Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Engineering

Elixir Comp. Engg. 80 (2015) 31128-31133

ISSN: 2229-712X

Academic Course Scheduling Decision Support System

Ajayi, Olusola Olajide[1], Ojeyinka, Taiwo O[1], Adefila, Olanrewaju Adebayo[1], Iyare, Rachel Nkem[2], Lawal, Muideen Adekunle[1] and Isheyemi, Olufemi Gabriel[1]

[1]Department of Computer Science, Faculty of Science, Adekunle Ajasin University, Akungba-Akoko, Ondo State, Nigeria.

[2]Department of Physics and Electronics, Faculty of Science, Adekunle Ajasin University, Akungba-Akoko, Ondo State, Nigeria.

ARTICLE INFO

Article history:

Received: 7 October 2014;
Received in revised form:
28 February 2015;
Accepted: 14 March 2015;

Keywords

Scheduling,
Decision,
Academic course.

ABSTRACT

Web-based decision support systems are used to model data and make adequate decisions based upon it. They address the needs of managers in solving structured and semi-structured problems. They have provided assistance to human discrimination to put right people in right places. This research paper focuses on Academic Course Scheduling Decision Support System, which provides a reliable tool that can be used to improve decision making process in academic course scheduling. Due to the dearth of adequate classrooms in most academic institutions, and pressure on the available ones, it becomes more and more difficult for school management to battle with the problem. This paper gives a model for appropriate scheduling of courses to classrooms, as well as determining courses priorities. To facilitate good implementation of this work, the researchers painstakingly took out time to collect enough data from the case study (Adekunle Ajasin University, Akungba-Akoko, Ondo State, Nigeria) as well as materials and journals from various authors/sites. The application was developed using a scripting language, PHP, and MySQL database as backend to effectively achieve the aims of this project.

Introduction

Time is the scarcest resource to humans. Scheduling is about making the most of a limited amount of time. Scheduling emerges in various domains, such as nurse scheduling, airplane landing scheduling, train scheduling, and production scheduling (Kassicieh*et al*, 1986).

The task of scheduling, especially when it comes to academic lecture, is a very complex endeavor. Satisfying a variety of needs and requirements while maintaining standards for efficiency and effectiveness is difficult due to political pressures exerted by those who are scheduled. The assignment of courses, time blocks, and classrooms impacts strategic planning issues such as the need for new buildings, expansion of course offerings and admission policies.

Decision Support Systems (DSS) comprise software systems that assist humans in making complex decisions in real-life problem domains (Flynn *et al*, 2002). With the advent of the Internet and powerful computing devices over the last decade, dynamic and intelligent decision support is rapidly emerging as the new research direction in the field of decision support systems.

Decision-making problems in real life are characterized by complex, unstructured nature of problem domains, unpredictable outcome of decisions due to the dynamic nature of problems and information, and the potential risks associated with making an incorrect/inaccurate decision. In such a scenario, a naive model that uses a stationary mapping from situation to decision is inadequate for making correct decisions.

Decision Support Systems is the area of Information Systems (IS) discipline that is focused on supporting and improving managerial decision-making. The study of Information Systems, originated as a sub-discipline of computer science, in attempt to understand and rationalize the management of technology within organization. Information Systems are generally used in organizations to help provide for information needs of the organization.

Academic course scheduling is a complex operation that requires the interaction between different users including instructors and course schedulers to satisfy conflicting constraints in an optimal manner. Academic lecture scheduling is a major administrative activity in any academic institution. A number of courses taught by the corresponding lecturers are allocated into a number of available classrooms and a number of timeslots, subject to constraints.

The objective is to maximize their preferences including the room(venue) and time of day; from the administration's point of view, the efficient utilization of the physical facilities is a concern as well. Academic lecture scheduling problem is said to be feasible if and only if it satisfies the following constraints : (a) every lecturer and every class must be present in the timetable in a predefined number of periods; (b) there cannot be more than one lecturer in the same class at the same period; (c) no lecturer can be assigned to more than one class at the same time; (d) there can be no "uncovered periods" (that is, periods when no lecturer has been assigned to a class). Violation of the constraints leads to an infeasible solution which usually happens in the manual preparation of the timetable. Therefore, these constraints have to be satisfied in order to get a feasible solution. The motivation behind this study arose due to pressures on academic lecture schedule. In order to subdue the pressures however, there is need to provide a better decision support system platform to handle academic lecture scheduling for academic institutions. The research will therefore aim at developing a comprehensive and responsive decision support system that will support academic planners in planning and scheduling academic resources to meet future instructional needs.

Tele: +234(0)8137044500 / +234(0)7056433798
E-mail addresses: ajayilastborn@yahoo.co.uk

In order to achieve the objectives of this research, the following generic stages involve in the development of any DSS, viz: data gathering and collation, data storage and access, data analysis and data reporting, will be applied.

Related Literature

In the early 1960s, organizations were beginning to computerize many of the operational aspects of their business. Information systems were developed to perform such applications as order processing, billing, inventory control, payroll, and accounts payable. The goal of the first management information systems (MIS) was to make information in transaction processing systems available to management for decision-making purposes. Unfortunately, few MIS were successful (Ackoff, 1967;Tolliver, 1971). Perhaps the major factor in their failure was that the IT professionals of the time misunderstood the nature of managerial work. The systems they developed tended to be large and inflexible and while the reports generated from managers' MIS were typically several dozen pages thick, unfortunately, they held little useful management information (Ackoff, 1967; Mintzberg, 1977).

The term "decision support systems" first appeared in a paper by Gorry and Scott Morton (1971), although Andrew McCosh attributes the birth date of the field to 1965, when Michael Scott Morton's PhD topic, "Using a computer to support the decision-making of a manager" was accepted by the Harvard Business School (McCosh, 2004). Gorry and Scott Morton (1971)constructed a framework for improving management information systems using Anthony's categories of managerial activity.

DSS are software applications that have been used over the last few decades to provide support for many structured and unstructured problems such as strategic planning, investment planning, stock portfolio management, enterprise planning, human resources management, supply chain planning, knowledge management, case-based reasoning and help desk automation (Clemen, 1996; Mallach, 2000; Marakas, 1998; Mora, Forgionne, & Gupta, 2002; Turban and Aronson, 1997).

Gorry and Scott Morton conceived DSS as systems that support any managerial activity in decisions that are semi-structured or unstructured. Keen and Scott Morton (1978) later narrowed the definition, or scope of practice, to semi-structured managerial decisions; a scope that survives to this day. The managerial nature of DSS was axiomatic in Gorry and Scott Morton (1971), and this was reinforced in the field's four seminal books: Scott Morton (1971), McCosh and Scott Morton (1978), Keen and Scott Morton (1978), and Sprague and Carlson (1982).

Decision support systems play an increasingly important role in higher education institutions. Doost (1997) described a potential academic workload database system. Deniz and Ersan (2001, 2002) proposed a DSS for student, course and programme assessment. Dasgupta and Khazanchi (2005) described intelligent agent enabled DSS for academic course scheduling. Vinnik and Scholl (2005) proposed an academic workload management DSS which focuses on academic capacity utilization using a process of balancing educational demand and supply in universities. Important part of academic DSS is academic Workload Planning System (WPS) which focuses on balancing load against capacity. Keys and Devine (2006) proposed future development of practical system for academic workload management using equity weighting for workload assignments.

Much of the early work on DSS was highly experimental, even radical (Alter, 1980; Keen, 1983). The aim of early DSS developers was to create an environment in which the human decision maker and the IT-based system worked together in an interactive fashion to solve problems; the human dealing with the complex unstructured parts of the problem, the information system providing assistance by automating the structured elements of the decision situation. The emphasis of this process was not to provide the user with a polished application program that efficiently solved the target problem. In fact, the problems addressed are by definition impossible, or inappropriate, for an IT-based system to solve completely. Rather, the purpose of the development of a decision support system is an attempt to improve the effectiveness of the decision maker. In a real sense, DSS is a philosophy of information systems development and use and not a technology.

Architecture of Decision Support Systems

Different authors identify different components in a DSS. For example, Sprague and Carlson identify three fundamental components of DSS: (a) the database management system (DBMS), (b) the model-base management system (MBMS), and (c) the dialogue generation and management system (DGMS).

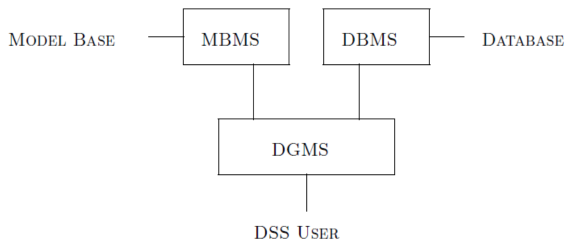


Figure 1. The architecture of a DSS

Source: DegifTeka 2008

Haag et al. describe these three components in more detail: The Data Management Component stores information (which can be further subdivided into that derived from an organization's traditional data repositories, from external sources such as the Internet, or from the personal insights and experiences of individual users); the Model Management Component handles representations of events, facts, or situations (using various kinds of models, two examples being optimization models and goal-seeking models); and the User Interface Management Component is, of course, the component that allows a user to interact with the system.

According to Power, academics and practitioners have discussed building DSS in terms of four major components: (a) the user interface, (b) the database, (c) the model and analytical tools, and (d) the DSS architecture and network.

Hättenschwiler identifies five components of DSS: (a) users with different roles or functions in the decision making process (decision maker, advisors, domain experts, system experts, data collectors), (b) a specific and definable decision context, (c) a target system describing the majority of the preferences, (d) a knowledge base made of external data sources, knowledge databases, working databases, data warehouses and meta-databases, mathematical models and methods, procedures, inference and search engines, administrative programs, and Reporting systems, and (e) a working environment for the preparation, analysis, and Documentation of decision alternatives.

Arakas proposes a generalized architecture made of five distinct parts: (a) the data management system, (b) the model management system, (c) the knowledge engine,(d) The user interface, and (e) the user(s).

Academic Course Scheduling

Due to the large number of dimensions and geometric scaling for even small problems, academic lecture scheduling attracts the interest of researchers from several disciplines. Empirical results from studies of scheduling solely within one College at a U.S. University are informative. Mooney, et. al. [1996] found that an objective of a small number of scheduling conflicts for a few courses is actually preferable to strenuously trying to keep the average number of conflicts small but allowing an arbitrary worst case schedule. Further, Mooney, et al found that even after many incremental improvements in the optimization model, "serious challenges… in the areas of… preferences, fairness, and robustness" Still exist.

Badri, et al. (1998) found that the "complex utility functions could limit… application when used on a practical, recurring basis…" As an example of the complexity of a typical timetabling problem, Badri, et al. (1998) proposed model at the institution he studied "…consists of 252 decision variables, 66 goal constraints and system constraints." Even when a well-understood model can be formulated, course scheduling can require factoring a larger problem into at least two smaller sub-problems that are then solved sequentially (Hertz and Robert, 1998). This multi-stage approach is seen as needed even when scheduling a single College within a University (Stallaert, 1997). As recently as 2002, one College at a University identified course scheduling as "…a major problem for the school…[And]…the root cause of [other, major logistical problems]" (Hinkin and Thompson, 2002). Fundamental variables that have to be managed by a department chair manually (even if they are not globally applied), such as faculty preferences for consecutive classes, were not included in prior, but ostensibly comprehensive, models (Hinkin and Thompson, 2002).

Academic course scheduling or timetabling has been traditionally viewed as a constraint satisfaction problem (Janssen, 1995). Various techniques including linear programming, logic programming and genetic algorithms), self-adaptive algorithms and heuristic-based approaches have been used to resolve conflicts in course scheduling problems. However, most of these algorithms assume that the constraints are already available at the central location performing the course scheduling.

Academic course scheduling is a complex operation that requires the interaction between different users including instructors and course schedulers to satisfy conflicting constraints in an optimal manner. Academic course scheduling has improved room utilization, significant reduction in unassigned courses, and a consistent approach to time period shifts. In addition, the model greatly reduced the time necessary to produce an acceptable schedule.

Traditionally, this problem has been addressed as a constraint satisfaction problem where the constraints are available at the central location that performs the course scheduling. Here, we address academic course scheduling in a networked environment using intelligent agents within a decision support framework. The general problem of scheduling faculty to courses, to classrooms, and to time of day is faced by every educational institution. From the faculty's point of view, the objective is to maximize their preferences including the room and time of day; from the administration's point of view, the efficient utilization of the physical facilities is a concern as well.

Course Allocation Scheduling

The general problem of scheduling faculty to courses, to classrooms, and to time of day is faced by every educational institution. From the faculty's point of view, the objective is to maximize their preferences including the room and time of day; from the administration's point of view, the efficient utilization of the physical facilities is a concern as well. The problem of scheduling faculty and subject assignments is well documented in the references and can be viewed as a network optimization model with the usual constraints of requiring that all scheduled sections be staffed; a maximum, and perhaps a minimum, number of assignments for a faculty member. The classroom and time assignments introduce the additional constraints that a faculty member cannot be assigned more than one course per time period; a room cannot be assigned more than one class per time period. The considerations of classroom and time assignments are important preferences for the faculty. The faculty may express preferences for certain times of day, certain days of the week, back-to-back scheduling, and to avoid certain times of day.

Lecture Scheduling

The general problem of scheduling faculty, courses, time slots, and classrooms has attracted a great deal of interest. Numerous solution procedures have been proposed and tested. Each approach is designed to address certain aspects of the general scheduling problem. Andrew and Collins (1971) suggested a linear programming model; Dyer and Mulvey (1976) pro-posed a network model in the context of an integrated decision system. Large-scale integer programming models have been developed by Tillet (1975), Breslaw (1976), and McClure and Wells (1984). None of these optimization models considers the problem of assigning a faculty, subject, or room combination to a particular time slot.

Academic Scheduling Environment

Universities world-wide tend to have a hierarchical structure consisting of faculties, degrees, and courses. Faculties are the basic administrative units, each responsible for normally a single scientific discipline in terms of offering study programs and courses related to it. Multidisciplinary faculties, in case their disciplines were grouped for merely administrative reasons, can be further divided into sub faculties, called units, to process each discipline separately. Provision of both the administrative (faculties) and the scientific (units) structure enables distinction between interdisciplinary and interfaculty relationships for better decision making.

The general problem of scheduling faculty, courses, time slots, and classrooms has attracted a great deal of interest. Numerous solution procedures have been proposed and tested. Each approach is designed to address certain aspects of the general scheduling problem. Andrew and Collins (1971) suggested a linear programming model; Dyer and Mulvey (1976) proposed a network model in the context of an integrated decision system.

Time Slot Assignment

The academic lecture scheduling is a weekly scheduling for all the classes of a school, avoiding teachers meeting two classes at the same time, and vice versa. This means that an event may be placed in the timetable only in such a way that it does not violate constraints.

Figure.2 shows the concepts of academic scheduling implementation at schools. Lessons in a course are taught by a lecturer to a corresponding class of students and the academic scheduling is a problem of allocating resources, i.e. assigning to lecturers and class of students, time slots and lessons. A time slot is a period and a lesson is an event associating a lecturer, a course and a class of students with in a time slot.

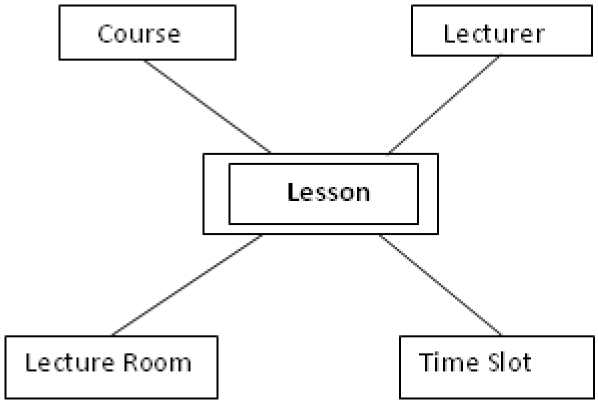


Figure 2. Concept of academic scheduling

Source: DegifTeka 2008

Procedure of the existing system

Academic scheduling is a major administrative activity in any academic institution. A number of courses taught by the corresponding lecturers are allocated into a number of available lecture rooms and a number of timeslots, subject to constraints.

The tasks that are considered in constructing the academic schedule are:

- Assigning periods to classes. There is a need to spread out lessons across the teaching cycle as much as possible, e.g. to avoid clashing of lessons on the same day.
- Some classes need 'double periods' (preferably 2 consecutive periods).

The academic scheduling problem is said to be feasible if and only if it satisfies the following constraints:

- Every lecturer and every class must be present in the academic schedule in a predefined number of periods;
- There cannot be more than one lecturer in the same class at the same period;
- No lecturer can be assigned to more than one class at the same time;

Problem Identification

Over the years, scheduling of academic activities has been a major problem in the universities and educational institutions at large. Indiscriminate fixing and clashing of lectures is now very common.Insufficient lecture rooms are the major factor contributing to the problem. This situation may be due to several factors,including a poorly managed and uncoordinated academic scheduling practices, etc.

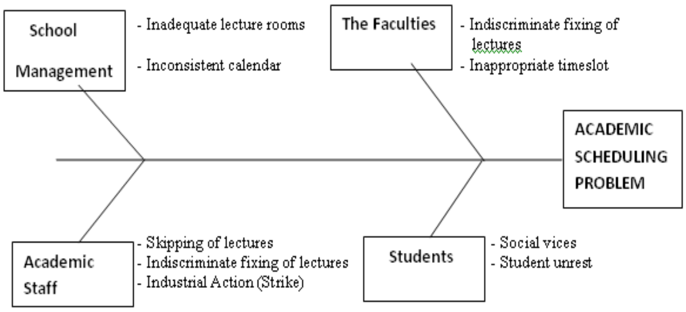


Figure 3. Ishikawa Diagram representing problems identified in Academic Scheduling

Source: Author

Design and Implementation

System design is the representation of how the real system is to function. In the design of this system, HTML tags were used to design the user interface with CSS for page formatting; JavaScript was also used to complement HTML for user interactivity.PHP served as the server-side scripting language, it was used to communicate with the host server. MySQL was used as the back-end database. The preferences for PHP and MySQL are because of their cross-platform and open source nature.

Specifications

Main Menu

Below is the proposed system's dashboard or control centre.

Input Design

This section shows different input designs of the proposed system. The input specification supplies data needed for course scheduling as well as the program that tells the computer how and what to do with the provided data.

Lecturer-Course-Time Input Design

Schedule-Courses-to-Classroom Input Design

Database Specification

The main goal of this database design is to give is a conceptual representation of the data structures that are required by a system to function. The design of a database is depicted as a special model called a database schema. Below are the database schemas for the various database files used**.**

Admin_LoginFile(Admin Log-in file for system's administrator)

Field	Logical data type	Physical data type	Key attribute
Username	Fixed length	VARCHAR	NK (Non-Key Attribute)
Password	Variable length	VARCHAR	NK (Non-Key Attribute)

LCTFile(Lecturer-Course-Time file)

Field	Logical data type	Physical data type	Key attribute
Id	Fixed Length	INT	PK (Primary Key)
Title	Fixed length	VARCHAR	NK (Non-Key Attribute)
Firstname	Variable length	VARCHAR	NK (Non-Key Attribute)
Lastname	Variable length	VARCHAR	NK (Non-Key Attribute)
Course_code	Variable length	VARCHAR	NK (Non-Key Attribute)
Time_slot	Variable length	VARCHAR	NK (Non-Key Attribute)
Venue	Variable length	VARCHAR	NK (Non-Key Attribute)
Days	Variable length	VARCHAR	NK (Non-Key Attribute)

Courses File

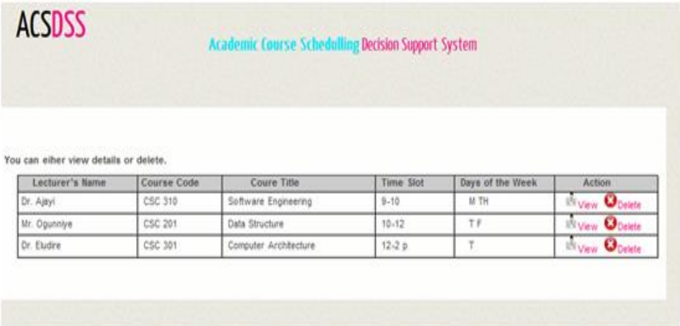
Field	Logical data type	Physical data type	Key attribute
Id	Fixed Length	INT	PK (Primary Key)
Course_code	Fixed length	VARCHAR	NK (Non-Key Attribute)
Course_title	Variable length	VARCHAR	NK (Non-Key Attribute)
Course_level	Variable length	VARCHAR	NK (Non-Key Attribute)
Course_units	Variable length	VARCHAR	NK (Non-Key Attribute)
Course_type	Variable length	VARCHAR	NK (Non-Key Attribute)
Time_slot	Variable length	VARCHAR	NK (Non-Key Attribute)

Classchart File

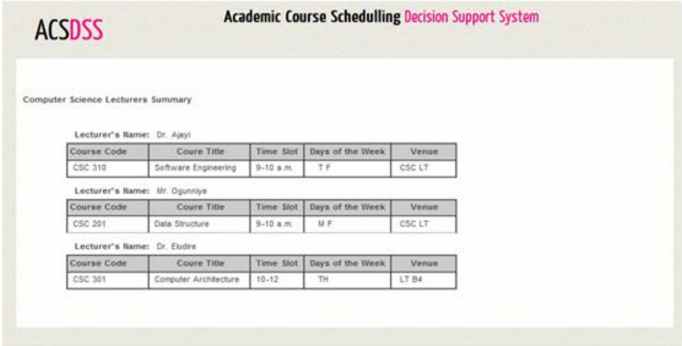
Field	Logical data type	Physical data type	Key attribute
Id	Fixed Length	INT	PK (Primary Key)
Course_code	Fixed length	VARCHAR	NK (Non-Key Attribute)
Course_level	Fixed length	VARCHAR	NK (Non-Key Attribute)
Course_unit	Fixed length	VARCHAR	NK (Non-Key Attribute)
Time_slot	Variable length	VARCHAR	NK (Non-Key Attribute)
Venue	Variable length	VARCHAR	NK (Non-Key Attribute)
Days	Variable length	VARCHAR	NK (Non-Key Attribute)

Interface Specification

View L-C-T Interface Design

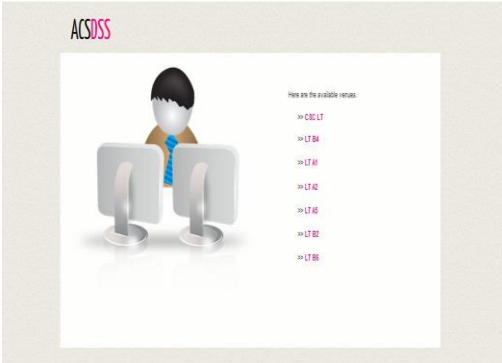


Lecturers Summary Interface Design



Output Specification

Course and Venue Schedule Output



Conclusion

As academic institution increasingly digitized their mode of operations, it has become pertinent for them to move their operations and services to a new paradigm by wholly embracing the use of decision support system for course scheduling. As part of management activities, issues relating to conflicts of access to lecture venues (theatres) should not be taken with levity hand in higher education setting. To aid this, management of higher institutions of learning can embrace this ‘robust’ and ‘unbiased’ decision making system in an attempt to curb the problem of course scheduling.

References

Ackoff J.H. (1967). “An Integrated Optimization/Information System for Academic Planning,” **Management Science,** Vol. 22, No. 12, pp.1332-1341.

Badri, G., Leonard, M., Haberle, C., & Lipschultz, W. (1998).Technology and Academic Advising. Academic Advising News.

Clemen G. (1996). A Rule-Based Expert System Approach to Academic Advising. Innovations in Education and Teaching Janssen M.K. (1995). The Next Generation of Computer-Assisted Advising and Beyond, NACADA Journal, 16 (1), 47-50.

Hertz, F.H. and Robert, S.A.(1995). A decision support system for academic advising. Paper presented at the 1995 ACM symposium on Applied computing, February 26-28, Nashville, TN, USA.

McCosh, V.B. (2004). “Interactive Scheduling: Historical Survey and State of the Art,” **AIIE** Hinkin T. and Thompson D. (2002).Applying a web-based expert system to the selection of a academic major.

M. Marte (2002). Models and Algorithms for School Timetabling, A Constraint-Programming Approach, Ph.D. dissertation, an der Fakult¨atf¨urMathematik, Informatik und Statistik der Ludwig-Maximilians-Universit ¨at M¨unchen, July, 2002.

Marakas, G. M. (1998). Decision Support Systems In The Twenty-First Century. Upper Saddle River, N.J.: Prentice Hall.

Marek J. Druzdzel and Roger R. Flynn (2002). Decision Support Systems.

Power, D. J. (1997). "What is a DSS?" The On-Line Executive Journal for Data-Intensive Decision Support.

Sprague, R. H. and E. D. Carlson (1982). Building Effective Decision Support Systems. Englewood Cliffs: N.J.,Prentice-Hall. ISBN0-130-86215-0.

Suleiman K. Kassicieh, Donald K. Burleson and Rodrigo J. Lievano (1986). Design and Implementation of a Decision Support System for Academic Scheduling.

Stallaert, C.L.; Guyote, M.J. and Keen, P.G.W. (1978). “Setting Priorities for DSS Development,”

Transactions, Vol. 10, No. 3 (September 1978), pp. 331-337. Hackathom, R. and Keen, P., “