negotiation with school or departmental timetabling representatives, who were usually teaching staff.

However, the situation at UW changed rather abruptly when the central administrators responsible for timetabling announced that they had purchased a computer-based timetabling package to replace the system described above. Henceforth, departmental representatives would have only to advise the central timetabling office of basic course requirements, from which the new system would automatically construct a timetable and allocate rooms for the coming year.

While, in principle, the teaching staff involved in timetabling welcomed the removal of a burdensome task, they had doubts about the ability of the package to provide a workable timetable. This was underlined by the insistence of the central timetabling staff that the system would produce a satisfactory timetable even though they appeared to have little idea of its capabilities, or how effectively it would cope with the complex requirements of UW. In particular, it was not clear how the system would resolve conflicts, as there was no provision in the proposed new procedure for any prior coordination between departments to reduce such conflicts, or for their fair and effective resolution if they occurred.

At this point, a number of senior members of the Waikato Management School (WMS) met with their Dean to discuss how best to respond to this new situation. It was decided that the WMS departmental timetable representatives would be invited to form a WMS timetable committee to coordinate the development of a WMS timetable for the coming year, attempting to: avoid clashes for teachers, be reasonably equitable for both WMS students and teaching staff, and to make effective and efficient use of the WMS teaching facilities These facilities comprise 18 rooms, ranging in size from a 394-seat lecture theatre, to 50-seat case study rooms, to 12-seat tutorial rooms. Most WMS teaching takes place in these rooms, which are seldom used by the rest of UW, and WMS has priority over the other schools in the timetabling of them.

2.1 Course Timetabling within the Waikato Management School

WMS comprises seven departments: accounting, economics, finance, management communication, marketing and international management, management systems, and strategic management and leadership. A complicating factor in the timetabling of WMS courses is the existence of a number of undergraduate degrees that involve, in addition to management courses, a wide range of courses taught by other schools within the university.

From the seven departments in WMS, a total of 166 courses a currently scheduled across two semesters and a summer school, of which 116 are undergraduate courses. Some of these courses are compulsory for the various undergraduate degrees. The courses, many with specialized facility needs, are taught in 50-minute periods between 8.00 and 19.00, Monday to Friday. However, in specifying the sessions required for their courses, it is common for teachers to request double (110-minute) or triple (170-minute) periods. The various course structures used are: (1,1), (1,1,1), (1,1,2), (1,2,1), (2,1,1), (2,2) and (3), where a ‘1’ indicates a 50-minute session, a ‘2’ indicates a 110-minute session, and a ‘3’ indicates a 170-minute session.

A further complicating factor is the existence of ‘piggybacking’. This is a system in which a course at one level and a similar course at one level higher are taught together in one room. Students enrolled in the higher level course are required to attend additional class sessions, held at other times, and to do

additional work, in order to be given credit for the higher-level course. Historically, the WMS timetable was based on a simple slot system comprising 14 slots of two, three or four hours scattered through the week. The timetabling process was basically a manual one, using a simple PC database and record-keeping system, with conflicting requests for rooms being resolved by negotiation, facilitated by the one central timetabler. Initially, when WMS comprised only two departments and had less than 1000 students, the system worked reasonably well. But recently, as the number of departments, courses, and students increased, the system became more and more difficult to operate. The process took longer and longer, negotiations became strained, there were increasing complaints from students over course clashes, and from staff over inappropriate room allocations and awkward times. Overall, there was a feeling within WMS that the timetables were more inconvenient than they needed to be and, in general, poor use was being made of the WMS teaching facilities.

The formation of the WMS timetable committee triggered some significant changes in the way that the WMS timetable was produced. At the outset, there was an agreement between WMS and the central timetablers that the first draft of the WMS timetable would continue to be produced by WMS, and that WMS would continue to have first priority over the use of its teaching rooms. The WMS timetable committee would devise a complete WMS timetable for submission to the central timetablers as a coordinated request, with the expectation that it would be incorporated into the university timetable with little alteration. Any alterations that were necessary would first be negotiated with WMS.

The committee felt that the procedure used in the past should be continued, but based on a more comprehensive slot system. In addition, a Decision Support System (DSS) should be created to keep records, check for inconsistencies, and provide suggestions and alternatives when required. The DSS would be designed to facilitate both the modification of a previous timetable, and also the creation of a completely new timetable. To minimize the committee members' time, it was agreed that meetings would be held only as needed, and that the committee chairperson, aided by the DSS, would devise the timetable by consulting with each departmental representative individually. The representatives would provide the chairperson with their departments' timetable needs at each level, along with preferred slots and rooms.

It was decided that, for timetabling purposes, the priority of courses should be: the compulsory first year courses, all other compulsory courses, remaining second year courses, remaining third year courses, remaining fourth year courses, and all graduate courses. At each level, clashes would be resolved by negotiation, facilitated by the chairperson, before considering the next level of courses. As the timetable was built up, a read-only version of it should be included on the WMS website to enable teaching staff to view the tentative assignments made at each level, and to make observations to their departmental representative. The DSS would also provide displays of timetables for individual teachers, groups, departments, and hours of the day, as well as various lists and statistics, showing slot details for undergraduate and graduate courses, and the utilization of individual rooms. We shall now discuss the individual components of the DSS that was developed, and its implementation.

3. The Decision Support System: SlotManager

Although the DSS, which is called SlotManager, has been developed specifically for use by WMS timetablers in the construction of the School's teaching timetable, it is potentially much more generally applicable. The only constraint on its use is that the timetable operates on a slot basis, but there is no, *a priori*, fixed definition of the slot structure, which can be freely and easily changed to suit different requirements. During the two main semesters, different slot systems are used by WMS for undergraduate courses and for graduate courses, and the DSS must deal with them simultaneously to devise a timetable for each semester, and to check for conflicts. It is possible to define each period within the week as a unique slot, so that classes can be timetabled on a period-by-period basis.

SlotManager is made up of three components: the database, the user-interface, and the model base, as illustrated in Figure 1. The system is written in Microsoft Access, within the Microsoft Windows

environment, to operate on a PC. The objective is to accept necessary information from a variety of sources, to organize this into relevant database files and tables, to construct progressively a feasible (i.e. clash-free) timetable for the semester concerned, and to allocate all courses to suitably-sized rooms. At any stage, the information processed to date can be displayed and printed in a variety of formats to provide relevant user information, relating to staff and student timetables, room allocations, slot structures, and classes on a period-by-period basis.

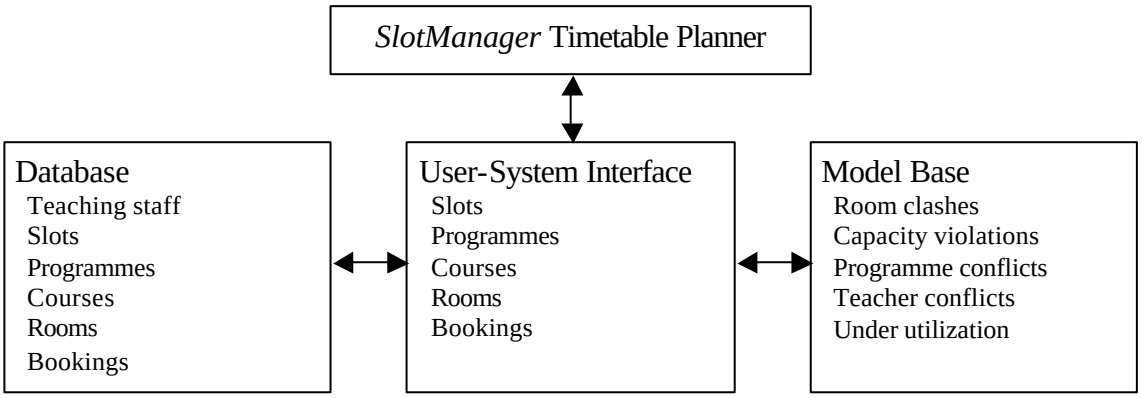

Figure 1: The structure of the four main components within SlotManager

The system comprises a series of modules, which are described in detail later in this paper. Each module is accessed from the master screen and provides access to an appropriate set of procedures, which are entirely mouse-driven, for inputting relevant information, allocating classes to periods and rooms, or outputting the results in a variety of formats. The database component, which holds all this information in a set of inter-related tables, is described in detail in the following section.

If an infeasible allocation is proposed, this will be because either: another course involving the same teacher or another course with significant student overlap has already been assigned to that slot, the room chosen has already been assigned to another course, or the room is not big enough. The system indicates the nature of this conflict and, using the model base, suggests a range of alternative actions. These are the various alternative feasible slots and rooms to which the current course could be assigned, or possible re-assignments of a previously assigned course with which the current course is in conflict. This is discussed in more detail in a later section. It should be emphasized, however, that the system never makes any allocations automatically, but merely suggests a range of feasible actions based on appropriate rules. The decision to make or accept a suggested allocation is always left with the timetabler.

3.1 The database component

At the heart of the database is a *Bookings* table within which the timetable details are compiled and stored. This table contains the basic information that defines the timetable for a specified year and semester, namely the day and time that a particular course meets, and in what room. In addition, this table must also capture any piggybacking arrangements that relate to this course. Apart from the year and semester concerned, all the information in this table relates to other tables in the database. For example, the room information is drawn from a *Rooms* table containing the room number and capacity of all UW teaching rooms, and the course concerned is specified in a *CourseDetail* table, which draws some of its information from the WMS main database.

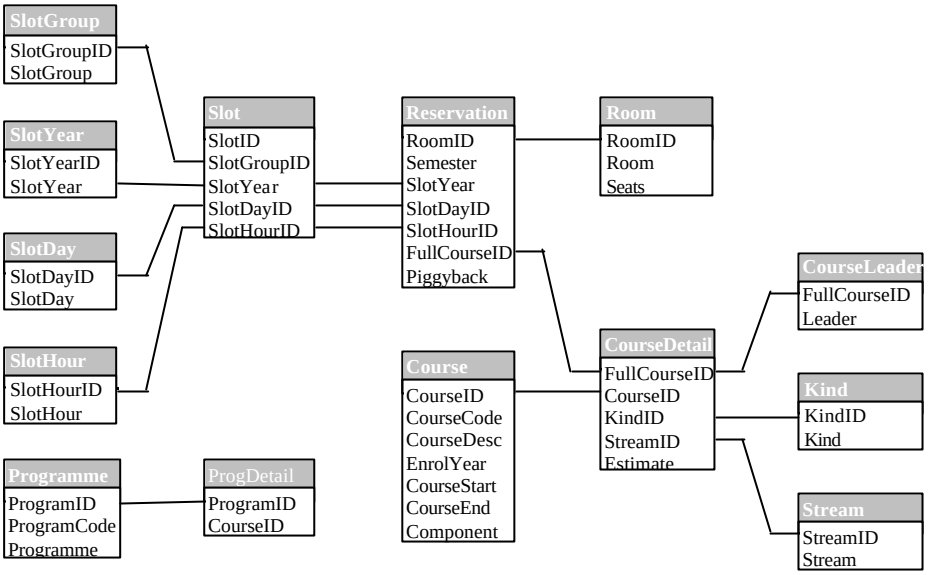

Figure 2: Database Relationships

The pre-existing course information that is held in the WMS main database is shown in Figure 2 as the *Course* table. Some of this information, such as the course start and end dates and description, is not needed by SlotManager, although it could be accessed if required, using the unique course identifier *CourseID*. The other important course information that is held in the *CourseDetail* table is an *Estimate* of the course enrollment, the *Kind* of class in question (lecture, tutorial or computer laboratory), and any *Streams* that exist for this course. It is also possible to identify the teacher responsible for each course (*Leader)* by reference to the *CourseLeader* table which links to another file in the WMS main database that contains the details of all WMS teaching staff.

The information about the timing of any class is drawn from a *Slot* table, which in turn is compiled from a series of input tables that define the *SlotGroup*, *SlotYear*, *SlotDay* and *SlotHour*. *SlotGroup* allows any number of sets of slots to be specified corresponding to different groups of students such as undergraduates, graduate students, and summer school, and *SlotYear* permits these sets to vary from year to year. An individual record in the *Slots* table will therefore define a particular period for some group of students in a particular year.

3.2 The model base

SlotManager was originally conceived as a computer based system to aid the process of allocating courses to slots and rooms by keeping records of various aspects of the allocation and facilitating the printing of a variety of reports. The only 'intelligence' initially built into the system was the ability to detect conflicting allocations where:

- A room was being allocated to more than one class at the same time, or
- A room was being allocated to a course whose enrollment exceeded the room capacity or
- A course leader was being timetabled to teach more than one class at the same time.

In particular, there were no suggestions as to how a conflict might be resolved, which was left to the experience and insight of the timetabler. However, during the evaluation of the initial prototype, a number of suggestions were made by the timetablers as to how the model base could be extended to examine additional features of a proposed allocation and to give a variety of suggestions on how conflicts might be resolved.

The incorporation of such suggestions is significantly more complex than simply detecting that a conflict or violation exists. For example, when a room clash is discovered, there are usually alternative rooms available at the relevant time, but these rooms may not be equally suitable. In particular, they will vary in their capacity, type, and location. The alternatives must therefore be filtered in some way to sift out the inappropriate ones, and those remaining should ideally be ranked in decreasing order of suitability. After detailed discussion with the timetablers, an extensive rule base was incorporated to replicate the factors that timetablers would consider when selecting an alternative room from those available. Thus SlotManager includes some features of an expert system. In those instances where the rule base would have filtered out all available rooms, the system preserves the best of these so that at least one alternative can be offered, but its unsuitability, for whatever reason, is flagged.

In a situation where the competition for teaching rooms is acute, particularly for large rooms in the more popular slots, it is important that such rooms be used efficiently. It was therefore felt to be appropriate not only to signal violations of room capacity, but also to indicate where a room is being allocated which is significantly larger than is required to accommodate the anticipated number of students. This is not a simple issue, in that under-utilization can be measured in various ways, and different measures are appropriate for different rooms. For very small rooms, there is generally no need to signal under-utilization at all, whereas for the larger rooms, it is usually a percentage utilization factor that is appropriate. For medium-sized rooms, the utilization measure should also take into account the actual number of seats unused. Again, a rule base was developed which signals when the allocated room might be too large, and proposes, where possible, a range of alternatives.

The course leader information described earlier is used to identify when more than one course, involving either the same teacher or the same group of students, has been allocated to the same slot. In either of these cases, one of the conflicting courses must be moved to an alternative slot. In passing, it is worth noting that the concept of a "group", comprising a number of courses being taken by the same group of students, can be introduced and used quite flexibly. It may comprise a group of courses within one year of an actual degree, or across a number of years when the courses may be taken at different stages. In some cases, however, a 'group' is simply an artificial construct to group together a set of courses that, for a variety of reasons, it is inconvenient to timetable simultaneously.

Finding an alternative slot for a conflicting course is, in a sense, similar to the task of finding an alternative room in the case of room clashes. The size and location of a room is analogous to the structure of a slot and its position within the week. Finding an alternative slot requires one that is 'similar' in structure and timing to the one originally allocated. The structure of any slot is defined in terms of its pattern of single, double and triple periods, as described earlier, and it is assumed that any alternative slot proposed should, if possible, have the same structure as that originally requested. If this is not possible, an alternative structure has to be used, but one that is close to the original. For example, if a (1,1,2) slot is required and none is available, then the closest structure is deemed to be (1,2,1) followed by (2,1,1) and finally (2,2). A distance matrix is used to measure the proximity within the week of all slots of a particular structure, and the possible alternatives having the same (or the nearest available) structure are listed in decreasing order of proximity.

In trying to resolve any of the conflicts, the search first considers alternatives for the course currently being timetabled. If none of these alternatives is attractive, the timetabler can use SlotManager to display alternative rooms and slots for the previously allocated course with which the current one is in conflict. This introduces a limited amount of backtracking into the model base, but this is invoked only if the current course cannot be satisfactorily accommodated. It would be possible, in principle, to build in more levels of backtracking, but experience in using the system indicates that it is almost always possible to resolve conflicts with just one level of backtracking, along with the timetabler's judgement.

3.3 The interface component

As previously stated, SlotManager comprises a number of modules that facilitate the acquisition and organization of information relating to: PROGRAMMES, SLOTS, COURSES, ROOMS and BOOKINGS in any year. The modules contain various procedures for inputting and outputting information, although the PROGRAMMES and ROOMS information tends to be reasonably static from year to year. The SLOTS and COURSES information usually requires a small amount of updating each year, whereas the BOOKINGS data can be expected to change substantially year by year. The first four modules are reasonably independent of each other, and so can be used in any order, but the BOOKINGS information for any year is usually inputted only after the other modules have been used to incorporate any changes that are necessary.

The BOOKINGS module is used to create the *Bookings* table described above. In particular, new bookings are added to the file by allocating a chosen course to a timeslot and room for a selected year and semester. The table may be displayed by room, course, time, or department, and reports can be printed by room or course, showing the allocations that have been made to date. An illustration of a typical screen in the BOOKINGS module is shown in Figure 3.

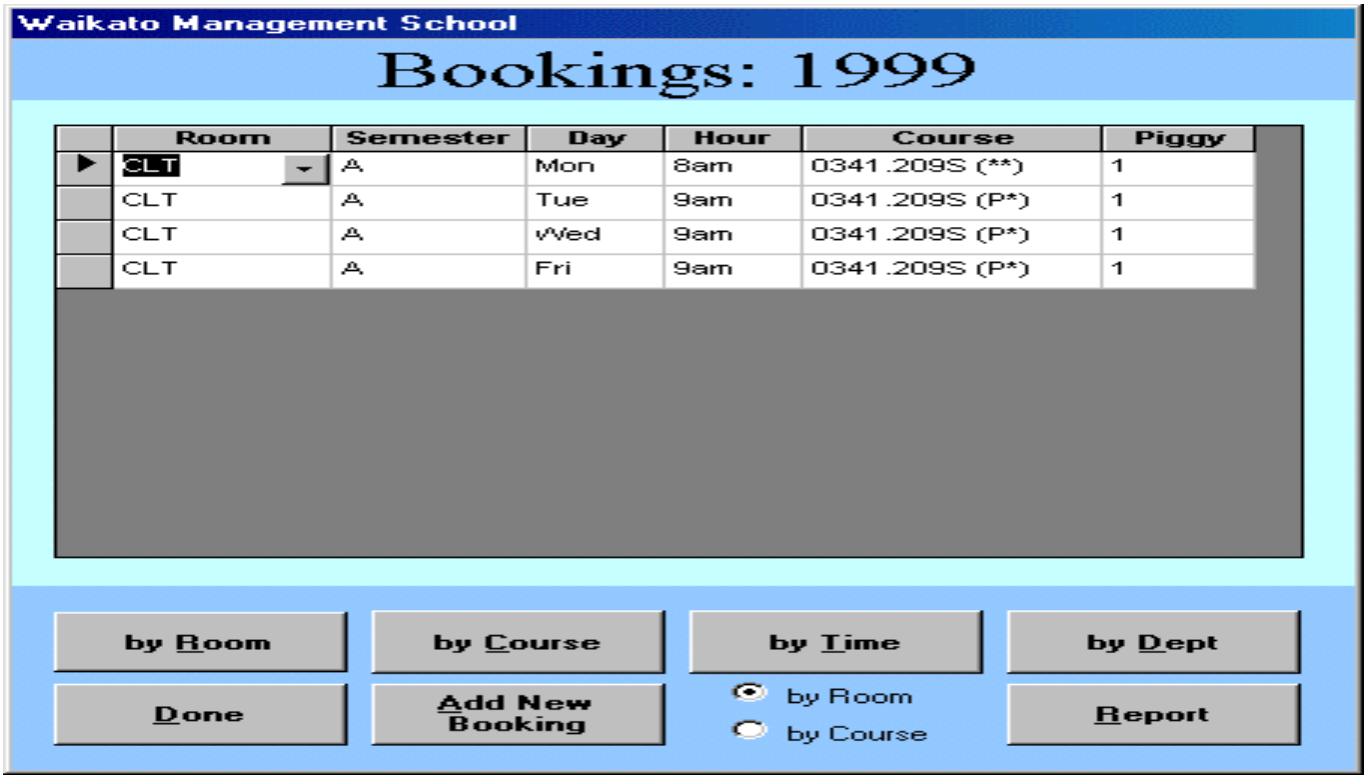

Figure 3: The BOOKINGS menu showing the various display alternatives available.

The PROGRAMMES module defines the various student groups (undergraduate, graduate, and summer school) being timetabled. The master screen allows any one of these groups to be selected, along with the year to be timetabled. The PROGRAMMES module displays lists of all necessary groups of courses that must be timetabled separately. This enables both new groups to be inputted and the course list for any group to be edited (i.e. courses added or deleted). The central dialogue box, through which a particular course can be added to the currently selected group, accesses the master course file in the WMS database so that all of the WMS courses are available for inclusion in any group. Both the list of groups and the course list for any group can be outputted in report form. An example of this display is shown in Figure.

4. As the groups use different slot patterns, the SLOTS module is then used to define the various periods (day and time) that make up each slot for the selected group. The only limit on the number of slots that may be specified is the number of periods in a week. The slot information can be displayed either chronologically in slot order or in the form of a two-way table, by day and hour. A report of this latter display can be printed if required.

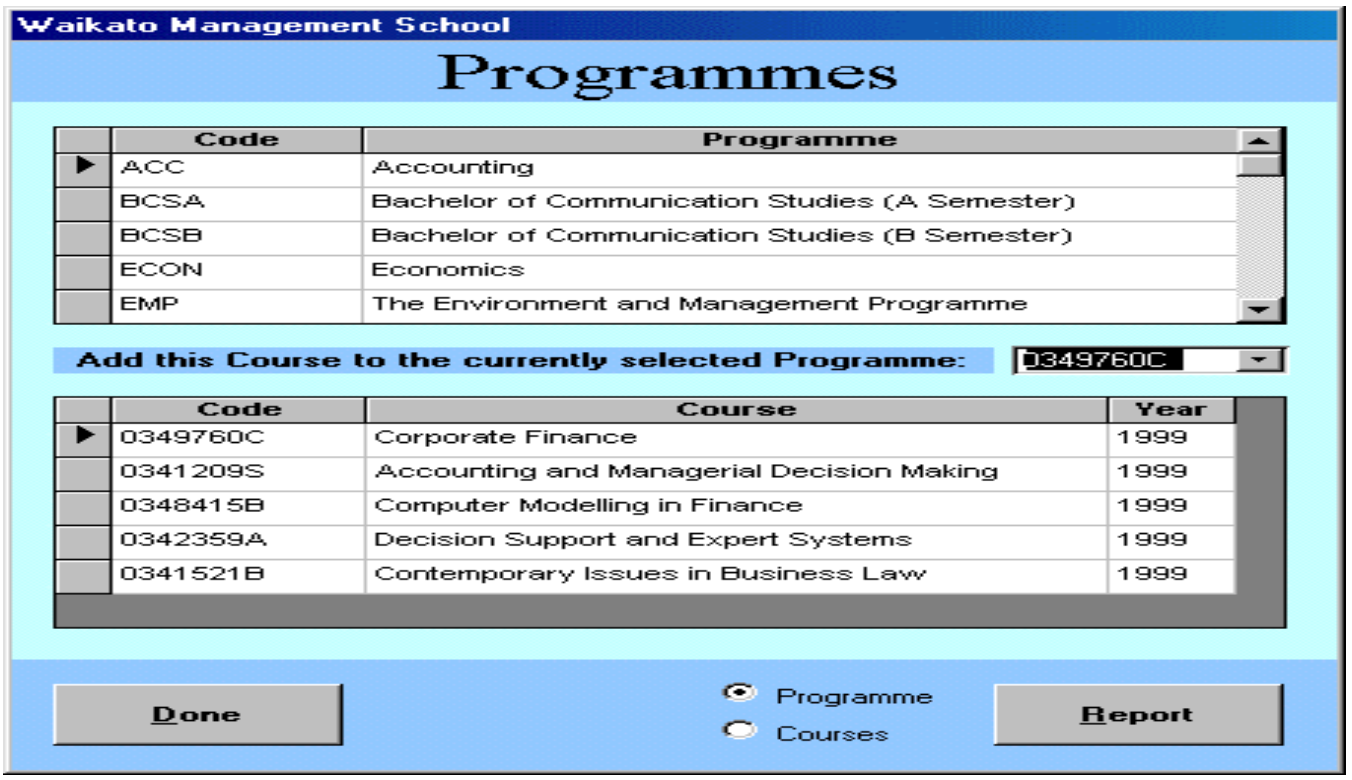

Figure. 4 The PROGRAMMES menu

The ROOMS module allows the list of teaching rooms and capacities to be created, amended, or printed as required. The COURSES module similarly accesses the course master file to permit the inclusion of information relating to class type, streams, course leaders and estimated enrollments. The COURSES module similarly accesses the course master file to permit the inclusion of information relating to course type, streams, course leaders, and estimated enrollments.

4. Implementation of the DSS

SlotManager is implemented within a local area network that provides access to certain WMS database files held on a central server. All data, other than that relating to WMS courses and staff, is held locally within the system. The DSS was developed using an iterative (prototyping) approach which, according to Turban [13], "is the most common in DSS development, since the information requirements are often not known precisely." Systematic testing ensured that each module was error-free before additional modules were incorporated, and that the modules currently in place interfaced correctly with each other.

The need for the involvement and participation of potential users was recognized from the outset. The WMS timetable representatives were involved not only in developing the initial specification for the system, but they also participated in exhaustive testing at each stage. Following the development of the original prototype, they gave valuable feedback on the layout of the user-interface screens, and also made suggestions for additional options and features that should be incorporated to support all aspects of the

timetabling process, as discussed earlier. This lead to minor changes in the structure of the files that make up the SlotManager system and, more significantly, to the facilities provided within the model base.

SlotManager does not automatically generate timetables that attempt (and most likely fail) to satisfy all the various constraints and complicating factors that will inevitably exist in any practical timetabling situation. Rather, the purpose of the DSS is to assist an experienced timetabler to allocate courses to both rooms and slots when producing a timetable, either from scratch or by modifying an existing timetable. As a consequence, SlotManager does not have a mathematically complex model base that uses sophisticated procedures to attempt to replicate the expertise of the timetabler. In practice this is likely to be difficult, if not impossible, to achieve, and even if it were achievable in certain situations, it may be counter-productive, because experienced timetablers are often reluctant to accept the solution of a model or process that they do not understand.

SlotManager includes all of the characteristics of an effective DSS from the generic DSS framework proposed by both Sprague & Carlson [11] and Turban [13], namely:

- It supports, but does not replace the decision-maker. It should therefore neither try to provide the 'solution' nor to impose a predefined sequence of analysis,
- It supports a semi-structured decision where parts of the analysis can be systematized for the computer, but where the decision-maker's insight and judgment are needed to control the process,
- It combines modelling techniques with database and presentation techniques,
- It emphasizes ease of use, user friendliness, user control, flexibility and adaptability,
- It supports all phases of decision-making, and
- It interacts with other computer-based systems to download and upload information.

SlotManager has been in use within WMS for many years and is now accepted as an integral part of the timetabling process. The time spent by the timetablers in developing the timetable has been reduced significantly, and the efficiency of the record-keeping and report preparation has been greatly improved. The creation of the timetable is still largely a judgmental process, but one which is now undertaken more professionally, and which, as a consequence, gives rise to far less dissatisfaction from both staff and students, and leads to more effective and efficient utilization of teaching rooms.

4 Summary and conclusions

We have described a DSS that has recently been developed and implemented at the Waikato Management School to improve the timetabling process. Although it has been developed in the context of a particular university and school, the only significant constraint on its general application is that the time periods being timetabled are divided into slots. SlotManager was designed to assist timetablers at every stage of the timetabling, but does not automate the process. Rather, it helps timetablers by providing powerful and wide-ranging tools to: designate the required slots, allocate courses to feasible slots and rooms, and create a wide variety of reports on the outcome of the timetabling process.

In general, DSS benefits are often uncertain and difficult to assess. With the prototyping approach, where development is evolutionary, this is especially so. The ongoing changes to the education system, and the continuing search for greater efficiencies and effectiveness in the use of both staff time and physical resources, make it even more so. The true value of a DSS is whether it improves both decision-makers' effectiveness and the quality of the decisions made, which are not easily measured. Therefore the traditional cost-benefit approach may not be able to capture all of the benefits of a particular DSS. In our case, however, some of the benefits can be measured, such as the time taken by the timetablers to produce an acceptable timetable. The system has clear benefits, even if they are not easily quantified. Usage of the increasingly scarce resource of teaching rooms has demonstrably improved, and there is clearly a greater level of acceptance, on the part of both students and teachers, of the perceived quality of the teaching timetable. Furthermore, conflicts over the slots or rooms allocated to courses are more efficiently and equitably handled by a system that shows the alternatives (or lack of them) that exist. The

'what-if' questions concerned with a perceived unsatisfactory allocation can be much more effectively explored and, if possible, resolved.

In summary, SlotManager is designed to augment the expertise of an experienced timetabler, usually in a situation where timetables evolve from one year to the next, rather than being created, *ab initio*, on each occasion. The reactions of experienced timetablers who have used SlotManager have been, on the whole, positive and it is accepted as an indispensable aid to timetabling at WMS.

Acknowledgement

The authors are indebted to Mr John Hunter, database programmer at the Waikato Management School for creating the SlotManager program.

References

[1] Badri, M.A., Davis, D.L., Davis D.F. and Hollingsworth, J. 1998, "A multi-objective course scheduling model: combining faculty preferences for courses and times," *Computers and Operations Research*, Vol. 25, No. 4, pp. 303-316.

[2] Carter, M.W., Laporte, G. and Chinneck, J.W. 1994, "A general examination scheduling system," *Interfaces*, Vol. 24, No. 3, pp. 109-120.

[3] Hilton, A.J.W. 1981, "School timetables," *Annals of Discrete Mathematics*, Vol. 11, pp. 177-188.

[4] Johnson, D.G. 1990, "Timetabling university examinations," *Journal of the Operational Research Society*, Vol. 41, pp. 39-47.

[5] Johnson, D.G. 1993, "A database approach to course timetabling," *Journal of the Operational Research Society*, Vol. 44, pp. 425-433.

[6] Kingston, J.H. 1995, "Bibliography on practice and theory of automated timetabling," http://liinwww.ira.uka.de/bibliography/Misc/timetabling.html

[7] Lawrie, N.L. 1969, "An integer linear programming model of a school timetabling problem," *Computer Journal*, Vol. 12, pp. 307-316.

[8] Mathaisel, D. and Comm, C. 1991, "Course and classroom scheduling: an interactive computer graphics approach," *Journal of Systems and Software*, Vol.15, pp. 149-157.

[9] Sampson,S.E., Freeland J.R. and Weiss, E.N. 1995 "Class scheduling to maximize participant satisfaction," *Interfaces*, Vol. 17, No. 2, pp. 30-41.

[10] Schmidt, G. and Ströhlein, T. 1979, "Timetable Construction – an annotated bibliography," *The Computer journal*, Vol. 23, pp. 307-316.

[11] Sprague, R.H. and Carlson, E.D. 1982, "Building Effective Decision Support Systems," *Prentice-Hall*, Englewood Cliffs, New Jersey.

[12] Stallaert, J. 1997, "Automated timetabling improves course scheduling at UCLA," *Interfaces*, Vol. 27, No. 4, pp. 67-81.

[13] Turban,E. 1993, "Decision Support and Expert Systems: Management Support Systems 3rd Edition," *Macmillan*, New York.

[14] de Werra, D. 1985, "An introduction to timetabling," *European Journal of Operational Research*, Vol. 19, pp. 151-1623.

[15] White, G.P. 1987, "A survey of recent management science applications in higher education," *Interfaces*, Vol. 17, pp. 97-108.

[16] Wood, J. and Whitaker D. 1998, "Student Centred School Timetabling," *Journal of the Operational Research Society*, Vol. 49, pp. 1146-1152.