This is to certify that the

thesis entitled

CONSTRUCTION OF A DEMONSTRATION PROTOTYPE OF AN EXPERT SYSTEM FOR SCHEDULING PART-TIME STUDENT WORKERS IN A UNIVERSITY RESIDENCE HALL FOODSERVICE OPERATION

presented by

POHSIANG TSENG

has been accepted towards fulfillment
of the requirements for

Master's degree in Institutional Admin.

Carol Sawyer
Major professor

Date May 24, 1990

# CONSTRUCTION OF A DEMONSTRATION PROTOTYPE OF AN EXPERT SYSTEM FOR SCHEDULING PART-TIME STUDENT WORKERS IN A UNIVERSITY RESIDENCE HALL FOODSERVICE OPERATION

By

Pohsiang Tseng

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Food Science and Human Nutrition

1990

# ABSTRACT

## CONSTRUCTION OF A DEMONSTRATION PROTOTYPE OF AN EXPERT SYSTEM FOR SCHEDULING PART-TIME STUDENT WORKERS IN A UNIVERSITY RESIDENCE HALL FOODSERVICE OPERATION

By

Pohsiang Tseng

The purposes of this study were to (1) obtain objective time data on manual scheduling of part-time student workers in a university residence hall foodservice operation, (2) construct an expert system demonstration prototype for scheduling part-time student workers in a university residence hall foodservice operation to demonstrate feasibility of use of expert systems in foodservice management, (3) develop a method for evaluation of the expert system, and (4) evaluate the demonstration prototype of the expert system in a laboratory environment. Time data recorded by residence hall schedulers for three terms were used to determine total time needed for scheduling manually per term; mean times for scheduling one student worker per term at three residence halls ranged from 16.66 to 62.76 minutes. The demonstration prototype has completed weekday scheduling in a laboratory environment. This result suggested that expert systems technology could effectively be applied to schedule part-time student workers in a university foodservice operation.

**Dedicated to My Parents,**
**Mr. and Mrs. Tseng**

ACKNOWLEDGMENTS

The author wishes to express her sincere thanks and deepest gratitude to her advisor, Dr. Carol A. Sawyer for her guidance, assistance, and encouragement in her graduate study. The author also wishes to thank Mr. Robert Hauser, the knowledge engineer of this project, for his understanding, assistance, and advice.

Genuine appreciation is extended to the members of Master's Degree committee, Dr. John A. Partridge, Dr. Jon Sticklen, and Ms. Burness G. Wenberg, for reviewing and commenting on this manuscript.

Sincere thanks and appreciation for their help are given to Mr. Mike Gardner who is the Food Service Manager at Brody Complex, Mr. Justin Terry and Douglas Fox who are supervisors at Brody Complex, Mr. Scott Pierpont who used to be the supervisor at Mason-Abbot Hall, and Ms. Anita Starks who is the supervisor at Shaw Hall. The author is also grateful to her friends and colleagues, especially to Ms. Renee Polega and Mr. Mung-Sen Sheu, for their help and encouragement.

Last but not least, the author wishes to express deepest appreciation to her parents, Mr. W.K. Tseng and Mrs. C. H. Tseng, and her host parent, Janice Cummings, for their love and support.

# TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

Chapter I

# INTRODUCTION

In warehouse/distribution facilities, supermarkets, retail stores, mining operations, hospitals, and accounting firms, computerized labor scheduling has been viewed as one way to improve manual labor scheduling. Computerized labor scheduling has successfully reduced labor-related expenditures through a combination of efficiency improvements and personnel cuts, enhanced employee morale and the firm's operations (Fensholt, 1988; Britton, 1987; Desai, 1987; Burns and Carter, 1985).

Published information on computerized labor scheduling in the foodservice industry is limited. Some software application programs have been identified. In 1984, the CBORD Group (Ithaca, NY), a leading supplier of institutional foodservice software, developed **Labor Scheduling System**, a multi-functional labor scheduling application program in a price of around $1000. This system could be used to aid in the planning of employee work schedules and to analyze the resulting labor costs. In 1986, **SUPERSKED** (Management Robotics, Inc.), in a price of $1395, was developed to determine work force and scheduling employees. Other similar software in the price range of $995 to $2000 was: **CIDER System** (Dining Data System; Benicia, CA), **LABORSERVE** (Practorcare Inc.; San Diego, CA), **Labor Scheduling** (RestaurantComp;

Larkspur, CA), and **People-Planner Scheduler** (Information Marketing Businesses Inc.; Cambridge, MA). These programs were characterized by the ability to schedule full-time employees; they were not capable of handling part-time employees with "intermittent" availability (e.g. an employee was available from 9:00 am to noon and 2:00 pm to 4:00 pm).

In the foodservice industry, large numbers of student workers were hired on a part-time basis by various segments, especially in colleges and universities. Residence hall foodservice personnel supervisors, such as those at Michigan State University (MSU), had found labor scheduling to be time-consuming. Each student worker must be matched with work shifts that did not conflict with his/her class schedule. Since personnel supervisors were also responsible for other duties, such as supervising full-time employees and part-time student workers, part-time student supervisors were employed to perform some scheduling tasks.

The current study focused on the computerized scheduling of part-time student workers in residence hall foodservice at MSU. Results of this study would contribute to the further development of computerized part-time labor scheduling for the foodservice and other industries.

Conventional computer software as well as expert systems have the potential to be used to construct computerized labor scheduling programs. Conventional software, to date, has not been equipped with the features described above. The strong point of conventional software in general has been the ability to use highly complex algorithms to sort through information (words, numbers, data) in thousands of database records and then read them to determine a final answer. The labor scheduling problem was not amenable to an algorithmic solution. On the other hand, expert systems were heuristic in nature. That is, they required the use of rules of thumb to achieve acceptable solutions for sufficiently narrow problems (Roadmmer and White, 1988). In addition, expert systems represented knowledge in a structured way that showed the relationships among knowledge, which could make the process of problem solving more efficient (reducing the time spent in processing unrelated knowledge; Waterman, 1986).

From this point of view, expert systems would seem to be a better choice for computerized scheduling of part-time student workers in the residence hall foodservice than conventional software. Thus, the expert system technique was used to develop a computerized program for scheduling part-time student workers in a residence hall foodservice at MSU.

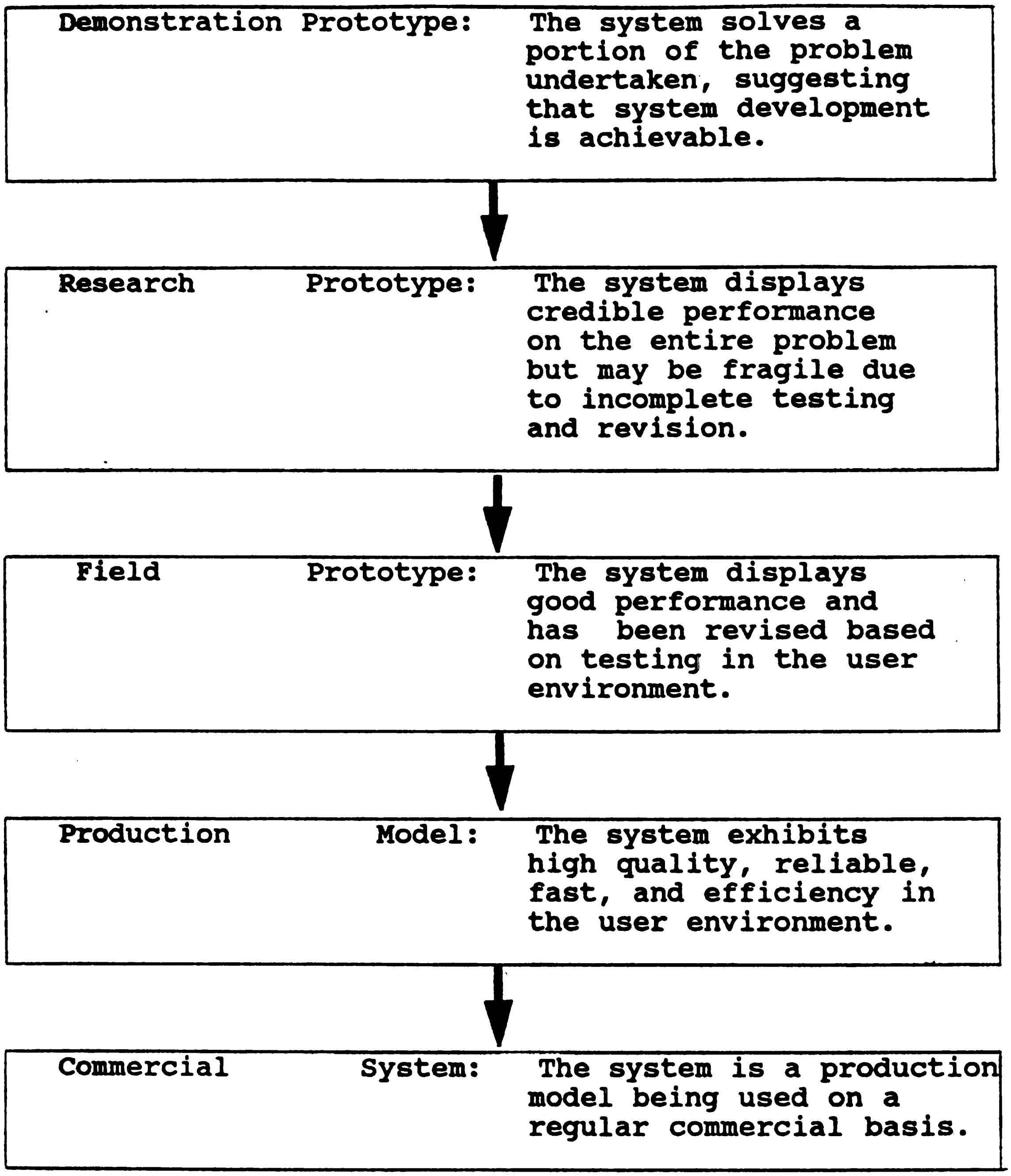
DSPL (Design Structures and Plans Language), an expert system building tool, was developed to perform problem solving of routine design in which standard methods of completing the task were fixed and well-known (Brown, 1987). Scheduling part-time student workers was not a typical routine design, but it may stretch the boundaries of routine design (Kern et al., 1989). Thus, DSPL was chosen to construct this expert system.

Testing and evaluation, important elements in the development of expert systems, enabled a feedback process to take place whereby the comments served as a basis for iterative refinements of the expert system (Liebowitz, 1986). During the development of an expert system, testing in a laboratory environment was usually needed before the system could be released for field testing (test in the user environment; Waterman, 1986). Therefore, methods for testing and evaluation of the expert system in both the laboratory and user environment should be developed at the time the system is being designed.

Also, to be able to compare the expert system to the manual system, a study was done to determine the amount of time needed for scheduling student workers manually. Data generated from time study were used to evaluate the completed expert system.

The evolution of an expert system could be divided into five stages: demonstration prototype, research prototype, field prototype, production model, and commercial system (Figure 1; Waterman, 1986). Thus, the objectives of this study were as follows:

1. To obtain objective time data (e.g. the amount of time used in term scheduling) on manual scheduling of part-time student workers in a university residence hall foodservice operation.




[a] Waterman (1986).

Figure 1. The evolution of an expert system (Waterman, 1986).

2. To construct an expert system demonstration prototype for scheduling part-time student workers in a university residence hall foodservice operation to demonstrate feasibility of use of expert systems in foodservice management.
3. To develop a method for evaluation of the expert system.
4. To evaluate the demonstration prototype of the expert system in a laboratory environment.

## Chapter II

## LITERATURE REVIEW

The literature review for the current study was divided into five parts: labor scheduling, time study, expert systems, and building an expert system.

### 2.1 Labor Scheduling

Discussion of the labor scheduling problem was divided into five parts: labor scheduling warehouse/distribution facilities, supermarkets, and grocery stores; underground mine planning labor scheduling; the public accounting firm labor scheduling; single shift scheduling with variable demands; and labor scheduling in the foodservice industry.

#### 2.1.1 Labor Scheduling in Warehouse/Distribution Facilities, Supermarkets, and Grocery Stores.

In warehouse/distribution facilities, supermarkets, and grocery stores, the major labor scheduling problem supervisors had faced was unable to use workers effectively (Fensholt, 1988). First, supervisors often did not know exactly how much time each task should take. Therefore, they were unable to ensure that the right number of people were located in the right place at the right time, and that the appropriate amount of work was assigned to each worker. Often employees found themselves with "leftover" empty hours and subsequently job efficiency decreased. Also, after supervisors spent many of their own working hours plotting labor schedules manually, they had an insufficient amount of time remaining to follow up on the performance of employees.

Based on the observations above, measuring and recording the amount of time each task should take may be a solution for such a labor scheduling problem. However, two problems may arise. First, it was difficult to identify all of the various tasks and then to give each worker the required number of working hours when a large number of employees and tasks were involved. Second, the time requirement for tasks varied with the number of orders and customers. When the number of customers or orders increased, so did the time requirement of tasks. Therefore, to react to changing demands by re-adjusting schedules in short periods of time was very difficult (Fensholt, 1988).

Computerized labor scheduling programs have been used in real world applications (warehouses, supermarkets, and grocery stores); two examples were the Baum system (Eric C. Baum Consulting Firm; Chicago, IL) and COMPU-SKED system (Compu-Sked Co.; Fort Lee, N.J.). These two programs could automatically calculate exactly how many people were needed per shift and print out individual picking orders that told each employee just how many tasks he/she was responsible for and how much time they had to finish each task. In this way, a warehouse/distribution facility has been able to reduce its labor-related expenditures by at least 25% through a combination of efficiency improvements and personnel cuts. About 200 Compu-sked installations, at Roundy's, Rice Supermarkets (Houston), Bells (Rochester, N.Y.), Crook's (Nashville) and others, had reported a fast return on their investment through a decrease in labor costs because labor hours (300 to 500 labor hours) were cut (Fensholt, 1988).

### 2.1.2 Labor Scheduling in Underground Mine Planning

The workplace of the average underground miner was filled with unseen hazards, tiring work tasks, and long periods of isolation. It was important to make an equitable arrangement (schedule) for all mining workers according to their experience, job skill required, their performance, acceptable levels of risk, etc. However, to make an equitable arrangement for mine managers or superintendents with so little time before shift changes was difficult (Britton, 1987).

An expert system (called CHOOZ and written in Turbo Prolog software), developed by Tanoma Coal Co. N.V., was to automatically search and compile work crews, premium work lists, and suggest strategies on deploying a work force for its maximum productive effort and achieving an equitable arrangement. At the same time, CHOOZ was also expected to provide varying levels of information based on acceptable risk, which would be valuable to the ever changing physical environment of underground mining (Britton, 1987).

## 2.1.3 Labor Scheduling in the Public Accounting Firm

Scheduling was an important administrative, planning, and operational function of the public accounting firm. When a scheduling staff was unable to consider all scheduling factors appropriately (e.g. staff utilization, manpower requirements, personnel skills, individual development), the firm's resources would be mis-allocated through the overuse of some staff and the underuse of others. Such inefficiency was frustrating for all involved. Administrators complained about the lack of chargeable time. Underemployed professional staff were dissatisfied because they were not challenged, while overemployed staff became burned out due to the heavy workload.

Since the foundation of a public accounting firm was its professional staff, a CAP firm incorporated the microcomputer into the manual scheduling process to enhance the firm's operations and reduce employee dissatisfaction. This system would provide a higher outlook on enhancing operations and employee satisfaction (Desai, 1987).

## 2.1.4 Single Shift Schedules with Variable Demands

In recent years, labor negotiations for seven-day-week organizations such as hospitals, mining companies and chemical industries have involved an increased emphasis on improving shift schedules. A study by Burns and Carter (1985) had given an exact lower boundary on the number of workers required to satisfy contractual commitments, such as ensuring that each employee received at least A out of every B weekends off, everyone worked exactly five days per week, and no one worked more than six consecutive days. A linear time algorithm was presented, which generated schedules satisfying all the primary objectives (Burns and Carter, 1985).

2.1.5 Labor Scheduling in the Foodservice Industry

Published information on computerized labor scheduling in the foodservice industry is limited. Some software application programs have been identified. In 1984, the CBORD Group (Ithaca, NY), a leading supplier of institutional foodservice software, developed **Labor Scheduling System**, a multi-functional labor scheduling application program in a price of around $1000. This system could be used to aid in the planning of employee work schedules and to analyze the resulting labor costs. In 1986, **SUPERSKED** (Management Robotics, Inc.), in a price of $1395, was developed to determine work force and schedule employees. That is. this program could determine how many people were needed on each shift based on "sales" (presumed indicator of amount of business. Then this program filled in employees on each shift (e.g. 6:00 am to 3:00 pm was one shift) according to their skill levels, maximum number of work day per week, maximum number of shifts per week, etc.). Other similar software in the price range of $995 to $2000 was: **CIDER System** (Dining

Data System; Benicia, CA), **LABORSERVE** (Practorcare Inc.; San Diego, CA), **Labor Scheduling** (RestaurantComp; Larkspur, CA), and **People-Planner Scheduler** (Information Marketing Businesses Inc.; Cambridge, MA).

However, these programs were characterized by the ability to schedule full-time employees; they were not capable of handling part-time employees with "intermittent" availability (e.g. an employee was available from 9:00 am to noon and 2:00 pm to 4:00 pm).

In the foodservice industry, large numbers of student workers were hired on a part-time basis by various segments, especially in colleges and universities. Residence hall foodservice personnel supervisors, such as those at Michigan State University (MSU), had found labor scheduling to be time-consuming. Each student worker must be matched with work shifts that did not conflict with his/her class schedule. Personnel supervisors were also responsible for other duties, such as personnel management, determining the amount of work force for next term, and adjusting the number of shifts. Personnel supervisors could be better able to use their time and abilities on the really tough problems if they could be relieved from handling scheduling problem which, for the experts, was routine (Briggs and Doney, 1988).

Therefore, efforts should be made to computerize the scheduling of part-time student workers in university foodservice operations.

2.1.5 Summary

Even though labor scheduling problems varied among different industries, computerized labor scheduling had been most often thought of as the best way to improve current manual labor scheduling. Existing applications had proved that computerized labor scheduling was more beneficial than manual scheduling in terms of reducing labor-related expenditures through a combination of efficiency improvements and personnel cuts, reducing employee dissatisfaction, and enhancing the organization's operations.

However, current applications were characterized by scheduling full-time employees. They were not capable of handling large numbers of part-time employees with "intermittent" availability (e.g. an employee was available from 9:00 am to noon and 2:00 pm to 4:00 pm).

Part-time employment has grown over most of the post-WWII period; more than 15% of all adults in non-farm civilian jobs work less than 35 hours a week (Otten, 1990). Therefore, efforts should also be made to schedule part-time employees by extending current available software or developing new programs.

Residence hall foodservice personnel supervisors, such as those at Michigan State University (MSU), had found labor scheduling to be time-consuming. To be better able to use their time and abilities on the really tough problems, personnel supervisors should be relieved from handling scheduling problem which, for the experts, was routine. Therefore, computerized scheduling of part-time student workers in the university residence hall foodservice operation would be needed in the future.

## 2.2 Time Study

Decisions regarding human work activities were often based on subjective beliefs rather than objective data. Work study could provide objective data about activities concerning people, facilities, and equipment as a means to improve those activities (Currie, 1977).

Two distinct, yet interdependent, approaches were used in work study: (1) method study that referred to the way in which work was done and (2) work measurement that pertained to the time required to complete a particular task (Block et al., 1985). One purpose of this study was to determine the amount of time needed for scheduling student workers. Therefore, in the section below, attention was given to the work measurement approach. Discussion of the work measurement approach is presented in two phases (Sections 2.2.1 and 2.2.2).

### 2.2.1 The First Phase of Work Measurement

The first phase of work measurement was to identify **work function activities** of the object studied (e.g. labor scheduling) and then to classify them into steps referred to as **elements** (Mundel, 1978; Block et al., 1985).

In the study by Block et al. (1985), **work function activities** were described as a group of similar activities that may be recognized by sight and may be considered homogeneous. **Work function activities** were categorized into three major groups: direct work, indirect work, and delays. Direct work functions were defined as any essential activity that contributed directly to the production of the end product. Indirect work functions were any catalytic activity that contributed to the production of the end product. Delays were all times when an employee was scheduled to be working but was not engaged in either direct or indirect work functions. The time study of a vegetable pre-preparation unit, an example, is shown in Table 1 (Block et al., 1985).

An **element** consisted of a unified group of motions, such as taking hold of an object, moving it, and placing it (Table 2). Each **element** was the smallest practical unit for time measurements and had the well-defined end point for timing (Mundel, 1978).

### 2.2.2 The Second Phase of Work Measurement

The second phase of work measurement was to choose a work measurement technique. Work measurement techniques that had been used were survey, work sampling, Master Standard Data (MSD) quantity food production code, and MM-3 dietary

**Table 1. Work function activities in a university residence hall vegetable pre-preparation unit[a]**

| | |
|---|---|
| Direct Work Function | **Pre-preparation**<br><br>carrots<br>celery<br>lettuce, head and leaf<br>onions, yellow mature<br>all other fruits and vegetable |
| Indirect Work Function | **Transportation of Food**<br><br>**Transportation of Equipment or Supplies**<br><br>pot and pan washing<br>housekeeping<br>instruction |
| Delay Time | **Personal delay**<br><br>meal time<br>break time<br><br>**Idle Time** |

[a] Block et al. (1985)

Table 2. Elements for carrot pre-preparation in a university residence hall vegetable pre-preparation unit[a]

| Work Function | Element |
|---|---|
| Pre-preparation | 1. move out of storage<br>2. open sack<br>3. remove from sack<br>4. peel and nub<br>5. wash<br>6. place in container<br>7. cover<br>8. move into storage<br>9. remove onto rack<br>10. weigh<br>11. remove from container<br>12. cut into sticks |

[a] Block et al. (1985)

methodology (Mundel, 1978; Zemel and Matthews, 1982; Olsen and Meyer, 1987).

The survey work measurement technique involved collecting data from predetermined samples with a questionnaire or worksheet, then analyzing data and calculating labor time spent in each practical unit (Mundel, 1978; Olsen and Meyer, 1987). Two steps for the development of the worksheet were as follows:

1. Develop a description of the **elements** of each work function and check these against the requirements for good **elements** (e.g., having easily detected and defined beginning/end points for timing, well-unified group of motions) for time study elements.
2. Adjust the **elements** as necessary and then detail the descriptions of the **elements** one by one.

Work sampling involved nonsequential observations at random times; the standard number of minutes that a qualified, properly trained, and experienced person required to perform a specific task when working at a normal pace could then be determined (Olsen and Meyer, 1987).

MSD quantity food production code was the simulation of production based on predetermined time values for specific conditions. Scaled layout and standardized production formulas were used to predetermine production time (Zemel and Matthews, 1982).

In MM-3 dietary methodology, data needed to calculate production time included number of batches of product produced per week, number of portions per week, and distances traveled. These data were entered onto forms in the MM-3 Dietary Methodology Manual to estimate production time (Zemel and Matthews, 1982).

### 2.2.3 Summary

The labor intensity of hospital foodservice, coupled with constantly rising labor costs, demanded that managers carefully allocated labor resources to the production of menu items to contain costs (Zemel and Matthews, 1982). In the public sector or institutional foodservice segment, access to information and research opportunities had been far greater than from the private sector (Olsen and Meyer, 1987). Therefore, work measurement had been a common approach used in the published research on productivity in institutional foodservice operations.

While studies of food production in the foodservice industry had been helpful in gaining the understanding of productivity, other functions performed by foodservice workers and supervisors, such as service and scheduling workers, had largely been ignored (Olsen and Meyer, 1987). Thus, information was limited on the amount of supervisor time required to schedule student workers in residence hall foodservice. Such information would be useful to gain an insight into the related labor (supervisor) costs and usefulness of facilitating scheduling by computer.

## 2.3 Expert Systems

Expert systems were characterized by a collection of general strategies that used knowledge in such a way that the complexity inherent in certain tasks was minimized (Firdman, 1986). Thus, expert systems were also viewed as very specialized computer programs which were created by using extensive, high quality, specific knowledge about some narrow problem area.

Expert systems differed from other computer programs in that they were characterized by goal-directed behavior (Firdman, 1986). Given the goals and strategies which exploited typically heuristic knowledge about the problem domain for goal achievement, expert systems were able to explore very large problem spaces to make plausible alternative options that contributed to goal achievement effectively and efficiently. For example, the design of an air-cylinder (AC) consisted of 19 requirements (e.g., rod diameter, air inlet diameter, air pressure). The values for each requirement may vary with different applications. Thus, this design problem could be viewed as a search problem in a large space. Each requirement of this space reflected a possible candidate for the answer to the design problem. Given a goal (e.g. a specific application) and strategies for the goal achievement, a subset of the 19 requirements with specific values was able to be identified without exhaustive search in the entire problem space.

The computational methods of conventional computer programs were not engaging in the exploration of underlying problem space to make possible hypotheses by use of explicit knowledge and general exploration strategies.

### 2.3.1 Rule-Based Expert Systems

The knowledge in a rule-based expert system was organized in a way that separated the knowledge about the problem domain from the system's other knowledge, such as general knowledge about how to solve problems or knowledge about how to interact with the user (Waterman, 1986). Therefore, an expert system usually consisted of two elements: a knowledge base and an inference engine (Figure 2).

The knowledge base, as shown in Figure 2, contained facts (e.g. student class schedules, job priority), and rules/or other representations. The rule (or other representation) referred to the type of knowledge that used those facts as the basis for decision making (e.g. if the student worker did not have class right before or after the work shifts, then assign the work shifts to the student worker) (Waterman, 1986; Roberts, 1988).

The inference engine contained an interpreter that decided how to apply the rules to infer new knowledge and a

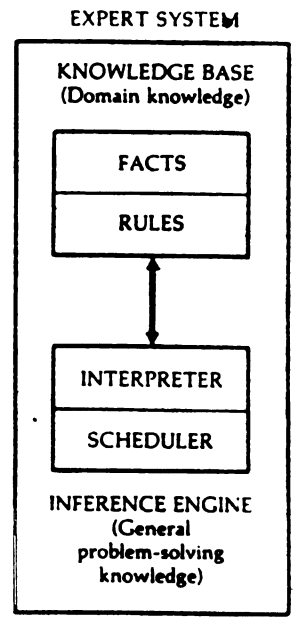



Figure 2. The structure of knowledge base and inference engine (Waterman, 1986).

scheduler that decided the order in which the rules (or other representations) should be applied. That is, the inference engine used a predefined control strategy to find the enabled rules or other heuristics and decided which one to apply (Boritz and Brown, 1986).

### 2.3.2 Generic Tasks: Expert Systems Beyond Rules

The separation of knowledge about the problem domain from that knowledge used in inference processes had resulted in the field of expert systems being stuck at too low a level of abstraction that obscured the essential nature of the tasks that current systems perform. For example, the fact that MYCIN (a medical diagnosis expert system) engaged in some form of classification problem solving was not readily visible at the level of the rule representation used to make inferences (Brown and Chandrasekaran, 1989).

The roots of this problem were two-fold. First, the designer had to undertake a complex programming effort in order to make the translation of the form in which knowledge was needed for design task from that of diagnosis. Second, seeking uniform mechanisms came at a cost: the architectures that supported this uniformity did so by suppressing the distinctions in control and inference between different kinds of tasks (e.g. routine design task, diagnosis task; Brown and Chandrasekaran, 1989).

The available paradigms often forced us to fit the problem to the programming language rather than to fashion the programming language to reflect the structure of the problem. Since structures and control regimes for one specific domain (e.g. medical diagnosis) continued to be different from other domains (e.g. labor scheduling), efforts had been made to propose an alternative level of abstraction for expert system building. This alternative level of abstraction was called **generic task**. The **generic task** was to reflect a theory specific for one type of problem solving (e.g. scheduling) in the expert-system-building tool (Chandrasekaran, 1983; Chandrasekaran, 1986; Brown, 1987).

DSPL was one of the efforts to reflect the theory of the problem solving of routine design tasks. DSPL had been thought of as a "task level" language. In this way, DSPL had been able to provide exactly the knowledge types and control structures necessary for the task; and the knowledge engineers were more easily able to express the domain knowledge in that language, which dramatically reduced the time required to develop a system (Brown, 1987).

### 2.3.3 Introduction to DSPL

Design Structures and Plans Language (DSPL), developed by Brown and Chandrasekaran (1989), was a programming language for designing expert systems that performed problem solving on routine design tasks.

Routine design was defined as a type of task in which standard methods of completing the task were fixed and well-known. Also, the task had been done many times before, each time with different but similar requirements, and would be done in the future again and again. For example, the design of an air-cylinder (AC) consisted of 19 requirements (e.g. rod diameter, air inlet diameter, air pressure). With different applications, the AC needed to be designed and manufactured again each time according to a subset of the 19 requirements. On the other hand, all requirements may be involved in the AC design but the values for each requirement would vary with different applications. Nevertheless, the method of designing an AC was fixed and repeated each time.

DSPL provided two useful knowledge representation facilities: agents (e.g. specialist, plan) and a design database. Domain knowledge was encoded both in the agent structure (inference structure) and the design database. The procedural knowledge (e.g. goals and strategies for solving scheduling problem) was reflected in the agent structure; while the declarative knowledge, used in agent structure (e.g. job priority), was represented in the design database (Brown and Chandrasekaran 1989).

Agents, such as specialist, plan, etc., were organized into a hierarchy. This hierarchy acted as a reasoning mechanism: the top levels of this hierarchy were **specialists** representing the more general concepts (e.g., designing the spring of AC); while the lower levels dealt with more specific instances of those concepts (e.g. showing how to design the spring, checking the requirements for design, designing the spring desired). Based on this hierarchy, conclusions were able to be generated.

The control regime for the DSPL hierarchy was top-down. The following was done recursively until a complete design was worked out: a **specialist** corresponding to a component (e.g. weekday scheduling) of the task (e.g. term scheduling) was called; the **specialist** (e.g. weekday specialist) in turn suggested further **sub-specialists** (e.g. lunch sub-specialist) to call to set other details of the design. Failures were passed up until appropriate changes were made by higher level **specialists**, so that **specialists** who failed at the first attempt may succeed on a re-try (Chandrasekaran, 1986).

DSPL provided a way of writing declarations of agents (e.g. specialist), which allowed the programmer to represent the knowledge easily. After all agents required were declared and checked, DSPL allowed its underlying system to link them by a top-down control regime and construct a hierarchy. Then the problem solving could be invoked by requesting a design from the top-most specialist. After a successful completion, the design data-base would contain the results of the completed design (Brown and Chandrasekaran 1989).

### 2.3.4 The Advantages of Expert Systems

Basically, the advantages of expert systems could be viewed from the following aspects:

1. Expert systems were able to mimic the reasoning process of human beings.
2. Expert knowledge became distributable and permanent.
3. Expert systems could reduce the time needed to solve the problem and reach a conclusion.

First, as in number one above, when human experts solved a problem (e.g. decision-making problem), they did not do it by solving sets of equations or performing other laborious mathematical computations. Instead, they chose symbols to represent the problem concepts and applied various strategies and heuristics to manipulate these concepts (Waterman, 1986).

Expert systems understood not just symbols, or information, but the information and its interconnections. Therefore, performance of expert systems was closer to that of human beings.

Second, expert systems could facilitate the preservation and dissemination of scarce expertise by encoding the relevant experience of an expert and making it generally available as a resource. In this way, they were able to help less expert professionals (e.g. trainees) improve the quality of their judgments, decisions and, possibly, their general problem-solving skills across many locations (Lippert, 1987; Boritz and Brown, 1986).

Third, an expert system had a perfect memory and represented the knowledge in an explicit and intelligible manner. Thus it could help compensate for limitations in human information-combining abilities and minimize biased interpretations and incorrect inferences to drastically reduce the time needed for problem solving (Boritz and Brown, 1986).

## 2.4 Building an Expert System

The evolution of an expert system had been divided into five stages: demonstration prototype, research prototype, field prototype, production model, and commercial system

(Figure 1; Waterman, 1986).

The expert system development could be viewed as five highly interdependent and overlapping phases: identification, conceptualization, formalization, implementation, and testing and evaluation (Buchanan et al., 1983). The **identification phase** involved identifying the type and scope of the problem, expert system building tools, and the goals or objectives of building an expert system. The **conceptualization phase** involved deciding concepts, relations, and control mechanisms needed to describe problem solving in the domain. The **formalization phase** involved expressing the key concepts and relations in some formal way, suggested by the expert system building language. These two phases, conceptualization and formalization, coupled together, could also be thought of as knowledge acquisition (Kidd, 1987; Waterman, 1986). The **implementation phase** turned the formalized knowledge into a working computer program. Finally, the **testing and evaluation phase** involved evaluating the performance and utility of the prototype program and revising it as necessary (Waterman, 1986).

### 2.4.1 The Identification Phase

The **identification phase** involved identifying the type and scope of the problem, and expert system building tools. As to the type of problems, the problem targeted for the expert system development required the use of heuristics (e.g.

rules of thumb, strategies, etc.). Problems involved in the use of heuristics were difficult to attack by conventional approaches, but may be amenable to expert system methodologies. In addition, the problem targeted for expert system development did not require knowledge from a very large number of areas. If it did, the amount of knowledge needed for the expert system would be probably beyond its acceptable limits. Also, it was difficult to combine very heterogeneous knowledge. Attempts to aggregate knowledge across a number of areas were certain to mask the true structure of that knowledge (Prerau, 1985; Waterman, 1986).

As to the problem designed for expert system development, the problem must be neither too easy (e.g. taking a human expert less than a few minutes) nor too difficult (e.g. requiring more than a few hours for an expert). If the problem was too easy, the development of the system would not warrant the effort; if too difficult, the amount of knowledge needed may be beyond the state of the art in knowledge base size. In other words, the problem targeted should be sufficiently narrow and self-contained: the aim was not for a system that was expert in an entire domain, but for a system that was an expert in a limited task within the domain (Prerau, 1985; Waterman, 1986).

Expert system building tools were programming languages and support packages used to build the expert system. Expert system building tools made it possible to develop an expert system in less time than would be required with the use of traditional development languages (Gevarter, 1987). However, selecting the right tool that made the development of an expert system easy and time saving was difficult because most tools had been developed to handle only one particular class of problem (e.g. rule-based problem solving). On the other hand, AI researches were not really sure what tool features were required by specific classes of problems. Despite the fact that there was no easy answer, some basic guidelines for deciding what tool was appropriate for a specific problem task were suggested by AI researchers (Waterman, 1986; Lippert, 1987).

1. Would the tool have the features suggested by the needs of the problem?
2. What level of nesting of rules and goals would be allowed?
3. What kind of inference procedure had been included? Which would be dominant in this tool?
4. Would the tool be reliable?
5. Would the tool have the features suggested by the needs of the application?

## 2.4.2 Knowledge Acquisition

Knowledge acquisition was the process of transferring knowledge from the expert to the computer. One view of knowledge acquisition had been to consider the expert's head as being filled with bits of knowledge and the problem of knowledge acquisition as being one of "mining those jewels of knowledge out of their heads one by one." In short, knowledge acquisition involved acquiring the right kinds of knowledge from the expert, mapping that knowledge into a coherent organizational structure, and then translating the mapped knowledge into a formal representation suggested by the expert system building tool to replicate the expert's knowledge (Naughton, 1989).

From this point of view, three major functions in knowledge acquisition could be given (Breuker and Wielinga, 1987).

1. The elicitation of domain knowledge. Elicitation was to "mine the knowledge out of experts heads."
2. The analysis of elicited knowledge. Analysis was the transformation of knowledge into an interpretative framework.
3. Knowledge representation. The organized knowledge was translated into a formal representation suggested by the expert system building tool.

2.4.2.1 The elicitation of domain knowledge. The interview was the most acceptable approach used in eliciting domain knowledge. There were three types of interview that had been used: (1) interview with "thinking-aloud" and "cross-examination", (2) on-site observation, and (3) intuitive interview.

Kuipers and Kassirer (1987) proposed an interview that began with a "thinking-aloud" segment and ended with a "cross-examination" technique. In this type of interview, the expert was asked to explain out loud as much as possible what he/she thought about as he/she solved a problem. Then the knowledge engineer asked probing questions about the expert's knowledge of particular topics. Thus, interview with "thinking-aloud" and "cross-examination" experiment was much more effective for determining the limits of the knowledge represented (Kuipers and Kassirer, 1987).

On-site observation relied on watching the expert solve realistic problems within the domain and not to say or do anything that might influence the expert's problem-solving approach. In other words, the knowledge engineer observed the expert solving real problems on the job rather than contrived problems in a laboratory setting; the knowledge engineer did not interfere but rather acted as a passive observer. This approach gave the knowledge engineer some insight into the complexity of the problem and into the type of interface facility needed by the user to interact with the finished system in the field (Waterman, 1986).

For the intuitive interview, the knowledge engineer studied and interacted with both experts and the literature of a field in order to become familiar with problem-solving methods. Then the investigator developed a representation of expertise which was checked against the opinion of other experts (Waterman, 1986).

2.4.2.2 The analysis of elicited knowledge. The second function of knowledge acquisition was analysis of elicited knowledge. In this function, elicited data were translated into an interpretative framework. Breuker and Wielinga (1987) proposed an **interpretation model** that could be used for communication between expert and knowledge engineer to check consistency and completeness of elicited knowledge and, in particular, to facilitate the mapping of those data into structures.

There were five different levels of representation of knowledge involved in the **interpretation model** of Breuker and Wielinga (1987). These levels were knowledge identification, knowledge conceptualization, epistemological analysis, logical analysis, and implementational analysis.

At the level of **knowledge identification**, individual concepts were identified. To reduce the amount of data, individual concepts were organized by identity (e.g. type of knowledge, such as classification knowledge). Later on, at the level of **knowledge conceptualization**, concepts became integrated, according to a number of relations between concepts (e.g. is-a and part-of).

At the level of **epistemological analysis**, the analysis uncovered structured properties of expertise by identifying and examining the following five types of information: object, knowledge source, model, structure, and strategy.

**Objects** were the input or conclusions (output) of inference processes; a **knowledge source** was a processor that inferred new objects from given objects; **models** were parts of supporting inference knowledge; **structures** could be sequences, resemblance groupings,or hierarchies along with some relations; and a **strategy** was a plan invoking problem-solving in the domain.

At the level of **logical analysis**, a framework expressing the relations among knowledge was formed for use at the **implementational analysis** level.

Finally, at the **implementational analysis** level, all levels previously described were used to make up the mechanism which then was used for facilitating knowledge representation.

2.4.2.3 Knowledge representation. The third function of knowledge acquisition was knowledge representation. In this function, key concepts, sub-problems, and control features were mapped into a more formal representation suggested by an expert system building tool. An expert system building tool may support one or more methods for representing knowledge. From this view, methods for representing knowledge varied with the expert system building tool chosen.

2.4.3 Testing and Evaluation of Expert Systems

Testing involved evaluating the performance and utility of the expert system prototype and revising it as necessary (Waterman, 1986). The performance referred to the accuracy of embedded knowledge and any advice or conclusions that the software provided; the utility was defined as whether the software produced useful results, the credibility of its results, its reliability, its efficiency and speed, its ease of interaction, and the extent of its capabilities (Gaschnig et al., 1983).

Evaluation, an important element in the development of expert systems, could uncover problems with knowledge representation (e.g. relations and missing concepts), unwieldy control mechanisms, interface, or conclusions represented at the wrong level of details. Such problems could suggest possible improvements for the current developed system. Therefore, evaluation enabled a feedback process to take place whereby the comments served as a basis for iterative refinements of the expert system (Liebowitz, 1986).

During the development of expert systems, several tests in a laboratory environment were usually needed before the system could be released for field testing. As the field testing was completed, new complications may arise and take additional time to correct. Thus, re-testing in a laboratory environment would again be required, and then field testing would be repeated (Waterman, 1986).

Two major tasks were involved in evaluation: the development of criteria for the evaluation of the test problems and the determination of evaluation approaches (Harrison, 1989; Miller et al., 1985). These two tasks are discussed below (Sections 2.4.3.1 and 2.4.3.2).

2.4.3.1 Establishing criteria. A review of the literature indicated that various criteria had been used to evaluate the quality of expert systems (Liebowitz, 1986). Boehm et al. (1978) studied the development of quality software and grouped characteristics of quality software into seven categories: portability, reliability, efficiency , human engineering, testability, understandability, and modifiability. Table 3 summaries the criteria in each category (Liebowitz, 1986).

Gaschnig et al. (1983) also identified evaluation characteristics of expert systems. The four characteristics identified by these authors included quality of the system decisions and advice, correctness of the reasoning techniques used, quality of the human-computer interaction (both its content and the mechanical issues involved), system's efficiency, and cost effectiveness.

Additionally, Miller et al. (1985) proposed four major sub-tasks for use in the evaluation of a commercial expert system: (1) evaluating the accuracy of the system, (2) evaluating the system sizing and the performance of hardware

Table 3. The characteristics of quality software*

| CATEGORY | CHARACTERISTICS |
|---|---|
| Reliability | Completeness: all of the software parts are present and each of its parts is fully developed.<br>Accuracy : its outputs are sufficiently precise to satisfy their intended use.<br>Consistency : it contains uniform notation, terminology, and symbology within itself, and its content is traceable to the requirements. |
| Portabillity | Device-independent: software can be executed on computer hardware configurations other than its current one. |
| Human-Engineering | Communicativeness : it facilitattes the specification of inputs and provides outputs whose form and content are easy to assimilate and useful. |
| Testability | Structuredness : it possesses a definite pattern of organization of its independent parts. |
| Efficiency | Device Efficiency : it fulfills its purpose without waste of resources.<br>Accessibility : it faciilitates the selective sue of its components. |
| Understandability | Self-descriptiveness: it contains enough information for a reader to determine its objectives, assumptions, constraints, inputs, outputs, components, and status.<br>Conciseness : no excessive information is present.<br>Legibility : its function and those of its component statements are easily discerned by reading the code. |
| Modifiability | Augmentability: it easily accommodates expansions in data storage requirements or component computational functions. |

* Boehm et al. (1978); Liebowitz (1986).

and software to find the lowest cost of implementation, (3) evaluating the quality of human/computer interaction, and (4) economic evaluation (Does the action carried out by the system justify the costs of hardware, software, training and maintenance of developed system).

Because initial goals and requirements of expert systems differed, criteria also varied with each expert system developed.

2.4.3.2 Evaluation approaches. A common approach used for evaluation of expert systems was the judgment of experts. There were two ways to use such expert judges. The first way was to use judges (experts) by giving them the output from the system and asking them to evaluate the results using pre-determine criteria. The second way was to use blind evaluation. That is, both experts and the developed system were asked to solve some test problems. The results may be put into a standard format and the judges (experts) were asked to evaluate the results without knowing whether the results were those of an expert or from the computer system (Chandrasekaran, 1983; Harrison, 1989).

Gaschnig (1982) described several evaluations of the Prospector System (an expert system used in medical diagnosis). These evaluations relied on comparisons of the evaluations by Prospector and by human experts. The data from each judge were gathered using a questionnaire that required numeric answers reflecting the degree of agreement or disagreement on a scale of -5 to +5 with regard to each question given. The results were converted into an overall score that reflected the goodness of fit. Then, the scores from the expert were compared with the Prospector scores (Harrison, 1989).

There had been no generally accepted method for evaluating expert systems, although some work in this direction had begun (Harrison, 1989; Liebowitz, 1986). In lieu of generally accepted methods, the three principles of evaluation, proposed by Gaschnig et al. (1983), could be used as a guide for evaluating expert systems.

1. Complex objects or processes could not be evaluated by a single criterion or number.
2. The larger the number of distinct criteria evaluated or measurements taken, the more information would be available on which to base an overall evaluation.
3. People would disagree about the relative significance of various criteria according to their respective interests.

The same problems of reliability and validity that made behavioral and cognitive evaluations of a human being difficult also made the evaluation of expert systems difficult (Harrison, 1989). Additionally, when expert systems were applied to the domains in which it was not easy to ascertain correct answers or in which there was disagreement about what the correct answer was, the evaluation would become difficult to perform (Miller et al., 1985).

Chapter III

# MATERIALS AND METHODS

The purposes of this study were to (1) obtain objective time data (e.g. the amount of time used in term scheduling) on manual scheduling of part-time student workers in a university residence hall foodservice operation, (2) construct an expert system demonstration prototype for scheduling part-time student workers in a university residence hall foodservice operation to demonstrate feasibility of use of expert systems in foodservice management, (3) develop a method for evaluation of the expert system, and (4) evaluate the demonstration prototype of the expert system in a laboratory environment.

Discussion of methods used in this study was divided into two parts: (1) collection of time data on manual scheduling of part-time student workers and (2) developing a demonstration prototype of an expert system for scheduling part-time student workers in a residence hall foodservice operation at Michigan State University (MSU). The first part was conducted by the researcher during Spring Term 1989, Fall Term 1989, and Winter Term 1990. The second part was conducted using a team approach; the researcher and a knowledge engineer worked together to lead to complete the demonstration prototype.

## 3.1 Collection of Time Data on Manual Scheduling of Part-time Student Workers

The main objective of collection of time data on manual scheduling of part-time student workers was to determine the amount of supervisor time needed for scheduling student workers each term (4 terms per year) at Michigan State University (MSU) and then use these data to evaluate the completed expert system. Based on the study of Block et al. (1985) and Olsen and Meyer (1987), collection of time data on manual scheduling of part-time student workers in the residence hall foodservice operation at MSU was divided into four steps: (1) developing the worksheet for collection of time data, (2) determining the sampling frame, (3) conducting the collection of time data, and (4) analyzing the data.

### 3.1.1 Developing the Worksheet for Collection of Time Data

The purpose of the worksheet was to enable personnel supervisors or their delegates to record time spent in scheduling part-time student workers.

As discussed in section 2.2.1, work measurement started with identifying **work function activities** and then classifying them into **elements** that were smallest units and able to facilitate time measurement (Block et al, 1985). Therefore, to develop the worksheet for collection of time data on manual scheduling of part-time student workers, the food manager and personnel supervisor at Brody Complex, MSU, were contacted and given brief descriptions of the purpose and methods of this study. A follow-up meeting with the food manager and personnel supervisor at Brody Complex was then scheduled by the researcher to identify scheduling functions (term, final, and volunteer scheduling), elements of each scheduling function, and descriptions of the elements. A rough list of scheduling functions, elements, and descriptions was then formed and checked by the personnel supervisor at Brody Complex and other two experts, personnel supervisors at Mason-Abbot and Shaw Hall, to make sure each element was distinct, describable, and measurable.

Finally, the worksheet (Table 4) and coding guides (Table 5, 6, and 7) were formulated for collection of time data on manual scheduling of part-time student workers. Each scheduling function was assigned a two-digit code number, as shown in Table 4. As indicated in Tables 5, 6, and 7, each element was assigned a four-digit code number: the first two digits indicated the scheduling function and the last two digits indicated the element within the specific scheduling function.

### 3.1.2 Determining the Sampling Frame

Three residence halls were chosen among all of the residence halls at Michigan State University (MSU), East Lansing, MI, based on the number of students staying in each

Table 4. Worksheet for collection of time data on manual scheduling of part-time student workers

SUPERVISOR TIME REQUIRED TO SCHEDULE STUDENT WORKERS

Hall ________________ Term ________________ Year ________________

| Code Number and Supervisor Levels | Scheduling Function |
|---|---|
| I Full-Time Personnel Supervisor | 01 Term Scheduling |
| II Part-Time Student Supervisor | 02 Final Scheduling |
| III Part-Time Student Worker as a Dishroom Supervisor | 03 Volunteer Scheduling |
| IV Part-Time Student worker as a Student Secretary | |

| Date | Supervisor (code) | Time Start/End | Scheduling Function (code) | Element (code) | Description (code) |
|---|---|---|---|---|---|
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

Notes:

Table 5. Code Number and Elements for Term Scheduling

| Code Number | Element | Description |
|---|---|---|
| 0101 | Organize to Schedule | 1. Go through all applications of student workers and sort them into pre-defined groups (e.g. lunch, breakfast, and dinner groups; returning and non-returning groups).<br>2. Check the job type and number of work shifts; add, delete, or remain the same.<br>3. Check the in/out time of each work shift.<br>4. Other (please specify). |
| 0102 | Schedule | 1. Match student workers with available work shifts by considering their past experience and performance, job, work time, and work day preferences, number of work hours desired, in/out time of available work shifts, alternate weekend, and class schedules.<br>2. Other (please specify). |
| 0103 | Check Schedule | 1. Make sure critical jobs are full.<br>2. Count the number of open and filled work shifts.<br>3. Look for student workers who can work extra to fill open work shifts.<br>4. Other (please specify) |
| 0104 | Copy Schedule | 1. Copy the schedule for use of student worker.<br>2. Copy the schedule for other purposes.<br>3. Prepare and send letters to student workers.<br>4. Other (please specify). |

Table 5 (cont'd.).

| | | |
|---|---|---|
| 0105 | File Schedule | 1. Supervisors file completed student schedules.<br>2. Other (please specify). |
| 0106 | Reschedule | 1. Re-do the scheduling when student workers are unable to complete their work shifts as scheduled due to illness, bad performance, drops and adds of classes, or other conflicts.<br>2. Other (please specify). |
| 0107 | Delegation | 1. Supervisors delegate the scheduling to other people.<br>2. Other (please specify). |
| 0108 | Other | (please specify) |

Table 6. Code Number and Elements for Final Scheduling

| Code Number | Element | Description |
|---|---|---|
| 0201 | Organize to Schedule | 1. Collect the final exam schedule sheet filled out by student workers.<br>2. Check the job type and number of work shifts; add, delete, or remain the same.<br>3. Check the in/out time of each work shift.<br>4. Other (please specify). |
| 0202 | Schedule | 1. Match student workers with available work shifts by considering their present position, performance, job, work time, and work day preferences, in/out time of available work shifts, and exam schedules.<br>2. Other (please specify). |
| 0203 | Check Schedule | 1. Make sure critical jobs are full.<br>2. Check if there are three work shifts assigned to each student worker.<br>3. Count the number of open and filled work shifts.<br>4. Look for student workers who can work extra to fill open work shifts.<br>5. Other (please specify) |
| 0204 | Copy Schedule | 1. Copy the schedule for use of student worker.<br>2. Copy the schedule for other purposes.<br>3. Prepare and send letters to student workers.<br>4. Other (please specify). |

Table 6 (cont'd.).

| | | |
|---|---|---|
| 0205 | File Schedule | 1. Supervisors file completed student schedules.<br>2. Other (please specify). |
| 0206 | Reschedule | 1. Re-do the scheduling when student workers are unable to complete their work shifts as scheduled due to illness, bad performance, or other conflicts.<br>2. Other (please specify). |
| 0207 | Delegation | 1. Supervisors delegate the scheduling to other people.<br>2. Other (please specify). |
| 0208 | Other | (please specify) |

Table 7. Code Number and Elements for Volunteer Scheduling

| Code Number | Element | Description |
|---|---|---|
| 0301 | Organize to Schedule | 1. Determine the job type and number of work shifts for holiday or special dinner.<br>2. Check the in/out time of each work shift.<br>3. Post the master schedule determined on board and ask student workers to sign up.<br>4. Other (please specify). |
| 0302 | Schedule | 1. Supervisors adjust and complete the master schedule.<br>2. Other (please specify). |
| 0303 | Check Schedule | 1. Make sure critical jobs are full.<br>2. Count the number of open and filled work shifts.<br>3. Look for student workers who can work extra to fill open work shifts.<br>4. Other (please specify) |
| 0304 | Copy Schedule | 1. Copy the schedule for use of student worker.<br>2. Copy the schedule for other purposes.<br>3. Prepare and send letters to student workers.<br>4. Other (please specify). |
| 0305 | File Schedule | 1. Supervisors file completed student schedules.<br>2. Other (please specify). |

Table 7 (cont'd.).

| | | |
|---|---|---|
| 0306 | Reschedule | 1. Re-do the scheduling when student workers are unable to complete their work shifts as scheduled due to illness or other conflicts.<br>2. Other (please specify). |
| 0307 | Delegation | 1. Supervisors delegate the scheduling to other people.<br>2. Other (please specify). |
| 0308 | Other | (please specify) |

hall. The premise was that scheduling part-time student workers in residence halls of various sizes could experience different problems. The three residence halls chosen were: Brody Complex, the largest with 2640 students and hiring an average of 288.33 student workers (N=3; Spring 1989, Fall 1989, and Winter 1990); Shaw Hall, middle with 920 students and hiring an average of 125 student workers (N=3; Spring 1989, Fall 1989, and Winter 1990); and Mason-Abbot Hall, the smallest with 750 students and hiring an average of 79.33 student workers (N=3; Spring 1989, Fall 1989, and Winter 1990).

### 3.1.3 Conducting the Collection of Time Data

Three separate meetings with personnel supervisors in each of three halls (one meeting/hall) were held to explain how the worksheet was to be used. First, scheduling work functions and elements in each scheduling work function were explained by the researcher. Then, personnel supervisors were shown how to use the coding guides with 24 elements (Tables 5, 6, and 7). Finally, a demonstration of how to fill out the worksheet (e.g. recording date, supervisor levels, the time of starting and the time of stopping scheduling) was given. In summary, personnel supervisors were asked to record the time of starting and ending scheduling when they performed a specific element of each of three scheduling functions.

Weekly visits with personnel supervisors at the three residence halls were set up to collect the worksheets (Table 4). The collection of time data was completed during Spring Term 1989, Fall Term 1989, and Winter Term 1990 to obtain an estimate of the amount of time spent in scheduling student workers for one school year (most residence halls at MSU were closed to students during Summer Term).

### 3.1.4 Analyzing the Data

Summarized data to be obtained from the worksheets were the amount of time spent in a specific element of one scheduling function per term. Then, the amount of time needed for each scheduling function per term (hours), the total time needed for scheduling student workers per term (hours), and the mean time needed for scheduling one student worker per term (min) were calculated.

## 3.2 Developing a Demonstration Prototype of an Expert System for Scheduling Part-Time Student Workers in a Residence Hall Foodservice Operation at Michigan State University (MSU)

### 3.2.1 The Characteristics of Manual Scheduling of Part-Time Student Workers

The residence hall foodservice at Michigan State University employed a large number of student workers on a part-time basis. Scheduling these student workers was characterized by the following features.

1. The scheduling of part-time student workers in residence hall foodservice operations (SPRF) had student constraints and job constraints. **Student constraints** included class and activity schedules, job preference (including position, time, and day preference), experience, number of work hours desired per week, and alternate weekend desired (weekend A or weekend B). **Job constraints** included job priority (critical jobs were on the top priority), in/out time of work shifts, and meal priority (lunch was scheduled first, then breakfast, and finally dinner).
2. SPRF dealt with a large number of students and work shifts. Because of class conflicts, most of the student workers could not work continuously for 12 hours. Therefore, each job (12 hours/job) must be broken into three shifts (breakfast, lunch, dinner) during the day. Also, on the average, student workers desired 12-15 hours per week. Thus, to fill all work shifts, the supervisor must employ a large number of student workers.
3. A master schedule was constructed each term. Each term, new student workers must be integrated into the scheduling; new class schedules of retained student workers must also be accommodated.

4. Since final exam schedules were different from regular class schedules, all student workers needed to be rescheduled for the final week each term.
5. Volunteer scheduling was needed for holidays (e.g. Mother's day, Easter) and for work periods between terms when student workers would not normally be available to work. The work shifts needed to be fill were determined by personnel supervisors and posted on a bulletin board. Student workers were asked to sign up if they were available at the time. Then personnel supervisors adjusted and completed the schedule.

Based on the features described above, much time would seem to be needed to implement the sophisticated matching between the large quantity of student workers and the large number of work shifts. Further, scheduling part-time student workers seemed to be a problem of routine design. Updating **student constraints** (e.g., class schedules, adding new student workers) and **job constraints** (e.g., changing in/out times, job priorities, etc.), rather than major changes in reasoning or decision-making processes, was all that was required each term. Therefore, personnel supervisor time could be saved if this type of routine design could be performed by a computer program.

## 3.2.2 Rationale for Choice of Conventional Software vs. Expert Systems

When personnel supervisors did SPRF (an abbreviation of "scheduling of part-time student workers in the university residence hall foodservice operation), they usually broke this problem into a set of tasks (e.g. decomposing term scheduling into weekday and weekend scheduling), worked on each task in a predetermined order (e.g. scheduling weekday first, then weekend), and then combined the schedules generated from each task into the master schedule (e.g. combining weekday and weekend scheduling to form the term schedule) (DHFS, 1988).

SPRF could be thought of as a problem solving process that tried to satisfy multiple constraints resulting in a goal state. For instance, one of the goal states was that there was no conflict between student workers' class schedules and work shifts scheduled. To meet this goal, personnel supervisors needed to check students' class schedules and in/out time of work shifts and saw if they had a class during, right before, or right after the time of the shift being considered.

The scheduling process may be repeated several times because all constraints could not be satisfied on the first attempt. The process of matching student workers and available jobs may be repeated by only considering one or a small subset of constraints instead of all constraints at one time. For example, one student worker had experience as a exit host (HE), the lowest priority job. According to the experience constraint (scheduling student workers to the jobs they had done before), this student should be scheduled to the HE. At the same time, the job priority constraint that student workers should be scheduled into critical jobs first must also be considered. In such a situation, this student worker could not meet those two constraints at the first attempt since HE was the lowest priority job. Therefore, only one constraint could be considered in the mind of the personnel supervisor at that time, instead of both constraints (experience and job priority). Usually the personnel supervisor would schedule this student worker into one of critical jobs rather than the job (HE) he/she had previous experience on, because filling all critical jobs was more important than placing student workers in jobs for which they had previous experience.

Therefore, to meet the needs of the scheduling task, a programming language should have the following features: (1) the ability to decompose a problem (scheduling) into tasks (e.g. term scheduling) and then combine them into a complete schedule, (2) the ability to consider multiple constraints in an effective manner (be able to reduce the time for scheduling), and (3) the ability to mimic personnel supervisors' (expert) reasoning processes, such as making an alternate plan if the original one could not satisfy all constraints.

Conventional software, to date, has not been equipped with the features described above. The strong point of conventional software has been the ability to manipulate data. That is, it could perform complex computations very fast; it could use highly complex algorithms to sort through information (words, numbers, data, or patterns) in thousands of database records and then read them to determine the final answer. Since the SPRF problem was not amenable to an algorithmic solution, conventional programming could be used to solve this problem, but the programming process would be complicated and time-consuming (Rauch-Hindin, 1988).

The primary purpose of expert systems was to perform tasks that only experts in the given domain could do. To achieve this purpose, expert systems were equipped with the ability to mimic the reasoning process of human experts. For example, information processing in expert systems was performed in a way that implied reasoning, inferring, and meaning as done by human experts. Therefore, an expert system may not only understand the concepts of "car," "start," and "gas," but also understand the relationship among them. Then it could infer "if a car will not start, it may be out of gas" (Rauch-Hindin, 1988). In addition, they represented information in a structured way that showed the relationships among each piece of information, which could facilitate the decomposition of a problem and make data processing more efficient (e.g., reducing the time spent in processing unrelated information; Rauch-Hindin, 1988).

Based on the above, expert systems seemed to be more qualified for computerized SPRF than conventional software.

### 3.2.3 Evaluating Suitability of DSPL for the SPRF Domain

Based on the characteristics of the SPRF problem (Section 3.2.1), a set of four features of expert system building tools for the SPRF domain were determined (Section 3.2.2). These features were problem decomposition, structured organization of tasks, constraint satisfaction in an effective manner, and failure handling (e.g. proposing alternate plans that would be proposed if the original plan was not satisfied; DHFS, 1988; Brown and Chandrasekaran, 1989).

As discussed in Section 2.3.3, DSPL provided a hierarchy that could facilitate problem decomposition. In addition, the hierarchy of DSPL was organized by a top-down control regime. Thus, all tasks could be linked together by a desired sequence, which in turn would be able to reduce the search space and increase the speed of problem solving. Additionally, DSPL provided the functions of failure handling and redesign. That is, if the problem could not be solved on the first attempt, the process of problem solving would be undertaken again by adjusting the original plan used in the first attempt.

Thus, DSPL was chosen as the system-building tool for the development of a SPRF expert system demonstration prototype.

### 3.2.4 The Development of a SPRF Expert System Demonstration Prototype

The process of building an expert system usually has involved interaction between the expert-system builder, knowledge engineer, and one or more human experts in some problem area. That is, the knowledge engineer has transferred knowledge from the human expert to a computer program and made the computer program solve problems in much the same manner as did the human expert.

However, an intermediary between the knowledge engineer and human expert (personnel supervisor) was needed by the present study. The human expert (personnel supervisor) was not available to work with the knowledge engineer for the entire time needed for expert system development for two reasons. First, the personnel supervisor was an entry-level position; people in this position were often promoted to higher positions after they have been in the position for approximate 6 months. Second, the turnover rate was high in university foodservice even at the supervisor level. Therefore, the researcher acted as an intermediary in this study.

Responsibilities of the researcher included gaining expertise on scheduling part-time student workers, communicating expertise gained to the knowledge engineer, and testing and evaluation of the SPRF expert system. Responsibilities of the knowledge engineer included knowledge representation and implementation of the SPRF expert system.

Based on the discussion in Sections 3.2.1, 3.2.2, and 3.2.3, the objectives for the development of a SPRF expert system demonstration prototype in DSPL were: (1) to construct a schedule of the same quality as the one constructed by the personnel supervisor at Brody Complex currently handling the task, (2) to greatly reduce time needed for scheduling student workers each term, and (3) to allow for easy modification of the master schedule so that the new schedule could be constructed easily each term by only updating **student constraints** and **job constraints** (e.g. student class schedules, job priority, job type).

The process of developing a SPRF expert system demonstration prototype in DSPL (Figure 3), based on the study of Buchanan et al. (Section 2.4), was divided into four phases: (1) identifying the scope of the SPRF problem, (2) knowledge acquisition, (3) implementation of the SPRF expert system, and (4) testing and evaluation of the SPRF expert system. The knowledge acquisition phase was to gain expertise on scheduling part-time student workers as done by the personnel supervisor at Brody Complex, MSU. The implementation phase was to turn the formalized knowledge into a working computer program. The testing and evaluation phase was to evaluate the SPRF expert system and revise it as necessary. Recycling through the various steps in the knowledge acquisition phase was needed to validate knowledge elicited from the personnel supervisor at Brody Complex and design

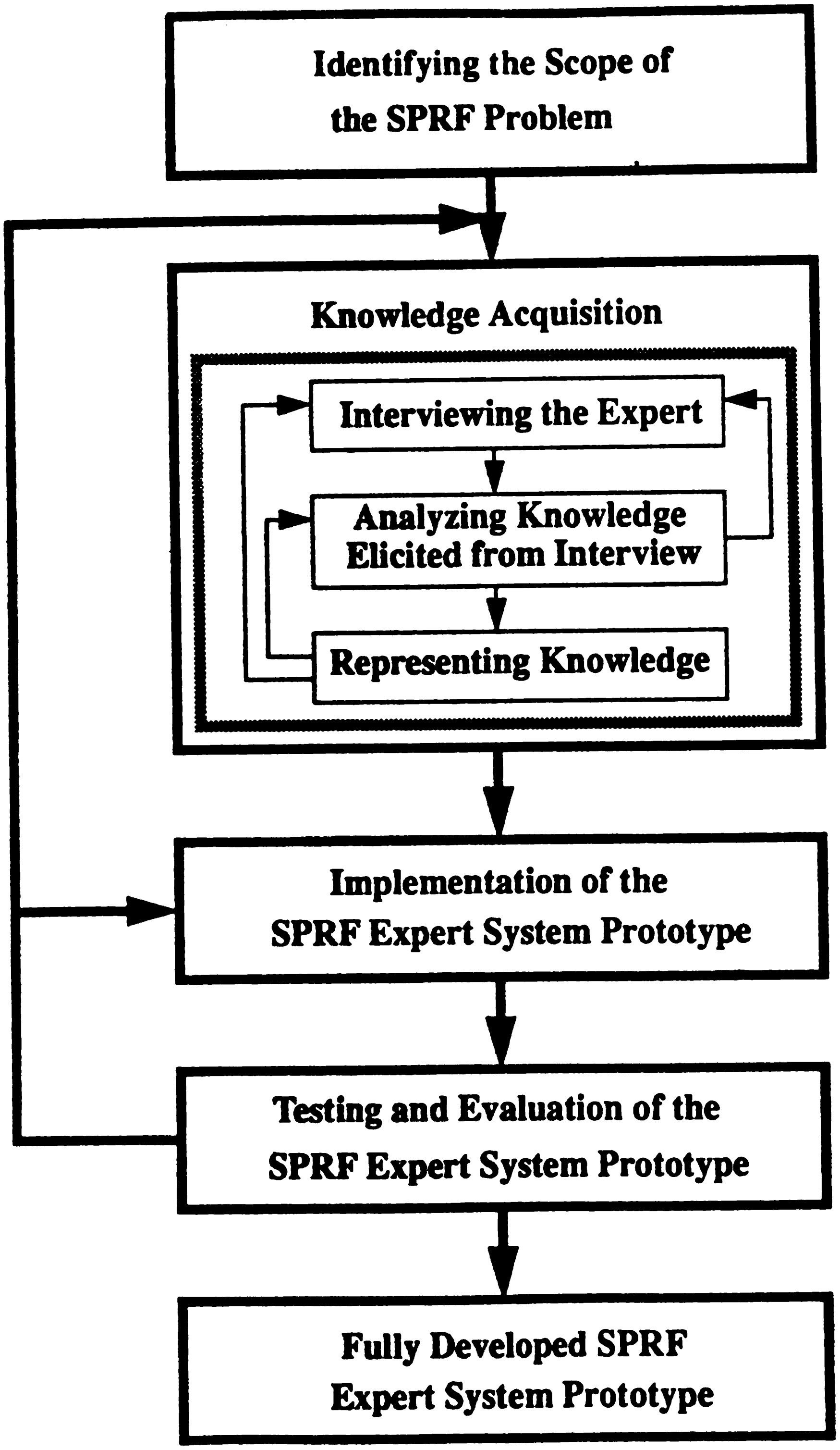



Figure 3. The process of developing a SPRF expert system demonstration prototype in DSPL.

concepts. Also, recycling, through the knowledge acquisition phase, implementation phase, and testing and evaluation phases, was needed to reformulate the concepts, refine strategies used in problem solving, and revise the control flow of problem solving until the objectives for the development of the SPRF expert system (discussed in the beginning of Section 3.2.4) could be achieved. In short, results obtained from the testing and evaluation phase would contribute to the improvements of the current knowledge acquisition and implementation phases. After improvements were implemented in the current knowledge acquisition and implementation phases, the testing and evaluation phase would be repeated to validate changes made and evaluate the performance of the currently improved SPRF expert system.

#### 3.2.4.1 Identifying the Scope of the SPRF Problem

As may be recalled from section 3.2.2, expert systems seemed to be more appropriate for computerized SPRF than conventional software. The problem targeted for expert system development should be of a manageable size that made the developed knowledge base interesting and also warranted the effort of developing the SPRF expert system (Section 2.4.1). Finally, the problem targeted for the SPRF demonstration prototype development should not require knowledge from a very large number of areas (Section 2.4.1).

The SPRF problem at MSU, taken as a whole, seemed not to be appropriate for expert system development for two reasons. First, the SPRF problem at MSU included both cafeteria and kitchen areas. Because those two areas were different in job types, in/out time of work shifts, etc., working on an expert system design for both areas at the same time would have made the problem scope unmanageable in size. Second, the method of performing SPRF varied slightly with personnel supervisors in different residence halls. Attempts to aggregate knowledge across different residence halls would have been certain to mask the true structure of scheduling methods.

To keep the scope from being beyond the acceptable limits of expert systems but still sufficiently broad to ensure that the software had practical application, scheduling part-time student workers for one area (cafeteria area) at one Hall (Brody Complex) at a time was considered for initial system development.

From this point of view, scheduling part-time student workers at the cafeteria area of Brody Complex was chosen as the domain for developing the SPRF expert system. The reason was that Brody Complex was the largest hall at Michigan State University and hired a greatest number of student workers (an average of 288.33 $\pm$ 63.55 student workers; N=3, Spring 1989, Fall 1989, and Winter 1990) of all MSU residence halls to fill approximate 1026 work shifts for the weekly schedule. The large number of jobs and student workers involved would make the SPRF expert system development more interesting and warrant the effort. Also, if this system was successful, it could be modified for use in smaller residence halls at MSU.

3.2.4.2 Knowledge Acquisition

The major purpose of knowledge acquisition in this study was to gain expertise on scheduling part-time student workers and then to use that knowledge as a basis to construct an expert system demonstration prototype. Since one of the objectives for the SPRF expert system development is to construct a schedule of the same quality as the one constructed by the personnel supervisor at Brody Complex, the SPRF expert system was developed to mimic the problem-solving process of the expert scheduler. Also, as discussed in section 3.2.4.1, the scheduling in the cafeteria area of Brody Complex was chosen as the domain for developing the SPRF expert system demonstration prototype. Therefore, the expert scheduler (the personnel supervisor at Brody Complex) was used as the resource for gaining expertise on scheduling part-time student workers.

The knowledge acquisition process of this study was divided into three steps: (1) interviewing the expert (the personnel supervisor at Brody Complex), (2) analyzing knowledge elicited from interview, and (3) representing knowledge.

Interviewing the expert, was undertaken to acquire knowledge about scheduling part-time student workers through direct interaction with the personnel supervisor at Brody Complex. An intuitive interview technique (discussed in Section 2.4.2.1) was used in this phase. The second phase, analyzing knowledge elicited, was conducted to map that knowledge into a coherent organizational structure. Finally, the mapped knowledge was translated into a formal representation suggested by the expert system building tool (DSPL) and refined until it replicated important and valuable parts of the knowledge of the personnel supervisor at Brody Complex.

3.2.4.2.1 Interviewing the expert. The approach, combining intuitive interview and on-site observation methods (discussed in 2.4.2.1), was used in this phase. The researcher talked with the personnel supervisor at Brody Complex to become familiar with the method used to schedule part-time student workers. During this interaction, the personnel supervisor introspected while solving a problem in front of the researcher. The researcher jumped in whenever it seemed appropriate, asking relevant questions to verify his basic scheduling method. The personnel supervisor was also asked to provide a step by step list of his scheduling process. At three follow-up meetings (one hour/meeting), the researcher and knowledge engineer questioned the personnel supervisor about his scheduling procedure to gain more detail about the method for scheduling part-time student workers.

3.2.4.2.2 Analyzing knowledge elicited from interview.

The purpose of this phase was to map the obtained knowledge from the personnel supervisor into an interpretative framework. Four tasks, derived from the study of Breuker and Wielinga (1987), were involved: (1) identifying knowledge, (2) conceptualizing knowledge, (3) constructing the reasoning process of scheduling, and (4) determining the structure of the scheduling process.

a. Identifying knowledge

Knowledge identification was mainly an information management task. A large amount of knowledge was gathered from the interview process, but the relative importance of each piece of knowledge was yet to be determined. Identifying knowledge provided an important reduction in the amount of knowledge through organizing it by knowledge type, such as classification, prioritization, and design. **Classification** referred to knowledge used to break a group of jobs or student workers into subgroups (e.g. knowledge for sorting student workers into lunch, breakfast, and dinner). **Prioritization** referred to knowledge used to rank jobs and student workers (e.g. grouping jobs according to job priority). **Design** referred to knowledge used implicitly by the scheduler in the scheduling process, such as decision-making strategies (e.g.

scheduling experienced student workers first; filling in critical jobs first).

b. Conceptualizing knowledge

The purpose of knowledge conceptualization was to integrate individual concepts into a conceptual framework representing the logical combination of tasks. For example, breakfast, lunch, and dinner scheduling (individual concepts) were integrated into weekday and weekend scheduling; weekday and weekend scheduling (individual concepts) were integrated into term scheduling. The result was that the concept of term scheduling was finally constructed.

c. Constructing the reasoning process of scheduling

The construction of a reasoning process was to organize reasoning strategies used by the personnel supervisor in a structured manner. For example, lunches were the hardest to staff so they were scheduled first, followed by breakfast and dinner. Weekend scheduling was done last because it was easy to staff.

d. Determining the structure of the scheduling process

Structural determination of the scheduling process was done to determine the scheduling structure by mapping the reasoning process into a conceptual framework. This structure facilitated interpreting the data and getting a fine-grained description of the architecture of the SPRF domain.

<u>3.2.4.2.3 Representing knowledge.</u> The knowledge representation was developed by the knowledge engineer. Appendix B, written by the knowledge engineer, shows how the knowledge was presented in the way suggested by DSPL.

In summary, DSPL provided two useful knowledge representation facilities, agents (e.g. specialist, plan) and a design database. Domain knowledge was encoded both in the agent structure (inference structure) and the design database. The procedural knowledge (e.g. scheduling method) was reflected in the agent structure; while the declarative knowledge, used in agent structure (e.g. job priority, in/out time of work shifts), was represented in the design database. The design database also provided the structures for storing the actual schedules that needed to be filled. Agents (e.g. specialist, plan) were organized into hierarchies in DSPL to break down the scheduling problem.

Major functions had a specialist associated with them along with their sub-specialists. The control regime for the DSPL hierarchy was top-down. The following was done recursively until a complete design was worked out: a **specialist** corresponding to a component (e.g. weekday scheduling) of the task (e.g. term scheduling) was called; the **specialist** (e.g. week specialist) in turn suggested further **sub-specialists** (e.g. lunch sub-specialist) to call to set other details of the design.

### 3.2.4.3 Testing and Evaluation of the SPRF Expert System

To evaluate the SPRF expert system, objectives with criteria plus two evaluation approaches were developed, but were not used in the present study due to time constraints. Since expert systems needed to be refined and tested in a laboratory environment before they could be released for field testing, two evaluation approaches were proposed: laboratory testing and field testing (user environment).

3.2.4.3.1 Evaluation criteria. As indicated in the beginning of Section 3.2.4.2, the SPRF expert system was developed to mimic the problem-solving process of the expert scheduler. Therefore, criteria used to evaluate manual scheduling of part-time student workers were also used to evaluate the SPRF expert system. Additional criteria used for evaluating expert systems were derived from the works of Boehm et al. (1978), Gaschnig et al. (1983), and Waterman (1986). These criteria were to evaluated expert systems in terms of their completeness, consistency, accuracy, ease of use, and the ability to update.

Finally, criteria for evaluating the SPRF expert system were formulated by combining criteria for evaluating manual scheduling of part-time student workers and SPRF expert systems (Table 8).

Table 8. Criteria for the evaluation of optimal manual and SPRF expert system scheduling of part-time student workers in the university residence hall foodservice operation.

| OBJECTIVE | CRITERIA |
|---|---|
| I. Manual scheduling and SPRF expert system | |
| 1. To minimize scheduling time | 1.1 Use the minimum amount of supervisor time to perform term, final, and volunteer scheduling. |
| 2. To minimize the supervisor time for training student workers | 2.1 As often as possible, try to schedule student workers into jobs in which they have previously performed with success. |
| | 2.2 Minimizing number of job types per student worker. |
| | 2.3 Student workers are able to handle the jobs assigned. |
| | 2.4 Schedule experienced student workers first. |
| 3. To be able to efficiently use available student workers | 3.1 All shifts are filled (least number of open shifts at end of the scheduling process). |
| 4. To maximize the satisfaction of student workers | 4.1 Student workers are capable of doing the job assigned. |
| | 4.2 80% of student workers are satisfied with the number of shifts given and the work time and day scheduled. |
| 5. To be able to meet the requirements of scheduling part-time student workers each time | 5.1 All criteria jobs are filled by student workers. |
| | 5.2 Student workers are not given the shifts for which they have classes immediately before or after. |
| | 5.3 All lunch work shifts are filled by student workers. |

Table 8. (Continued)



| OBJECTIVE | CRITERIA |
|---|---|
| **II. SPRF Expert System** | |
| 6. To make sure the knowledge embedded in the SPRF expert system is complete, consistent, and accurate. | 6.1 Critical problem-solving strategies (e.g. previous experience and job preference were considered when scheduling) involved in scheduling are incorporated into this computer program.<br>6.2 Be able to perform all of the job elements in scheduling done by personnel supervisors: organize to schedule, schedule, check, copy, file schedule, and re-schedule (indicated in Table 5).<br>6.3 The completed schedule always meets the requirements of scheduling part-time student workers (e.g. no conflict between shifts assigned and student classes; all critical jobs are filled by student workers).<br>6.4 Knowledge (e.g. problem-solving strategies) and the structure for storing the design artifact are appropriately organized and ordered and presented at the correct level of detail.<br>6.5 The embedded process allows the system to replicate important and valuable parts of the performance of expert schedulers.<br>6.6 The completed schedule is sufficiently precise to satisfy its intended use. |
| 7. To be user friendly and easy to use | 7.1 The display window is able to facilitate the input of student data and job data.<br>7.2 Be able to perform the following functions selectively: changing student database, job database, completed schedule database; term, final, and volunteer scheduling; weekday and weekend scheduling; check, copy, and copy schedule; and re-scheduling. |

Table 8. (Continuted)



| OBJECTIVE | CRITERIA |
| --- | --- |
| 7. To be user friendly and easy to use. | 7.3 All functions provided for the user are organized in a structured way.<br>7.4 A brief description is given for each function that is provided to the user.<br>7.5 A user menu is provided and expressed in the user's own terminology.<br>7.6 Be able to detect incorrect input values immedi-cately and allow to be corrected at the same time.<br>7.7 Be able to determine the form and content of outputs (e.g. completed schedules, student data file) as needed.<br>7.8 Be able to display clear and useful error messages in the user's own terminology.<br>7.9 Its outputs are easy to read and use; no excessive information is present. |
| 8. To be able to be updated easily. | 8.1 Be able to change job type, job priority, in/out time of shifts, and other constraints easily by the user.<br>8.2 Be able to modify the output to reflect changes based on new input.<br>8.3 Be able to provide the flexibility in data structure for accommodate possible new input parameters (e.g. student past performance). |

* SPRF: Scheduling part-time student workers in the university residence hall foodservice operation.

a

e

e

a

c

(C

co

of

5.

8)

ot

ab

ev

Ta

wo

fr

sy

ha

Th

Af

to

Bas

3.2.4.3.2 Evaluation approaches. The two evaluation approaches for the SPRF expert system were: laboratory evaluation and field evaluation.

A check-list approach was developed for the laboratory environment evaluation. The researcher examined test results according to the questions indicated on the check-list. The check-list (Table 9) used in this study was based on Table 8 (Criteria 1.1, 3.1, 5.1- 5.3, 6.1, 6.4, 6.5, and 7.1-7.9).

For field testing, the suggested evaluation approach consisted of three parts: (1) evaluating the system in terms of its completeness, consistency, and accuracy (criteria 5.1-5.3; 6.1-6.6 in Table 8), efficiency (criteria 3.1 in Table 8), ease of use (satisfaction of personnel supervisors and other users; criteria 2.1-2.4; 7.1-7.9 in Table 8), and ability to update (criteria 8.1-8.3 in Table 8), (2) evaluating the system in terms of its speed (criteria 1.1 in Table 8), and (3) evaluating the satisfaction of student workers with schedules given (criteria 4.1-4.2 in Table 8).

For the first part, the method would involve responses from a group of experts (judges) who rate the SPRF expert system using a questionnaire. Personnel supervisors of MSU who have done scheduling for at least one year will act as judges. They will be instructed in the use of the SPRF expert system. After becoming familiar with this system, they will be asked to work on the same scheduling problem by using this system. Based on their expertise in manual scheduling, they will then

Table 9. A check-list for evaluation of the SPRF[a] expert system demonstration prototype in a laboratory environment[b]

| Question |
| --- |
| 1. Are problem-solving strategies that are incorporated into the SPRF expert system demonstration prototype complete, accurate and consistent ? |
| 2. Does the SPRF expert system demonstration prototype solve problems in the natural way the expert scheduler prefers ? |
| 3. Are the results of scheduling organized and presented at the right level of details that satisfy the schedule's intended use ? |
| 4. Does the SPRF expert system demonstration prototype help the user in some significant way ? |
| a. Greatly reduce the scheduling time |
| b. Efficient utilization of available student workers |
| 5. Is the interface friendly and clear for different groups of users (e.g. personnel supervisor, student supervisor) ? |

[a] SPRF: Scheduling part-time student workers in the university residence hall foodservice operation.
[b] Boehm et al.(1978); Waterman (1986).

evaluate the SPRF expert system. Finally, the responses to the questionnaire from each test site (each judge) will be gathered and analyzed statistically.

For the second part of the evaluation approach suggested for field testing, the worksheet used in collection of time data on manual scheduling of part-time student workers will be also used to collect time data on scheduling by the SPRF expert system. However, to make the worksheet specific for scheduling by the SPRF expert system, the coding at the worksheet for manual scheduling (Table 4) and coding guides (Table 5, 6, 7) should be modified. That is, each element in the scheduling function will be able to represent scheduling by the SPRF expert system rather than manual scheduling.

For the third part of the evaluation approach suggested for field testing, a questionnaire to student workers will be used as a survey instrument; students who work in the cafeteria area of residence halls that use the SPRF expert system will act as the sampling frame. Those students will be asked to fill out the questionnaire and the responses from each student will be compiled and analyzed statistically. The development of this questionnaire will be based on related criteria 4.1-4.2 in Table 8.

## Chapter IV

## RESULTS AND DISCUSSION

The purposes of this study were to (1) obtain objective time data (e.g. the amount of time used in term scheduling) on manual scheduling of part-time student workers in one university residence hall foodservice operation, (2) construct an expert system demonstration prototype for scheduling part-time student workers in one university residence hall foodservice operation to demonstrate feasibility of use of expert systems in foodservice management, (3) develop a method for evaluation of the expert system, and (4) evaluate the demonstration prototype of the expert system in a laboratory environment.

Discussion of results below was divided into three parts: (1) manual method to schedule part-time student workers, (2) collection of time data on manual scheduling of part-time student workers, and (3) the development of a SPRF expert system demonstration prototype.

Information on the manual method to schedule part-time student workers was obtained through formal meetings with three personnel supervisors (discussed in Section 3.1.1) and weekly visits with three personnel supervisors for the collection of time data on manual scheduling of part-time student workers (discussed in Section 3.1.3). These tasks were done by the researcher.

The objective of collection of time data on manual scheduling of part-time student workers was to determine the amount of time needed for scheduling student workers per term and then use the information as a basis to evaluate the SPRF expert system demonstration prototype. This collection of time data on manual scheduling of part-time student workers was done by the researcher. Data obtained from the worksheets for collection of time data were compiled and analyzed. The amount of time needed for each scheduling function (term, final, and volunteer scheduling) per term (hours), total time needed for scheduling student workers per term (hours), and mean time needed for scheduling one student worker per term (min) were calculated for each of three residence halls (Tables 10, 11, 12). Findings and implications suggested are discussed in section 4.2 below.

The development of a SPRF expert system demonstration prototype was done using a team approach. The researcher, an intermediary between the knowledge engineer and human expert, gained expertise on scheduling part-time student workers from the human expert, communicated expertise gained to the knowledge engineer, and then tested and evaluated SPRF expert system demonstration prototype. The knowledge engineer represented knowledge (expertise) in the formal way suggested by DSPL, the expert system building tool, and implemented formalized knowledge on a working computer program to make it solve problems in much the same manner as the human expert.

Table 10. Number of students and times used to manually schedule cafeteria part-time student workers at Mason-Abbot Hall, Michigan State University.

| Item \ Term | Spring 1989 | Fall 1989 | Winter 1990 | Mean of Three Terms |
|---|---|---|---|---|
| Number of Students | 75 | 88 | 75 | 79.33 ± 6.13 |
| Time Spent in Term Scheduling (hrs) | 16.65 | 78.8 | 17.5 | 37.65 ± 29.10 |
| Time Spent in Final Scheduling (hrs) | 7.42 | 13.25 | 4.91 | 8.53 ± 3.49 |
| Time Spent in Volunteer Scheduling (hrs) | 1.00 | 0.00 | 0.00 | 0.33 ± 0.47 |
| Total Time Needed Per Term (hrs) | 25.07 | 92.05 | 22.41 | 46.51 ± 32.22 |
| Mean Time Needed Per Student (min) | 20.06 | 62.76 | 17.93 | 35.12[a] |

[a] (Mean Time Needed Per Term (hrs)/Mean Number of Students Per Term) * 60 = 35.12 (min)

Table 11. Number of students and times used to manually schedule cafeteria part-time student workers at Shaw Hall, Michigan State University.

| Item \ Term | Spring 1989 | Fall 1989 | Winter 1990 | Mean of Three Terms |
|---|---|---|---|---|
| Number of Students | 120 | 135 | 120 | 125.00 ± 7.07 |
| Time Spent in Term Scheduling (hrs) | 49.06 | 161.50 | 100.00 | 103.52 ± 45.97 |
| Time Spent in Final Scheduling (hrs) | 12.67 | 34.5 | 18.00 | 21.72 ± 9.29 |
| Time Spent in Volunteer Scheduling (hrs) | 14.25 | 0.00 | 2.00 | 5.42 ± 6.30 |
| Total Time Needed Per Term (hrs) | 75.98 | 196.00 | 120.00 | 130.66 ± 49.57 |
| Mean Time Needed Per Student (min) | 37.99 | 87.11 | 60.00 | 62.72[a] |

[a] (Mean Time Needed Per Term (hrs)/Mean Number of Students Per Term) * 60 = 62.72 (min)

Table 12. Number of students and times used to manually schedule cafeteria part-time student workers at Brody Complex, Michigan State University.

| Item \ Term | Spring 1989 | Fall 1989 | Winter 1990 | Mean of Three Terms |
|---|---|---|---|---|
| Number of Students | 235 | 200 | 250 | 288.33 ± 63.55 |
| Time Spent in Term Scheduling (hrs) | 46.92 | 40.30 | 78.49 | 55.24 ± 16.66 |
| Time Spent in Final Scheduling (hrs) | 9.52 | 8.75 | 42.50 | 20.26 ± 15.73 |
| Time Spent in Volunteer Scheduling (hrs) | 13.72 | 0.00 | 0.00 | 4.57 ± 6.47 |
| Total Time Needed Per Term (hrs) | 70.16 | 49.05 | 120.99 | 80.07 ± 30.19 |
| Mean Time Needed Per Student (min) | 17.91 | 14.72 | 29.04 | 16.66[a] |

[a] (Mean Time Needed Per Term (hrs)/Mean Number of Students Per Term) * 60 = 16.66 (min)

The SPRF expert system demonstration prototype has completed weekday scheduling (vs. weekend, final, and volunteer scheduling) and been evaluated in a laboratory environment. This suggested that expert systems technology could effectively be applied to the scheduling problem, and that the system development would be beneficial for the university foodservice operation. Based on the results of testing and evaluation, suggestions for improving the SPRF expert system demonstration prototype are given in Section 4.3. In addition, criteria and approaches for evaluating the SPRF expert system demonstration prototype were also developed and were presented in Sections 3.2.4.3.1 and 3.2.4.3.2.

## 4.1 Manual Method to Schedule Part-Time Student Workers

The basic manual method, used to schedule part-time student workers by personnel supervisors at Brody Complex, Shaw Hall, and Mason-Abbot Hall, consisted of term scheduling, final scheduling, and volunteer scheduling. The term scheduling procedure below was obtained from written (DHFS, 1988) and oral information provided by the personnel supervisors at Brody Complex, Shaw Hall, and Mason-Abbot Hall, which was similar in part to the written method of MSU's Department of Housing and Food Service (DHFS, 1988).

1. Briefly look through all student applications, make mental notes of those with lunch work shifts open, check for returning and outstanding student workers, and check specific jobs that student workers want. The personnel supervisor at Brody Complex sorted all applications into lunch, breakfast, and dinner groups at the same time (Personnel supervisors at Shaw and Mason-Abbot Hall did not do this task).
2. Take all of student applications and begin filling in jobs. Student workers who turned in their applications first were scheduled first. Lunch work shifts were scheduled first, then breakfast, dinner, and weekend. Also, critical jobs were filled first.
3. Schedule student workers.
    a. Consider previous positions or experience held by student workers.
    b. Consider job preference of student workers.
    c. Consider the number of work hours desired by student workers.
    d. Consider work time and day preferred by student workers.

e. Consider class schedules of student workers.

f. Give student workers the work time and day they prefer unless they have lunch work shifts open but not mark preferred.

g. Give student workers the jobs they prefer. Otherwise, try to fill in critical jobs first. The personnel supervisor at Brody Hall filled critical jobs first and did not consider student worker preference (the personnel supervisors at Shaw and Mason-Abbot Hall did do this).

h. If the student worker only had certain time periods open that were early filled by other student workers, then supervisor went back through the completed schedules and made some changes.

4. If there were still open work shifts during the week, split any remaining work shift as a last resort so that two student workers shared one work shift.

5. The personnel supervisor filled breakfast work shifts with student workers who had no class at 10:20 am (personnel supervisors at Shaw and Mason-Abbot Hall did not do this).

6. Double check the completed schedule.

The final scheduling contained the following steps:

1. Fill in the last meal served first and work backwards (e.g. Friday Dinner, Friday Lunch, Friday Breakfast, then Thursday Dinner, etc.) because the last meal during final week is the hardest to fill.
2. If possible, schedule student workers to the jobs they have done during the term (The personnel supervisor at Brody Hall did not follow this rule). At the same time, consider student worker job preferences (The personnel supervisor at Brody Hall did not do this).
3. Give a similar amount of work shifts to each student worker.
4. Fill in each meal completely before going on to the next meal.
5. Double check the completed schedule.

The volunteer scheduling was needed for holidays (e.g. Mother's Day, Easter) and for work periods between terms when student workers would not normally be available to work. The work shifts needed were determined by personnel supervisors and posted on the bulletin board. Student workers were asked to sign up if they were available at the time. Then personnel supervisors adjusted and completed the schedule.

## 4.2 Collection of Time Data on Manual Scheduling of Part-Time Student Workers

As indicated in Table 13, the amount of supervisor time needed for scheduling student workers varied among terms (Spring Term 1989, Fall Term 1989, and Winter Term 1990) in each of three residence halls. For example, the personnel supervisor at Mason-Abbot Hall spent 25.07 hours scheduling 75 student workers during Spring Term 1989, 92.05 hours scheduling 88 student workers during Fall Term 1989, and 22.41 hours scheduling 75 student workers during Winter Term 1990. Also at Mason-Abbot Hall, mean times needed for scheduling one student worker Spring Term 1989, Fall Term 1989, and Winter Term 1990 ranged from 17.93 to 62.76 minutes whereas times required by supervisors at Shaw Hall ranged from 37.99 to 87.11 minutes, and at Brody complex from 14.72 to 29.04 minutes.

The amount of supervisor time needed for scheduling student workers did not appear to be related to the number of student workers scheduled. Time required for scheduling did not increase directly with number of student workers scheduled. For example, schedulers at Mason-Abbot Hall, with an average of 79.33 $\pm$ 6.13 student workers (N=3; Spring 1989, Fall 1989, and Winter 1990), spent an average of 46.51 $\pm$ 32.22 hours in scheduling student workers per term; Shaw Hall with an average of 125 $\pm$ 7.07 student workers spent an average of 130.66 $\pm$ 49.57 hours; and Brody Complex with an average of

Table 13. Number of students and times used to manually schedule cafeteria student workers at Mason-Abbot Hall, Shaw Hall, and Brody Complex, Michigan State University.

| Term | Factor | Hall | | |
|---|---|---|---|---|
| | | Mason-Abbot | Shaw | Brody Complex |
| Spring 1989 | Number of Student Workers | 75 | 120 | 235 |
| | Total Time Spent in Scheduling (hrs) | 25.07 | 75.98 | 70.16 |
| | Mean Time Needed for Scheduling Per Student Worker (min) | 20.06 | 37.99 | 17.91 |
| Fall 1989 | Number of Student workers | 88 | 135 | 200 |
| | Total Time Spent in Scheduling (hrs) | 92.05 | 196.00 | 49.05 |
| | Mean time Needed for Scheduling Per Student Worker (min) | 62.76 | 87.11 | 14.72 |
| Winter 1990 | Number of Student Workers | 75 | 120 | 250 |
| | Total time Spent in Scheduling (hrs) | 22.41 | 120.00 | 120.99 |
| | Mean Time Needed for Scheduling Per Student Worker (min) | 17.93 | 60.00 | 29.04 |
| Mean of Three Terms | Number of Student Workers | 79.33 ± 6.13 | 125.00 ± 7.07 | 288.33 ± 63.55 |
| | Total time Spent in Scheduling (hrs) | 46.51 ± 32.22 | 130.66 ± 49.57 | 80.07 ± 30.19 |
| | Mean Time Needed for Scheduling Per Student Worker (min) | 35.12 | 62.72 | 16.66 |

$288.33 \pm 63.55$ student workers spent an average of $80.07 \pm 30.19$ hours.

The results were discussed with personnel supervisors (schedulers) in three residence halls to determine possible factors accounting for the variability of results. The factors are listed immediately below and are then discussed in Sections 4.2.1 to 4.2.6.

When the personnel supervisors were in training for their job, those personnel supervisors had only been given brief, basic oral instruction on scheduling (e.g. when do the term and final scheduling need to be started; the requirements of scheduling, such as scheduling critical job first) and then required to practice scheduling on a given sample; no written material that showed how to schedule student workers step by step had been provided to them. These three personnel supervisors learned scheduling on the job. Therefore, scheduling methods used by each of the three personnel supervisor were not expected to be the same. With different methods used, the time spent in scheduling could be expected to be different.

In addition, the quantity of returning vs. new student workers varied. Also, human error effecting the precision of time data collection on manual scheduling of part-time student workers was also a possible factor accounting for the variability of results among halls.

4.2.1 Scheduling Methods Differ Among Halls

The manual method of scheduling used at Brody Complex was not entirely suitable for Shaw or Mason-Abbot Hall because of the smaller quantity of student workers. There were only a limited number of student workers at Shaw and Mason-Abbot Hall; the average number of student workers per term was 125 and 79.33 respectively vs. 228 at Brody Complex (Table 13). With such a limited number of student workers, personnel supervisors at Shaw and Mason-Abbot Hall briefly looked through all the applications, made mental notes of those student workers with lunch open, and checked for returning and outstanding student workers, instead of physically breaking all student job applications into lunch, breakfast, and dinner piles. Also, personnel supervisors at Shaw and Mason-Abbot Hall were better able to be more concerned with the needs of student workers (e.g. job preference, work time and day preferences) when scheduling a smaller number of student workers.

Thus, not only the number of student workers employed, but also the scheduling method used by personnel supervisors in different residence halls at MSU accounted for the variation in the amount of time needed for scheduling student workers.

4.2.2 Scheduling Methods Differ Among Schedulers in One Hall

Student supervisors, assigned by the personnel supervisor to do some scheduling, may not have had adequate training and practice to perform the scheduling task using the preferred method with "normal" amount of time. Thus, scheduler was also a factor effecting the amount of time needed. For example, the student supervisor at Brody Complex, Winter Term 1990, considered the job preference of student workers when performing final scheduling. Thus, rather than looking at the individual final exam student employment schedule sheet (Figure 4), he also referred to the individual application for student foodservice employment sheet (Figure 5) which indicated the job preference of student workers. In this way, more time was needed for assigning to student workers the jobs which they preferred.

4.2.3 The Quantity of Returning vs. New Student Workers

The quantity of returning student workers was a factor which, in part, accounted for the time needed for scheduling at Brody Complex, Show, and Mason-Abbot Hall. For example, in Spring Term 1989, Mason-Abbot Hall had a greater number of returning student workers; nine out of 10 student workers were returning student workers in Spring Term 1989, while three out of 10 student workers were returning student workers in Fall Term 1989. As indicated in Table 10, the mean time needed to schedule one student worker was 20.06 minutes in Spring Term

Figure 4. Final exam student employment schedule sheet

MICHIGAN STATE UNIVERSITY

DIVISION OF HOUSING AND FOOD SERVICES
FINAL EXAM STUDENT EMPLOYMENT SCHEDULE

Michigan State University is an Affirmative Action/Equal Opportunity Institution

NAME________________________ STUDENT NO.__________ DATE__________
(Last) (First) (M)
CAMPUS ADDRESS________________________ PHONE__________

| | M | T | W | Th | F |
|---|---|---|---|---|---|
| 7:45 - 9:45 | | | | | |
| 10 - 12 | | | | | |
| 12:45 - 2:45 | | | | | |
| 3 - 5 | | | | | |
| 5:45 - 7:45 | | | | | |
| 8 - 10 | | | | | |

← EXAM SCHEDULE

Place an X in the appropriate spaces for your exams.

All student employees are rescheduled for exam week, from Friday dinner before exams through Friday of exams. As a condition of employment you are required to work up to six meals during this period.

Your work assignment may be changed at any time by an authorized supervisor and you may be required to work in any area.

WORK PREFERENCE

Indicate work preference with a small "x" at the appropriate meal.

| SHIFT | FRIDAY | SATURDAY | SUNDAY | MONDAY | TUESDAY | WEDNESDAY | THURSDAY | FRIDAY |
|---|---|---|---|---|---|---|---|---|
| BREAKFAST | | | | | | | | |
| LUNCH | | | | | | | | |
| DINNER | | | | | | | | |

At the beginning of next term, work will be on a voluntary sign-up basis, first come/first serve for job preferences. We reserve the right to change your work assignment. The sign-up will be posted during finals week.

Please help us out and sign up for shifts. Also, please fill out a schedule form for next term and hand it in before leaving campus for the break. We do not guarantee any job term to term.

Day and time leaving campus________________ Day arriving next term________________

BELOW THIS LINE IS FOR OFFICE USE ONLY

---

Final Exam Work Schedule

NAME________________________ ADDRESS________________________

Final exam week begins Friday__________ at dinner. Please be prompt when punching in for work.

| SHIFT | FRIDAY | SATURDAY | SUNDAY | MONDAY | TUESDAY | WEDNESDAY | THURSDAY | FRIDAY |
|---|---|---|---|---|---|---|---|---|
| BREAKFAST | | | | | | | | |
| LUNCH | | | | | | | | |
| DINNER | | | | | | | | |

Figure 5. Application for student foodservice employment sheet.

MICHIGAN STATE UNIVERSITY
Division of Housing and Food Services
**APPLICATION FOR STUDENT FOOD SERVICE EMPLOYMENT**

*Michigan State University is an Affirmative Action/Equal Opportunity Institution*

TERM
F W Sp Su

| OFFICE USE ONLY | |
|---|---|
| | Dept. ______ |
| | Prod. Cntr. ______ |
| | Wage Rate ______ |
| PB Mr TC W4 | Date Hired ______ |

Full Name ______ (Last, First, MI) Student No. ______ Major ______ Class (Year) ______

Campus address ______ Soc. Sec. # ______ Telephone ______

Home address ______ Date of birth ___/___/___

If you have worked at MSU before, where ______ when? ______ Positions held ______

Previous work experience, special skills, job preference ______

IMPORTANT: Have you been awarded any of these financial aids? National Direct Student Loans ______ Equal Opportunity Grant ______ Work-Study Program ______ Other ______

What physical conditions (e.g., limitations, allergies, etc.) do you have which may be aggravated by employment? ______

Number of work hours wanted each week ______ Check which weekend you want to work A ______ B ______ Both ______
(A and B designate alternate weekends)

A Mark all classes and activities with an X

| | 8-8:50 | 9:10-10 | 10:20-11:10 | 11:30-12:20 | 12:40-1:30 | 1:50-2:40 | 3-3:50 | 4:10-5 | 5:20-6:10 | 7-9 |
|---|---|---|---|---|---|---|---|---|---|---|
| M | | | | | | | | | | |
| T | | | | | | | | | | |
| W | | | | | | | | | | |
| Th | | | | | | | | | | |
| F | | | | | | | | | | |

B Indicate work preferences with small Xs at appropriate meals in box below. DO NOT WRITE ANYTHING ELSE.

WORK SCHEDULE Weekend ______

| SHIFT | M | T | W | Th | F | St | Sn |
|---|---|---|---|---|---|---|---|
| Breakfast | | | | | | | |
| Lunch | | | | | | | |
| Dinner | | | | | | | |

C

Name ______ Campus Address ______ Effective ______ Weekend ______

| Shift | M | T | W | Th | F | St | Sn |
|---|---|---|---|---|---|---|---|
| Breakfast | | | | | | | |
| Lunch | | | | | | | |
| Dinner | | | | | | | |

1989, while 62.76 minutes were required in Fall Term 1989.

Because those returning students were familiar with jobs and shifts possible for them to work, they wrote down the shifts they were able to work on individual application sheets in Section C (Figure 5). Normally only supervisors filled out this section. However, returning student workers, who knew their supervisor well, completed this task for him whereas new student workers did not complete Section C (Figure 5). Thus, when looking at their applications, the personnel supervisor at Mason-Abbot Hall gave returning student workers the shifts they requested if those shifts were on the list of top job priorities. Since the personnel supervisor at Mason-Abbot Hall did not need to go through student worker class schedules, job preference, number of work hours desired, etc. as they had to do for new student workers, much supervisor time was saved.

On the other hand, in Fall Term 1989, Mason-Abbot Hall employed many new student workers (75 percent of 88 student workers were new student workers). As they turned in application sheets, the personnel supervisor at Mason-Abbot Hall scheduled them right away. Since they turned in application sheets with only Sections A and B (Figure 5) completed individually, the personnel supervisor at Mason-Abbot Hall was required to perform scheduling in Section C (Figure 5) and, at the same time, communicate with new student workers individually. Therefore, more supervisor time was spent in this way.

### 4.2.4 Human Error on Collection of Time Data

In this study, schedulers were asked to record the amount of time they spent in scheduling on the worksheet (Table 4). However, schedulers did not always remember to record the time spent in scheduling immediately after each scheduling session. When they tried to go back at a later time and recall how much time they had spent, schedulers could only estimate the time. Also, schedulers forgot to consider the amount of time taken by interruptions (e.g. answering the phone, asked to meet some people or help something) during scheduling.

Because the researcher was unable to be with schedulers 100% of the time, scheduler accuracy could not be evaluated. Based on weekly interaction with schedulers (to collect the worksheet) in three residence halls, the researcher estimated that such a human error (e.g. not always remember to record the time spent in scheduling immediately after each scheduling session) occurred 40% of the time.

### 4.2.5 Other Factors

Personnel supervisors in the three residence halls usually delegated part of scheduling task to student supervisors. Time needed for communicating with student supervisors, frequency of delegation, and the productivity of student supervisors also accounted for differences in the amount of time needed for scheduling student workers per term.

### 4.2.6 Summary

Table 14 summarizes the possible reasons accounting for differences in the amount of time needed for scheduling student workers per term at three residence halls.

## 4.3 The Development of a SPRF Expert System Demonstration Prototype

The development of a SPRF expert system demonstration prototype was done using a team approach. The researcher, an intermediary between the knowledge engineer and human expert, gained expertise on scheduling part-time student workers from the human expert, communicated expertise gained to the knowledge engineer, and then tested and evaluated the demonstration prototype. The knowledge engineer represented knowledge (expertise) in the formal way suggested by DSPL (the expert system building tool) and implemented formalized knowledge on a working computer program to make it solve problems in much the same manner as the human expert.

The SPRF expert system demonstration prototype has completed weekday scheduling (vs. weekend, final, and volunteer scheduling) and been evaluated in a laboratory environment. This suggested that expert systems technology could effectively be applied to the scheduling problem, and that the system development would be beneficial for the university foodservice operation. The results of evaluation are discussed in Section 4.3.2.

Table 14. Factors accounting for the variability of results generated from the collection of time data at three residence halls during three terms.



| Hall / Term | Mason-Abbot Hall | Shaw Hall | Brody Complex |
|---|---|---|---|
| Spring Term 1989 | 1. **Returning student workers**<br><br>90% of 75 student workers were returning student workers who were able to self-schedule and saved time of schedulers.<br><br>2. **No delegation**<br><br>The personnel supervisor performed scheduling himself; no time was needed for communicating with student supervisors and other delegates. | 1. **Returning student workers**<br><br>(60% of 120 student workers were returning student workers).<br><br>2. **Much time spent in volunteer scheduling**<br>(19% of total time spent in scheduling attributed to volunteer scheduling, while 0% in Fall 1989 and 1.6% in Winter 1990). | 1. **Returning student workers**<br><br>(80% of 235 student workers were returning student workers).<br><br>2. **Much time spent in volunteer scheduling**<br>(20% of total time spent in scheduling attributed to volunteer scheduling, while 0% in Fall 1989 and 0% in Winter 1990). |

Table 14. (Continuted)



| Hall / Term | Mason-Abbot Hall | Shaw Hall | Brody Complex |
|---|---|---|---|
| Fall Term 1989 | 1. **Short of student workers**<br><br>Increased efforts were made to look for new student workers (e.g. post sign).<br><br>2. **Returning student workers**<br><br>25% of 88 student workers were returning student workers.<br><br>3. **Much time spent in final scheduling**<br>30% of total time spent in scheduling attributed to final scheduling, while 14% in Fall 1989 and 22% in Winter 1990.<br><br>4. **Human error for recording time**<br>Rather than recording the time spent in re-scheduling each student, the student supeervisor recorded the time he needed to stay in office and deal with problems of student workers. | 1. **Short of student workers**<br><br>2. **Returning student workers**<br><br>(20% of 135 student workers were returning student workers).<br><br>3. **New scheduler**<br><br>Scheduling was taken over by two student supervisors. | 1. **Short of student workers**<br><br>2. **Returning student workers**<br><br>(25% of 200 student workers were returning student workers).<br><br>3. **New scheduler**<br><br>Scheduling was taken over by a new full time supervisor and three student supervisors.<br><br>4. **Human error for recording time**<br>Schedulers forgot to record the time spent in final scheduling; it occurred an estimated 60% of the time. |

Table 14. (Continued)



| Term \ Hall | Mason-Abbot Hall | Shaw Hall | Brody Complex |
|---|---|---|---|
| Winter Term 1990 | 1. **New scheduler**<br><br>The final scheduling was done by a new personnel supervisor.<br><br>2. **Returning student workers**<br><br>80% of 75 student workers were returning student workers. | 1. **New scheduler**<br><br>The scheduling was taken over by a new personnel supervisor.<br><br>2. **Human error in recording time**<br>Schedulers forgot to record the time spent in scheduling; it occurred an estimated 80% of the time. | 1. **Different scheduling method**<br><br>The method of final scheduling used by the student supervisor was different from that used in Spring 1989 and Fall 1989 because he did consider student job preference).<br><br>2. **Much time spent in final scheduling**<br>35% of total time spent in final scheduling, while 14% in Spring 1989 and 18% in Fall 1989.<br><br>3. **Returning student workers**<br><br>90% of 250 student workers were returning student workers. |

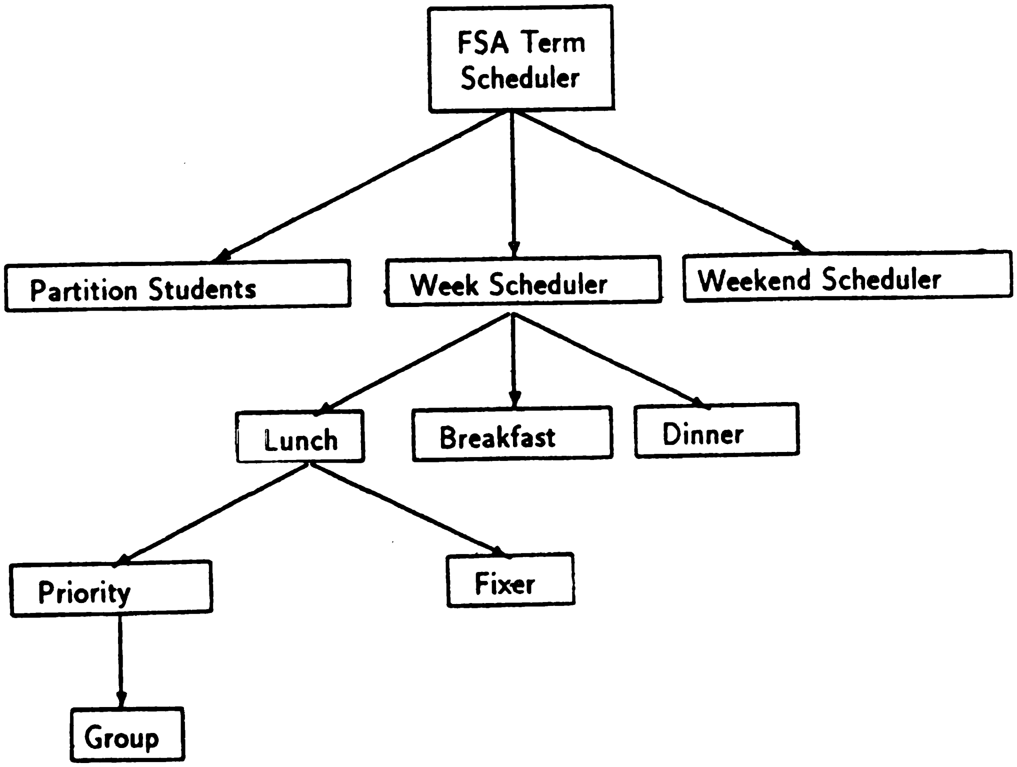
4.3.1 Status of the SPRF Expert System Demonstration Prototype

The interface and scheduling strategies implemented in the SPRF expert system demonstration prototype were developed by the knowledge engineer. Appendix C, written by the knowledge engineer, is a detailed description of the interface and scheduling method used by the SPRF expert system demonstration prototype.

In summary, Figure 6 displays the basic scheduling structure implemented in the SPRF expert system demonstration prototype. This scheduling structure consisted of the top specialist and its sub-specialists. The **Term Scheduler** was responsible for all of the scheduling. The plan it used called the **Partition Students** specialist, then the **Week Scheduler** specialist, and finally the **Weekend Scheduler** specialist. The **Partition Students** specialist loaded in student workers from the file and placed them into lunch, breakfast, and dinner groups by examining their class schedules to determine into which meal during the week student workers could be scheduled.

The **Week Scheduler** specialist was to schedule lunch, then breakfast, and finally dinner. The plan used by the **Lunch** sub-specialist was to call the **Priority** specialist until all of student workers in lunch group had been scheduled. The **Priority** specialist's plan was to set up a group of jobs and student workers needed to fill the jobs, and then to call the sub-specialist **Group**. The sub-specialist **Group** was responsible for scheduling a group student workers into a group of jobs.

**Figure 6. The scheduling structure implemented in the SPRF expert system demonstration prototype (Appendix C).**



4.3.2 Testing and Evaluation of the SPRF Expert System Demonstration Prototype in a Laboratory Environment

Laboratory testing of the SPRF expert system demonstration prototype consisted of scheduling the 213 actual student workers of Brody Complex, Winter Term 1990. This test was evaluated using a predetermined check-list (Table 9 in Section 3.2.4.3.2). After the answer to each question, suggestions were given for further improvements of the SPRF expert system demonstration prototype. Results of the evaluation were summarized in Table 15.

For the first question in the check-list "**Are problem-solving strategies that are incorporated into the SPRF expert system demonstration prototype complete, accurate, and consistent ?**", the answer was no. The problem-solving strategies incorporated in the SPRF expert system demonstration prototype were not complete for two reasons:

1. Rather than just considered student worker class schedules as the SPRF expert system demonstration prototype did, the personnel supervisor also considered student worker job preference, work time preference, and work day preference.
2. The personnel supervisor tried to give each student worker the number of shifts he/she desired rather than to schedule each student workers into three shifts per week as the SPRF expert system demonstration prototype did.

Table 15. The results of evaluation of the SPRF[a] expert system demonstration prototype in a laboratory environment[b]

| Question | Evaluation |
|---|---|
| 1. Are problem-solving strategies that are incorporated into the SPRF expert system demonstration prototype complete, accurate, and consistent ? | No |
| 2. Does the SPRF expert system demonstration prototype solve problems in the natural way the expert scheduler prefers ? | No |
| 3. Are the results of scheduling organized and presented at the right level of details that satisfy the schedule's intended use ? | Yes |
| 4. Does the SPRF expert system demonstration prototype help the user in some significant way ? | |
| a. Greatly reduce the scheduling time | * |
| b. Efficient utilization of available student workers | * |
| 5. Is the interface friendly and clear for different groups of users (e.g. personnel supervisor, student supervisor) ? | * |

[a] SPRF = Scheduling part-time student workers in the university residence hall foodservice operation.

[b] Boehm et al. (1978); Waterman (1986).

* There is no answer available yet.

To satisfy student workers but at the same time make sure their needs were fulfilled, personnel supervisors usually gave student workers the work time and day they preferred unless they had lunch work shifts open but not marked preferred.

The SPRF expert system demonstration prototype did not consider student workers' preferences of job, time and day at all because these considerations could make the SPRF expert system demonstration prototype too large. Thus, it could be possible that student workers would ask to be rescheduled if they were not happy with the work time and day assigned. With this demonstration prototype, human judgement (from human scheduler) would still be part of the final solution (rescheduling student workers to meet their needs).

deHass (1983) pointed out that computer programs to handle a scheduling problem on a large scale were relatively simple, but could be expensive to use due to the large number of human judgments going into the solution. The SPRF expert system demonstration prototype could result in excessive costs. However, considering student workers' preferences of job, time and day could be done by further programming efforts in the stage of research prototype (Figure 1).

The SPRF expert system demonstration prototype attempted to schedule each student worker into 3 shifts for weekday scheduling and 2 shifts for weekend scheduling instead of considering the number of work hours desired by student workers. The quantity, three and two shifts, were the average number of work shifts assigned to each student worker for weekday and weekend scheduling respectively each term. However, some student workers could not work more work shifts than they had indicated on their applications because they may have wanted to spend more time in studying and/or other employment activities; in the meantime, other student workers who wanted to obtain more work shifts to earn extra money may not get the extra work shifts. The method used by the SPRF expert system demonstration prototype may have been able to speed the scheduling process due to less complexity in decision-making process, but would require extra time of personnel supervisors to reschedule student workers according to their specific desires.

For the second question in the check-list "**Does the SPRF expert system demonstration prototype solve problems in the natural way the expert scheduler prefers ?**", the answer was no. The embedded reasoning process of scheduling did not seem to work in the natural way preferred by the expert scheduler. For example, lunches had traditionally been the hardest to staff; therefore, lunch work shifts were scheduled first. However, in a laboratory test, 20 lunch work shifts, pertaining to the first six job priorities, were not filled after the scheduling was completed by the SPRF expert system demonstration prototype (Table 16). On the other hand, only eight lunch work shifts were not filled after the same scheduling (Winter Term 1990) was completed by the personnel

Table 16. The contrast between manual scheduling of part-time student workers (MSPS) and scheduling by the SPRF[a] expert system demonstration prototype on the number of open work shifts of the completed weekday lunches schedule at Brody Complex, Michigan State University, Winter Term 1990.

| Priority | Job Name | Total Possible Shifts | The Number of Open Work Shifts by MSPS | The number of Open Work Shifts by SPRF | Difference of MSPS vs. SPRF |
|---|---|---|---|---|---|
| 1 | nFLUT | 5 | 0 | 4 | -4 |
| | nCON1 | 5 | 1 | 0 | 1 |
| | eFLUT | 5 | 0 | 0 | 0 |
| | eCON1 | 5 | 0 | 0 | 0 |
| | sTC1 | 5 | 0 | 0 | 0 |
| | dDR1 | 5 | 0 | 0 | 0 |
| 2 | nBD1 | 5 | 0 | 0 | 0 |
| | eBD1 | 5 | 0 | 0 | 0 |
| | sUT1 | 5 | 1 | 0 | 1 |
| | sTC2 | 5 | 0 | 0 | 0 |
| | dP&P1 | 5 | 0 | 4 | -4 |
| 3 | nCON2 | 5 | 0 | 0 | 0 |
| | eCON2 | 5 | 0 | 0 | 0 |
| | dDR2 | 5 | 0 | 0 | 0 |
| | dDR3 | 5 | 1 | 0 | 1 |
| 4 | nBD2 | 5 | 0 | 0 | 0 |
| | eBD2 | 5 | 0 | 0 | 0 |
| | sL-Stock | 5 | 3 | 3 | 0 |
| | dP&P2 | 5 | 2 | 4 | -2 |
| 5 | nBD3 | 5 | 0 | 0 | 0 |
| | eBD3 | 5 | 0 | 0 | 0 |
| | sUT2 | 5 | 0 | 0 | 0 |
| | dDR4 | 5 | 0 | 2 | -2 |
| 6 | nDESS | 5 | 0 | 3 | -3 |
| | dDR5 | 5 | 0 | 0 | 0 |
| Total | | 125 | 8 (6.4%) | 20 (16%) | -12 |

[a] SPRF = Scheduling part-time student workers in the university residence hall foodservice operation.

supervisor at Brody Complex. Therefore, to make sure that critical jobs in lunch scheduling were filled by student workers, improvements should be implemented in the SPRF expert system demonstration prototype.

For the third question in the check-list **"Are the results of scheduling organized and presented at the right level of details that satisfy the schedule's intended use ?"**, the answer was **yes**. The master schedule, indicated in Appendix D, generated by the SPRF expert system demonstration prototype, was similar to that generated by the expert scheduler manually. The only difference was that the student number was shown in the master schedule generated by the SPRF expert system demonstration prototype, instead of the student worker name. However, it would be very easy for the knowledge engineer to change student worker number into student worker name. As to the list of all student workers (indicated in Appendix E), personal information such as address and phone number, job preference, work time and day preferences, and the number of work hour desired need to be included to keep a complete student worker data file for the use of personnel management (e.g. rescheduling).

Additionally, the flexibility for determining the format and content of outputs would be beneficial, which could be an improvement in the SPRF expert system demonstration prototype.

For the fourth question in the check-list **"Does the SPRF expert system demonstration prototype help the user in some**

**significant way ?"**, there was no answer available yet. The SPRF expert system demonstration prototype seemed to be faster than manual scheduling when performing the element of term scheduling function, **Schedule**, under the same condition (scheduling 213 student workers at Brody Complex, Winter Term 1990). **Schedule** was defined as the process of matching student workers with available work shifts by considering their past experience and performance, alternative weekend, job, work time, and work day preferences; number of work hours desired; in/out time of available work shifts; and class schedule (Table 5).

The total time that the personnel supervisor at Brody Complex spent in performing the element **"Schedule"** of term scheduling during Winter Term 1990 was 44.33 hours. The personnel supervisor was unable to exactly separate the time spent in weekday scheduling from that spent in weekend scheduling; he estimated that one-eighth of the total time spent in term scheduling was used to perform weekend scheduling. In other words, approximate 38.79 hours were spent in weekday scheduling. Compared to 2.18 hours that the SPRF expert system demonstration prototype spent for performing the element **"Schedule"** of weekday scheduling under the same condition (scheduling 213 student workers at Brody Complex, Winter Term 1990), the SPRF expert system demonstration prototype seemed to be faster than manual scheduling.

With the same number of shifts and student workers, the percent of weekly shifts filled by the SPRF expert system demonstration prototype was 86.86%, while that by the manual labor scheduling was 92.21% (Table 17). At this point, manual scheduling was more efficient; however, the efficiency of the SPRF expert system demonstration prototype could be increased by further programming improvements.

For the fifth question in the check-list **"Is the interface friendly and clear for different groups of users (e.g. personnel supervisors, student supervisors) ?"**, there was no answer available yet. The SPRF expert system demonstration prototype did not yet have a friendly and clear interface. The criteria (7.1-7.9) indicated in Table 8 would be the goal to achieve in the near future.

Table 17. The contrast between manual scheduling of part-time student workers (MSPS) and scheduling by the SPRF[a] expert system demonstration prototype on the number of open work shifts of the completed weekday schedule at Brody Complex, Michigan State University, Winter Term 1990.

| Item \ Meal | Lunch | Breakfast | Dinner | Total Per Day |
|---|---|---|---|---|
| Total Number of Weekday Work Shifts | 195 | 100 | 216 | 411 |
| The Number of Open Work Shifts Remaining by MSPS | 22 | 10 | 0 | 32 |
| Percent of Work Shifts Filled by MSPS | 88.72% | 90.00% | 100.00% | 92.21% |
| The Number of Open Work Shifts Remaining After Scheduling by SPRF Expert System | 54 | 0 | 0 | 54 |
| Percent of Work Shifts Filled by the SPRF Expert System | 72.31% | 100.00% | 100.00% | 86.86% |

[a] SPRF: Scheduling part-time student workers in the university residence hall foodservice operation.

## Chapter V

## CONCLUSIONS

### 5.1 Implications of This Study for the Foodservice Industry

AI research papers described only scantily the methods employed in the development of expert systems, which were not specifically related to foodservice applications. Therefore, the method used in this study to construct the SPRF expert system may provide the foodservice industry with a model of how expert systems could be developed and used.

The process of building an expert system usually involved interaction between the expert-system builder, knowledge engineer, and one or more human experts in some problem area. However, an intermediary between the knowledge engineer and human experts (personnel supervisors) was needed by the present study. The human expert (personnel supervisor) was not be able to work with the knowledge engineer for the entire time needed for the expert system development for two reasons. First, the personnel supervisor was a entry-level position; people in this position were often promoted to higher positions after they had been in the position for approximate 6 months. Second, the turnover rate was high in the university foodservice even at the supervisor level. Therefore, an intermediary was needed for this study; it may be necessary for the expert system development in the foodservice industry.

## 5.2 Difficulties with the SPRF Expert System Demonstration Prototype Development

Discussion of difficulties occurred in the development of SPRF expert system could be divided into two parts: (1) inherent limitations of expert systems and (2) limitations of the expert system building tool "DSPL". Each part is discussed further below (Sections 5.2.1 and 5.2.2).

### 5.2.1 Inherent Limitations of Expert Systems

Even though expert systems have better ability than conventional computer programs to mimic the human reasoning process of problem solving (discussed in Section 3.2.2), they still could not behave totally as humans experts for the foreseeable future (Vedder, 1987). In this way, a challenge faced in the development of the SPRF expert system is balancing time efficiency and performance quality to produce the most optimal results.

Additionally, the power and utility of the resulting SPRF expert system depends on the quality of the underlying representation of human scheduler knowledge. The human knowledge may not all be mapped into the system; therefore, eliciting the right amount of knowledge form the human scheduler and representing that knowledge at the right level of details, which allows the developed expert system to replicate important and valuable parts of the performance of expert schedulers, becomes another challenge.

5.2.2 Limitations of the Expert System Building Tool "DSPL"

DSPL could not recognize incorrect or inconsistent knowledge. Therefore it is likely to produce incorrect results or advice when inconsistencies and/or errors are incorporated into the knowledge base (e.g. job priority).

DSPL was a programming language for designing expert systems that performed problem solving on routine design tasks (see Section 2.5). However, the SPRF problem domain was not a typical routine design. Some tasks in SPRF problem were of a classificatory nature. For example, the personnel supervisor at Brody Complex sorted student workers into lunch, breakfast, and dinner groups. Some tasks in the SPRF problem were of rule-based nature (if.., then..), such as scheduling n student workers into m jobs (e.g. if the student worker could not fill in any lunch shifts, then place this student worker on the underscheduled list).

As to the problem of rule-based nature, the domain knowledge was usually represented as sets of rules that were checked against a collection of facts about the current situation. For example, a rule in the SPRF domain may be that if the student worker have previous experience on one critical job, then the student worker is a candidate for this job. Each time, when the computer reads in one student worker, this rule would be checked against the previous experience (the fact) of the student worker. When the IF portion of a rule was satisfied by the fact (student previous experience), the action specified by the THEN portion was performed. That is, this student worker would be moved into the candidate list for the job.

Since DSPL did not support classification and rule-based methods for representing knowledge and controlling the process of problem solving, some tasks performed by the human scheduler could not efficiently be modeled using DSPL. In the present study, the programmer developed a **LISP** function to classify student workers into lunch, breakfast, and dinner group and to do the basic scheduling of n students into m jobs. The **LISP** function returned the problem back to **DSPL** when the scheduling process (scheduling n students into m jobs) was done. Thus, **DSPL** was able to function more effectively.

## 5.3 The Potential Benefits of Using the SPRF Expert System

The completion of weekday scheduling has suggested that system development would be possible. Based on the testing of weekday scheduling in a laboratory environment and data obtained from collection of time data on manual scheduling of part-time student workers (discussed in Section 4.2), the following potential benefits would be expected in the future by using a well-developed version of the SPRF expert system to facilitate scheduling.

### 5.3.1 Ensuring Consistent Performance

A human scheduler at different times or schedulers in different places may reach different conclusions about a particular scheduling situation (discussed in Section 4.2). An expert system always provides the same conclusion to the same problem. This consistency is an obvious advantage in using the SPRF expert system in matters of fairness among employees. In addition, the SPRF expert system would be able to eliminate the occasional inadvertent omissions or errors that schedulers make (e.g. forgetting to write the shift planned to give the student worker on his/her individual schedule sheet).

### 5.3.2 Reducing Supervisor Time for Scheduling

The SPRF expert system has a potential to greatly reduce the time needed to schedule student workers per term and relieve supervisors from handling problems which, for the experts, are routine. Consequently, the Department of Housing and Food Service at MSU would be better able to use its supervisors' time and abilities on the really tough problems, such as personnel management, determining the amount of work force for next term, and adjusting the number of shifts. This switch in emphasis would also help the human experts by making their jobs more interesting and rewarding.

5.3.3 Reducing Labor Costs

The personnel supervisor at Brody Complex delegated most of the scheduling tasks to student supervisors in Fall Term 1989 and Winter Term 1990. The hourly salary for a student supervisor was approximately $5.00 per hour. As discussed in Section 4.3.2, the amount of time needed to perform the element "schedule" of weekday scheduling manually was 38.79 hours, while the SPRF expert system took only 2.18 hours. In other words, by using the SPRF expert system demonstration prototype, 36.61 hours of labor at a cost $5.00/hour (hiring student supervisor to do scheduling), $183.05, could be saved.

Overall, savings in labor costs are hard to determine at this time. A formal cost benefit analysis, including the cost of this system and of hiring employees to perform scheduling, should be completed.

5.3.4 Preserving and Expanding Expertise

The SPRF expert system can help preserve existing knowledge. Also, expertise may be extended to all residence halls at MSU, even foodservice operations at other institutions employing part-time workers, through use of portable floppy disks.

# Chapter VI

# RECOMMENDATIONS FOR FURTHER RESEARCH

The SPRF expert system demonstration prototype has completed weekday scheduling (vs. weekend, final, and volunteer scheduling) in a laboratory environment. This suggested that expert systems technology could effectively be applied to the scheduling part-time student workers problem, and that the system development would be beneficial for the university foodservice operation. Therefore, the current SPRF expert system has passed the stage of demonstration prototype and has entered the stage of research prototype, and then will enter the stages of field prototype, production model and commercial system, as shown in Figure 1.

## 6.1 Research Prototype

Further developments in the research prototype stage should include the following:

1. To improve the current SPRF expert system in terms of efficient use of available student workers, scheduling time, completeness of embedded problem-solving strategies, and accuracy of the reasoning process of scheduling (discussed in 4.3).
2. To extend the current SPRF expert system to include the final scheduling and volunteer scheduling functions.

3. To develop the selective use of the following elements of scheduling functions:
   a. Rescheduling of a single (or few) student worker.
   b. Check schedule
   c. Copy schedule
   d. File schedule
   e. Change and update student database, job database, and completed schedule database.
4. To establish the user interface. The criteria 7.1-7.9 in Table 8 (Section 3.2.4.3.1) can act as a guide for the development of user interface.
5. To test and refine the research prototype by (1) bringing in additional experts to help validate the system's accuracy, and (2) using test cases not encountered by the system during its previous development (Waterman, 1986).

In addition, one of objectives of all expert systems was to help less expert professionals (e.g. trainees) improve the quality of their judgments and decisions and, possibly, their general problem-solving skills (Boritz and Brown, 1986). This objective was not included in this study. However, faced with turn over of labor schedulers (e.g. three personnel supervisors had taken over the scheduling between Spring Term 1989 and Winter Term 1990 at Shaw Hall, two in Mason-Abbot

Hall, and three in Brody Complex), it would be beneficial to have the SPRF expert system also acting as a trainer.

To be able to play a role in training new schedulers, the SPRF expert system needs to be equipped with the capabilities of explaining how the scheduling is done, which would allow users to explore each facet of their reasoning. Thus, a new scheduler could be allowed to track, step-by-step, the analysis leading to the solution or recommended decision. Also, a new scheduler could use this system to support or contradict their solution to a problem. In this way, the SPRF expert system also holds promise to reduce training costs and ensure consistency in the presentation of problem solving skills.

As the SPRF expert system performs with adequate reliability in the laboratory environment, it could pass from the research prototype to the stage of field prototype.

## 6.2 Field Prototype

In the field prototype stage, the SPRF expert system research prototype could be revised and refined based on extensive testing in the user environment (cafeteria foodservice at a residence hall of MSU). The further research at this stage should include:

1. To design and conduct the experiment of collection of time data on manual labor scheduling by taking the variables discussed in Section 4.2 into account.

2. To design and conduct the experiment of collection of time data on scheduling by the SPRF expert system.
3. To evaluate the SPRF expert system on the aspect of time-saving according to the time data obtained from 1 and 2.
4. To evaluate the SPRF expert system according to the criteria shown in Table 8. The approach to evaluate the SPRF expert system has been proposed in Section 3.2.4.3.2.
5. Compare the SPRF expert system scheduling to manual scheduling in terms of labor cost and investment.

## 6.3 Production Model and Commercial System

At the stage of production model, the further work would be to transfer the whole program from a sun computer system to an IBM or Apple computer and run the program on-site at residence halls of MSU. The portability of the completed SPRF expert system field prototype can be troublesome. DSPL used to build the SPRF expert system requires LISP or LISP-based software (and even hardware) environments which are not readily available for the user. A finished SPRF expert system field prototype may need to be recorded in a more conventional programming language (like C) for the sake of portability. To use a production model on a regular commercial basis, more effort would be needed, especially in determining the target market, the needs of target market, and identifying existing or/and possible competitors. In addition, the user interface may need to be tailored to fit the needs of different groups of users.

# GLOSSARY

Artificial intelligence (AI). The sub-field of computer science concerned with developing intelligent computer programs. This includes programs that can solve problems, learn from experience, interpret visual scenes, and, in general, behave in a way that would be considered intelligence if observed in a human.

Domain expert. A person who, through years of training and experience, has become extremely proficient at problem solving in a particular domain.

Domain knowledge. Knowledge about the problem domain (e.g. knowledge about scheduling part-time student workers in a university residence hall foodservice operation).

User. The person who uses the finished expert system; the person for whom the system was developed.

Expert system. A computer program that uses expert knowledge to attain high levels of performance in a narrow problem area. These programs typically represent knowledge by using heuristics, and examining and explaining their reasoning processes.

Expert-system-building tool. The programming language and support package used to build the expert system.

Heuristic. A rule of thumb or simplification that limits the search for solutions in problem domains that are difficult and poorly understood.

Inference engine. The part of a knowledge-based system or expert system that contains the general problem-solving knowledge. The inference engine processes the domain knowledge (located in the knowledge base) to reach new conclusions.

Inference method. The technique used by the inference engine to access and apply the domain knowledge (e.g. forward chaining and backward chaining).

Knowledge acquisition. The process of extracting, structuring, and organizing knowledge from some source, usually human experts, so it can be used in a program.

Knowledge base. The portion of a knowledge-based system of an expert system that contains the domain knowledge.

Knowledge engineer. The person who designs and builds the expert system. This person is usually a computer scientist experienced in applied artificial intelligence methods.

Knowledge engineering. The process of building an expert system.

Knowledge representation. The process of structuring knowledge about a problem in a way that makes the problem easier to solve.

Master schedule. A schedule consisting of the jobs (including the starting and ending times) that need to be filled for each meal on each day.

Rule. A formal way of specifying a recommendation, directive, or strategy, expressed as IF (premise), THEN (conclusion) or IF (condition), THEN (action).

Search. The process of looking through the set of possible solutions to a problem in order to find an acceptable solution.

SPRF. The problem domain of scheduling part-time student workers in the university residence hall foodservice operation.

SPRF expert system. The computer program developed by this study to perform scheduling of part-time student workers in a residence hall foodservice operation.

Symbol. A string of characters that stands for some real-world concept.

Symbolic reasoning. Problem solving based on the application of strategies and heuristics to manipulate symbols standing for problem concepts.

Work shift. A particular job for a particular meal on a particular day.

# APPENDICES

Appendix A-1. Data obtained from collection of time data on manual scheduling of cafeteria part-time student workers at Mason-Abbot Hall, Michigan State University.

Page 1 of 2

| Year | Term | Number of Student Workers Employed | Scheduling Function and Its Elements | Amount of Time Needed Per Term (hrs) | Average Time Needed Per Student Worker (min) | N |
|---|---|---|---|---|---|---|
| 1989 | Spring | 75 | **Term Scheduling** | **16.65** | **13.32** | **30** |
| | | | Organize to Schedule | 1.63 | 1.31 | 5 |
| | | | Schedule | 12.18 | 9.75 | 18 |
| | | | File Schedule | 1.67 | 1.33 | 4 |
| | | | Reschedule | 1.17 | 0.93 | 3 |
| | | | **Final Scheduling** | **7.42** | **5.90** | **11** |
| | | | Organize to Schedule | 3.00 | 2.40 | 3 |
| | | | Schedule | 2.75 | 2.20 | 4 |
| | | | Copy Schedule | 0.42 | 0.30 | 2 |
| | | | File Schedule | 1.00 | 0.80 | 1 |
| | | | Reschedule | 0.25 | 0.20 | 1 |
| | | | **Volunteer Scheduling** | **1.00** | **0.80** | **1** |
| 1989 | Fall | 88 | **Term Scheduling** | **78.80** | **53.75** | **64** |
| | | | Organize to Schedule | 2.00 | 1.36 | 3 |
| | | | Schedule | 48.67 | 33.18 | 27 |
| | | | Copy Schedule | 2.50 | 2.50 | 2 |
| | | | Reschedule | 25.67 | 17.50 | 32 |
| | | | **Final Scheduling** | **13.25** | **9.03** | **16** |
| | | | Organize to Schedule | 3.00 | 2.05 | 2 |
| | | | Schedule | 8.67 | 5.91 | 9 |
| | | | Reschedule | 1.58 | 1.08 | 5 |
| | | | **Volunteer Scheduling** | **0.00** | **0.00** | **0** |

Appendix A-1. (Continuted) 

| Year | Term | Number of Student Workers Employed | Scheduling Function and Its elements | Amount of Time Needed Per Term (hrs) | Average Time Needed Per Student Worker (min) | N |
|---|---|---|---|---|---|---|
| 1990 | Winter | 75 | **Term Scheduling** | **17.50** | **14.00** | **24** |
| | | | Organize to Schedule | 0.92 | 0.73 | 2 |
| | | | Schedule | 12.75 | 10.20 | 15 |
| | | | File Schedule | 1.75 | 1.40 | 2 |
| | | | Reschedule | 2.08 | 1.67 | 5 |
| | | | **Final Scheduling** | **4.91** | **3.94** | **8** |
| | | | Organize to Schedule | 1.00 | 0.80 | 1 |
| | | | Schedule | 3.83 | 3.07 | 6 |
| | | | Delegate | 0.08 | 0.07 | 1 |
| | | | **Volunteer Scheduling** | **0.00** | **0.00** | **0** |

**N** = Quantity of entries obtained and added together to determine amount of time needed per term.

Appendix A-2. Data obtained from collection of time data on manual scheduling of cafeteria Part-time student workers at Shaw Hall, Michigan State University.



| Year | Term | Number of Student Workers Employed | Scheduling Function and Its Elements | Amount of Time Needed Per Term (hrs) | Average Time Needed Per Student Worker (min) | N |
|---|---|---|---|---|---|---|
| 1989 | Spring | 120 | **Term Scheduling** | **49.06** | **24.55** | **41** |
| | | | Organize to Schedule | 10.78 | 5.39 | 8 |
| | | | Schedule | 26.70 | 13.35 | 18 |
| | | | Copy Schedule | 3.75 | 1.88 | 6 |
| | | | File Schedule | 5.83 | 2.92 | 6 |
| | | | Reschedule | 0.75 | 0.38 | 1 |
| | | | Delegate | 1.25 | 0.63 | 2 |
| | | | **Final Scheduling** | **12.67** | **6.33** | **6** |
| | | | Schedule | 12.67 | 6.33 | 6 |
| | | | **Volunteer Scheduling** | **14.25** | **7.14** | **9** |
| | | | Organize to Schedule | 2.00 | 1.88 | 2 |
| | | | Schedule | 2.00 | 1.00 | 2 |
| | | | Check Schedule | 1.50 | 0.75 | 1 |
| | | | Copy Schedule | 5.75 | 2.88 | 3 |
| | | | Delegate | 1.25 | 0.63 | 1 |
| 1989 | Fall | 135 | **Term Scheduling** | **161.50** | **71.78** | **35** |
| | | | **Final Scheduling** | **34.50** | **15.33** | **10** |

Appendix A-2. (Continuted) 

| Year | Term | Number of Student Workers Employed | Scheduling Function and Its Elements | Amount of Time Needed Per term (hrs) | Average Time Needed Per Student Worker (min) | N |
|---|---|---|---|---|---|---|
| 1990 | Winter | 120 | **Term Scheduling** | **100.00** | **50.01** | **65** |
| | | | Organize to Schedule | 2.25 | 1.13 | 4 |
| | | | Schedule | 66.50 | 33.25 | 25 |
| | | | Check Schedule | 7.25 | 3.63 | 13 |
| | | | Copy Schedule | 14.50 | 7.25 | 11 |
| | | | File Schedule | 0.50 | 0.25 | 2 |
| | | | Reschedule | 9.00 | 4.50 | 10 |
| | | | **Final Scheduling** | **18.00** | **9.00** | **7** |
| | | | Schedule | 7.00 | 3.50 | 2 |
| | | | Check Schedule | 3.50 | 1.75 | 2 |
| | | | Copy Schedule | 7.50 | 3.75 | 3 |
| | | | **Volunteer Scheduling** | **2.00** | **1.01** | **4** |
| | | | Organize to Schedule | 0.50 | 0.25 | 1 |
| | | | Schedule | 1.00 | 0.50 | 1 |
| | | | Check Schedule | 0.25 | 0.13 | 1 |
| | | | Copy Schedule | 0.25 | 0.13 | 1 |

* **N** = Quantity of entries obtained and added together to determine amount of time needed per term.

Appendix A-3. Data obtained from collection of time data on manual scheduling of cafeteria part-time student workers at Brody Complex, Michigan State university.



| Year | Term | Number of Student Workers employed | Scheduling Function and Its Elements | Amount of Time Needed Per Term (hrs) | Average Time Needed Per Student worker (min) | N |
|---|---|---|---|---|---|---|
| 1989 | Spring | 235 | **Term Scheduling** | **46.92** | **11.98** | **37** |
| | | | Organize to Schedule | 5.92 | 1.51 | 5 |
| | | | Schedule | 27.83 | 7.11 | 23 |
| | | | Check Schedule | 4.00 | 1.02 | 1 |
| | | | Copy Schedule | 2.00 | 0.51 | 1 |
| | | | File Schedule | 0.50 | 0.13 | 1 |
| | | | Reschedule | 6.67 | 1.70 | 6 |
| | | | **Final Scheduling** | **9.52** | **2.43** | **10** |
| | | | Schedule | 9.52 | 2.43 | 10 |
| | | | **Volunteer Scheduling** | **13.72** | **3.50** | **7** |
| | | | Organize to Schedule | 11.05 | 2.82 | 5 |
| | | | Schedule | 2.67 | 0.68 | 2 |
| 1989 | Fall | 200 | **Term Scheduling** | **40.30** | **12.10** | **47** |
| | | | Organize to Schedule | 13.42 | 4.03 | 12 |
| | | | Schedule | 21.88 | 6.57 | 22 |
| | | | Check Schedule | 0.50 | 0.15 | 3 |
| | | | Copy Schedule | 0.50 | 0.15 | 2 |
| | | | File Schedule | 3.33 | 1.00 | 3 |
| | | | Reschedule | 0.67 | 0.20 | 5 |

Appendix A-3. (Continued)



| Year | Term | Number of Student Workers Employed | Scheduling Function and Its Elements | Amount of Time Needed Per Term (hrs) | Average Time Needed Per Student Worker (min) | N |
|---|---|---|---|---|---|---|
| 1989 | Fall | 200 | **Final Scheduling** | **8.75** | **2.63** | **5** |
| | | | Organize to Schedule | 6.50 | 1.95 | 2 |
| | | | Schedule | 2.25 | 0.68 | 3 |
| | | | **Volunteer Scheduling** | **0.00** | **0.00** | **0** |
| 1990 | Winter | 250 | **Term Scheduling** | **78.49** | **18.84** | **74** |
| | | | Organize to Schedule | 8.50 | 2.04 | 19 |
| | | | Schedule | 44.33 | 10.64 | 23 |
| | | | Check Schedule | 10.00 | 2.40 | 8 |
| | | | Copy Schedule | 11.83 | 2.84 | 18 |
| | | | File Schedule | 3.33 | 0.80 | 5 |
| | | | Delegate | 0.50 | 0.12 | 1 |
| | | | **Final Scheduling** | **42.50** | **10.20** | **16** |
| | | | Organize to Schedule | 2.00 | 0.48 | 2 |
| | | | Schedule | 40.50 | 9.72 | 14 |
| | | | **Volunteer Scheduling** | **0.00** | **0.00** | **0** |

* **N** = Quantity of entries obtained and added together to determine amount of time needed per term.

## Appendix B. Knowledge Representation in DSPL (By Robert Hauser, Department of Computer Science, Michigan State University)

The knowledge elicited from the expert led to its presentation in DSPL. DSPL provides two useful knowledge representation facilities, agents (e.g. specialist, plan, task, step, etc.) and a design database. Expert knowledge on the scheduling process was encoded both in the agent structure and the design database. The procedural knowledge (scheduling method) was reflected in the agent structure while the declarative knowledge (e.g. job priority, the in/out times of work shifts) was represented in the design database. The other major function of the design database was to provide the structures for constructing the schedules. Agents and the design database are detailed below.

Agents were organized into hierarchies in DSPL to break down the problem. Major functions had a specialist associated with them along with their sub-hierarchy of agents. The SPRF expert system had the term specialist at the top of the hierarchy and it was responsible for the successful design of a complete term schedule. Other specialists were responsible for designing weekday, weekend, lunch, breakfast, and dinner parts of the term schedule.

The interaction of these agents, their corresponding functions provided the implementation of the problem solving method of the expert as determined from the steps of interviewing the expert and analysis of elicited knowledge. The agents functioned by changing attributes in the design database. These changes constitute the design activity and the scheduling process.

The design database contained knowledge used in the scheduling process of the agents (e.g. job priority, the in/out times of work shifts) and also the structures for storing the actual schedules that needed to be filled. Both kinds of knowledge were represented in attributes of components. Attributes were very small pieces of knowledge that could have only one value (e.g. student number). Components were related groups of attributes (e.g. student workers).

As the problem solving is expanded beyond term scheduling, more agents and design database components may be added.

Appendix C. FSA (Foodservice Scheduling Assistant; the SPRF expert system demonstration prototype) interface and scheduling strategies (By Robert Hauser, Department of Computer Science, Michigan State University).

# 1 FSA Interface

The following sections describe the current working aspects of the FSA software. A windowing interface based on Xwindows is being added to the software.

## 1.1 Entering the Master Schedules

The master schedules consist of the jobs, including the starting and ending times, that need to be filled for each meal on each day. These schedules must be complete and provided to the FSA software.

Since the schedules may change over time facilities for modifying them must be provided. The schedules are currently hard coded and not modifiable.

## 1.2 Entering the Students

The students must all be entered before scheduling. Students may be entered by selecting the "Modify Schedules/Students" choice from the FSA MENU and then selecting "Enter New Students". The user is prompted for the name of the student file. If the file exists the students are added to the file otherwise the file is created.

Then the user is prompted for the student last name, followed by the student number.

Then the student's schedule must be entered, one day at a time. Each day is entered as a list of free time periods having a starting time and ending time for each time period. A sample list for Monday could be "((400 1000) (1400 2000))". The times are entered as military time without colons. The sample indicates that the student has free time on Monday from 4am to 10am and from 2pm to 8pm.

The last entry for a student is a list of the student's previous jobs experience. The experience is represented as the types of jobs a student has had. An example would be "(flut tc dr)".

To finish entering students, the return key is presssed when prompted for the student's last name.

Currently entered students are not modifiable.

## 1.3 Changing Scheduling Parameters

Some parameters to the FSA scheduling software are deeply related to the algorithm itself and modification of these cannot be done. Other parameters could be modified by knowledgeable users to tune the scheduling. Currently all parameters are hard coded and not modifiable.

### 1.4 Running the Scheduler

To schedule the students that have been entered one just selects the "Run Scheduler" option on the FSA MENU. This begins scheduling. When prompted the user enters the file name of the file containing the students. The rest of the scheduling is non-interactive.

During the scheduling the time for different parts of the scheduling process is written to a file "FSA.RunCase".

### 1.5 Generating Output

Upon completion of the scheduling two reports are automatically generated. The first report is a listing of the master schedule generated with the jobs filled in with the students that were given that shift. This report is stored in the file "FSA.WeekSchedule.Report".

The other report generated is listing of all the students. Each student printout includes the shifts in which he or she was scheduled. This report is stored in the file "FSA.FinishedStudents.Report".

An optional report, which shows the number of empty and full shifts can be generated by selecting "Utilities" from the FSA MENU then "Analyze Jobs Empty/Filled". The report is stored in the file "FSA.Analysis.Report".

Any of the files may be printed to the local line printer.

## 2 FSA Scheduling Strategies

This section, written by Robert Hauser, is an high level description of the scheduling method used by the FSA software. The boxes roughly correspond to the specialist agents in the DSPL problem solver. For each specialist, the plan that is used is outlined.

### 2.1 Term Scheduler

Figure 1 displays the top specialist and its sub-specialists. The FSA Term Scheduler is responsible for all of the scheduling. The plan it uses calls the Partition Students specialist, then the week scheduler specialist, and finally the weekend scheduler specialist.

The partition students specialist loads in the students from the file and places them into lunch, breakfast and dinner groups by examining their schedules to determine which meal during the week the student could be scheduled into. Also experienced students are moved to the front of each list. Other students are kept in first-come-first-served order.

The weekend scheduler is much the same as the week scheduler( see section 2.2).There are two major differences. The first is that two separate weekends are to be scheduled and used alternately during the term. The second is that each

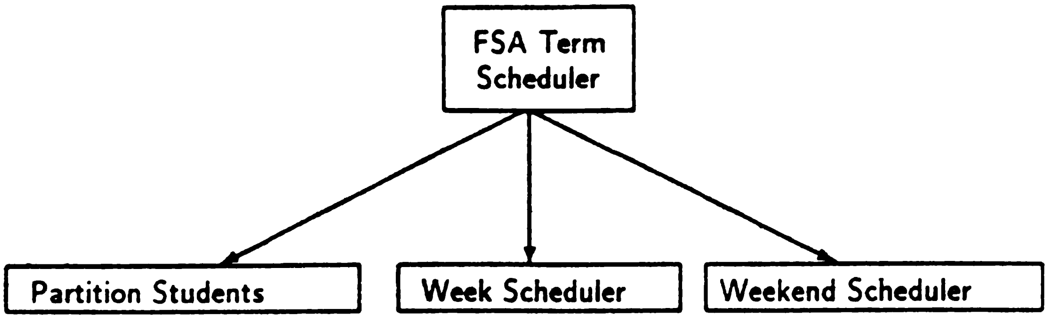



Figure 1: FSA Term Scheduler

meal of each day is considered separately. That is, Friday's dinner is scheduled, then Saturday's breakfast, then Saturday's lunch and so on.

## 2.2 Week Scheduler

Figure 2 shows the sub-specialists of the week specialist. The plan used by the week specialist is to schedule lunch then breakfast then dinner. Lunch is scheduled first because it is the most difficult meal to schedule.

Breakfast and dinner scheduling is done in the same way as lunch.

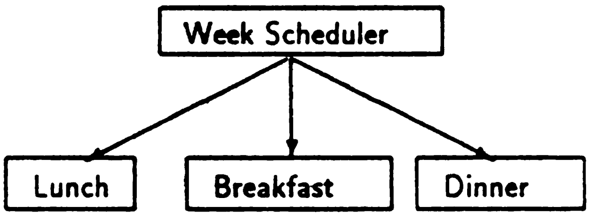



Figure 2: Week Scheduler

## 2.3 Lunch

The lunch sub-specialists are shown in figure 3. The plan that the lunch specialist uses is to call the priority specialist until all of the students in the lunch group have been scheduled. Then the fixer specialist is called.

Some of the students may not have been scheduled and some of the jobs may not be full. The fixer specialist notifies the user of any problem students or jobs.

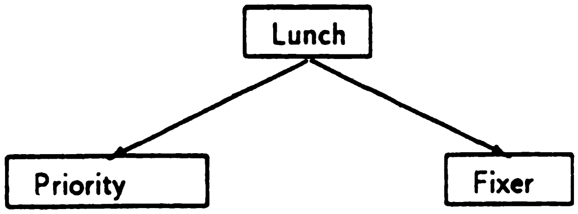



Figure 3: Lunch

## 2.4 Priority

The priority specialist's plan is to set up a group of jobs to schedule, set up the number of students needed to try to fill the jobs, and then to call the sub-specialist prioritygroups, figure 4, to schedule the group of students into the group of jobs.

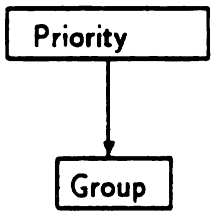



Figure 4: Priority

The jobs are listed in priority order in the design database. Therefore, the most important jobs will be filled first.

## 2.5 Group

The group specialist, shown in figure 4, is responsible for scheduling a group of students into a group of jobs. No sub-specialists are called.

The plan is to select a student and a job and schedule the student into the job until no more students remain. If the student doesn't fit another job is selected. If no job is found the start and end times of the shifts are relaxed and a fit is tried again for the jobs. If the student still does not fit then he/she joins the next group of students to be scheduled unless this type of failure has already occurred in which case the student joins the group of students designated for the next meal.

In the process of selecting a job for the student, the student's previous experience is considered. Jobs in which the student has experience are attempted first.

Appendix D. Master schedule (a sample print-out before completion of user interface).

Friday's Breakfast Schedule

STC1 --------- (1061412 640 1015)

TOAST -------- (1132330 615 1000)

SUT1 --------- (1133794 635 1000)

SL-SETUP1 ---- (113015 515 930)

SL-SETUP2 ---- (1135736 515 930)

SLS1 --------- (1139871 640 1000)

EFLUT -------- (1104450 640 1015)

ECON1 -------- (1132510 640 1015)

ECON2 -------- (1118725 700 1005)

EEH1 --------- (1126347 700 1005)

EEH2 --------- (1112655 715 1005)

EBD1 --------- (1123875 640 1000)

EBD2 --------- (1051148 700 1000)

EBD3 --------- (1136092 715 1000)

DDR1 --------- (1134555 710 1030)

DDR2 --------- (1084356 710 1030)

DDR3 --------- (123 730 1030)

DDR4 --------- (1138492 730 1030)

DPP1 --------- (1130239 630 1030)

DPP2 --------- (1114661 715 1030)

Appendix E. Student worker list (a sample print-out before completion of user interface).

Student Name Student Number Previous Experience
MAZZUIHI 1135826 (NONE)
Shifts
((BREAKFASTTUESDAY DPP1 630 1030) (LUNCHTUESDAY NCON1 1025 1430)
(LUNCHTHURSDAY NCON1 1025 1430))
Class Schedule
((MONDAY (1455 2200)) (TUESDAY (400 2200)) (WEDNESDAY (1455 2200))
(THURSDAY (400 2200)) (FRIDAY (1455 2200)))

Student Name Student Number Previous Experience
MAYS 1127438 (PP BD DR)
Shifts
NIL
Class Schedule
((MONDAY (1345 1545)) (TUESDAY (1345 1845)) (WEDNESDAY (1345 1545))
(THURSDAY (1345 1845)) (FRIDAY (1345 1545)))

Student Name Student Number Previous Experience
MAYFIELD 1106216 (NONE)
Shifts
((BREAKFASTMONDAY SUT1 635 1000) (BREAKFASTWEDNESDAY SUT1 635 1000)
(LUNCHFRIDAY DDR7 1145 1530))
Class Schedule
((MONDAY (400 1005) (1455 2200)) (TUESDAY (1455 2200)) (WEDNESDAY (400
1005) (1455 2200)) (THURSDAY (1455 2200)) (FRIDAY (400 2200)))

# LIST OF REFERENCES

## LIST OF REFERENCES


Barnes, R.M. 1980. Motion and Time Study: Design and Measurement of Work. 7th ed. New York: John Wiley & Sons.

Beach, B. L. and Ostenso, G. L. 1969. "Entree Serving Times: Relationship of Serving Time to System Capacity." Journal of The Dietetic Association. 54(4): 290-296.

Block, Arla A., Roach, Faith R., and Konz, Stephan A. 1985. "Occurrence Sampling in a Residence Hall Foodservice: Cleaning Times for Selected Vegetables." Journal of The American Dietetic Association. 85(2): 206-209.

Boehm, Barry W. et al. 1978. Characteristics of Software Quality. New York: North-Holland Publishing Company.

Boritz, Efrim J. and Brown, Donald A. 1986. "The Expert-Ease of Expert Systems." CA Magazine. 119(8): 50-56.

Breuker, Joost and Wielinga, Bob. 1987. "Use of Models in the Interpretation of Verbal Data." Knowledge Acquisition for Expert Systems. Ed. Kidd, Alison L. New York: Plenum Press.

Briggs, Steven and Doney, Lloyd D. 1988. "Can HR Expertise Be Computerized ?" Personnel Journal. 67(4): 106-111.

Britton, Scott G. 1987. "Computer-Based Expert System Aids Underground Mine Planning." Coal Age. 92(1): 69-70.

Brown, David C. "Routine Design Problem Solving." 1987. KBS in Engineering and Architecture. Ed. Gero, J. Reading (PA): Addison-Wesley.

Brown, David C. and Chandrasekaran, B. 1989. Design Problem Solving Knowledge Structures and Control Strategies. San Marcos (CA): Pitman Publisher.

Buchanan, B. et al. 1983. "Constructing an Expert Systems." Building Expert Systems. Ed. Hayes-Roth, F., Waterman, Donald A., and Lenat D. Reading (PA): Addison-Wesley.

Burns, R.N. and Carter M.W. 1985. "Work Force Size and Single Shift Schedules With Variable Demands." Management Science. 31(5): 599-607.

Chandrasekaran, B. 1983. "Towards A Taxonomy of Problem Solving Types." The AI Magazine. (Winter/Spring): 9-17.

Chandrasekaran, B. 1983. "On Evaluation AI Systems for Medical Diagnosis." The AI Magazine. (Summer): 34-37.

Chandrasekaran, B. 1986. "Generic Tasks in Knowledge-Based Reasoning: High Level Building Blocks for Expert System Design." IEEE Expert. (Fall): 23-30.

Currie, R.M. 1977. Work Study. 4th ed. London (UK): Pitman Publishing.

Curtis, John W. 1987. "Robotics and Artificial Intelligence for the Food Industry." Food Technology. 41(12): 62-64.

deHaas, Pat. 1983. "Computerizing the Reference Desk Schedule." Library Journal. 108(5): 973-975.

Desai, Harsha B. et al. 1987. "Scheduling of Professional Staff: Microcomputer Integration." CPA Journal. 57(1): 109-116.

DHFS 1988. (The Division of Housing and Food Service, Michigan State University, East Lansing, MI 48824) "How to Schedule." Summer.

Fensholt, Carol. 1988. "Labor Scheduling: New Software Enhance Savings and Service." Supermarket Business. 43(5): 55-59.

Firdman, H. E. 1986. "Artificial Intelligence: Understanding the Basic Concepts." Design News. 42(3): 89-95.

Gaschnig, J. 1982. "Application of the PROSPECTOR System to Geological Exploration Problems." Machine Intelligence. 10: 301-323.

Gaschnig, J. et al. 1983. "Evaluation of Expert Systems: Issues and Case Studies." Building Expert Systems. Ed. Hayes-Roth, F., Waterman, Donald A., and Lenat D. Reading (PA): Addison-Wesley.

Geissman, James R. and Schultz, Roger D. 1988. "Verification and Validation of Expert Systems." AI Expert. 3(2): 26-33.

Gevarter, William B. 1987. "The Nature and Evaluation of Commercial Expert System Building Tools."Computer. (May): 24-40.

Hayes-Roth, F., Waterman, D. A., and Lenat D. 1983. Building Expert Systems. Reading (PA): Addison-Wesley.

Harrison, Patrick R. 1989. "Testing and Evaluation of Knowledge-Based Systems." Structuring Expert Systems. Ed. Liebowitz, Jay and De Salvo, Daniel A. Englewood Cliffs (NJ): Yourdon Press.

Ho, A. and Matthews, M.E. 1978. "Activity Sampling in Two Nursing Home Foodservice Systems." Journal of The American Dietetic Association. 73(12): 647-653.

Krebs, Valdis E. 1987. "How Artificial Intelligence Assists HR Managers." Personnel. 64(9): 58-62.

Kern, J., Hauser, R., and Sticklen Jon. 1989. "Configuration and Scheduling: A Preliminary Report on Using A Routine Design Approach." Proceedings of First Great Lakes Computer Conference, Oct. (Grand Rapids, MI).

Kidd, Alison L. Ed. 1987. Knowledge Acquisition for Expert System. New York: Plenum Press.

Kinoglu, Faith. 1986. "Artificial Intelligence: Expert System Model of the Design Process." Design News. 42(3): 149-156.

Kuipers, Benjamin and Kassirer Jerome P. 1987. "Knowledge Acquisition by Analysis of Verbatim Protocols." Knowledge Acquisitionfor Expert Systems. Ed. Kidd, Alison L. New York: Plenum Press.

Liebowitz, Jay. 1986. "Useful Approach for Evaluating Expert Systems." Expert Systems. 3(2): 86-96.

Liebowitz, Jay. 1989. "Problem selection for Expert Systems Development." Structuring Expert Systems. Ed. Liebowitz, Jay and De Salvo, Daniel A. Englewood Cliffs (NJ): Yourdon Press.

Lippert, Renate. 1987. "Expert Systems: New Instructional Tools." ADCIS, Nov. (San Francisco, CA).

Matthews, M.E., Zardain, M.S., and Mahaffey, M.J. 1986. "Labor Time Spent in Foodservice Activities in One Hospital: A 12-Year Profile." Journal of The American Dietetic Association. 86(5): 636-643.

Mundel, M.E. 1978. Motion and Time Study: Improving Productivity. 5th ed. Englewood Cliffs (NJ): Prentice-Hall.

Miller, F.D. et al. 1985. "The ACE Experiment: Initial Evaluation of an Expert System for Preventive Maintenance." Artificial Intelligence in Maintenance. Ed. Richardson, J.J. Park Ridge (NJ): Noyes Publications.

Naughton, James M. 1989. "A Specific Knowledge Acquisition Methodology for Expert Systems Development: A Brief Look at Precision Knowledge Acquisition." Structuring Expert Systems. Ed. Liebowitz, Jay and De Salvo, Daniel A. Englewood Cliffs (NJ): Yourdon Press.

"Computer Solves Labor Scheduling Mess." 1986. Chain Store Age Executive with Shipping. 62(9): 148-152.

Olsen, Michael D. and Meyer, Mary Kay. 1987. "Current Perspectives on Productivity in Food Service and Suggestions for the Future." School Food Service Research Review. 11(2): 87-92.

Otten, Alan L. 1990. "Part-Time Work Is Fine, If It Is What You Want." The Wall Street Journal. (April): B1.

Prerau, David S. 1985. "Selection of an Appropriate Domain for an Expert System." The AI Magazine. (Summer): 26-41.

Rauch-Hindin, Wendy B. 1988. A Guide to Commercial Artificial Intelligence. Englewood Cliffs (NJ): Prentice-Hall. pp 1-110.

Rich, Elaine. 1983. Artificial Intelligence. New York: McGraw-Hill, Inc.

Roberts, Hilary. 1988. "Expert Systems and the Personnel Department." Personnel Management. 20(6): 52-56.

Rodammer, Frederick A. and White, K. Preston. 1988. "A Recent Survey of Production Scheduling." IEEE Transactions on Systems, Man, and Cybernetics. 18(6): 841-851.

Rothenberg, Jeff et al. 1987. Evaluating Expert System Tools: A Framework and Methodology. Santa Monica (CA): RAND Corporation.

Slagle, James. and Wick, Michael. 1988. "A Method for Evaluating Candidate Expert System Applications." The AI Magazine. 9(3): 44-53.

Stinson, Joel P. and Guley, Helen M. 1982. "Use of a Branch and Bound Algorithm to Schedule Food Production in a Semi-Conventional Foodservice System." Journal of The American Dietetic Association. 81(11): 562-567.

Taylor, W.A. 1986. "Artificial Intelligence: Potentials and Limitations." Design News. 42(3): 75-83.

Ten Dyke, Richard P. 1986. "Artificial Intelligence: Integrating Expert Systems." Design News. 42(3): 131-133.

The CBORD Group, Inc. 1984. Labor Scheduling System: for IMB PS/2, PC XT, AT. Ithaca, NY.

Turpin, Bill. 1986. "Artificial Intelligence: Project Needs." Design News. 42(3): 104-113.

Vedder, Richard. 1987. "The Opportunities and Challenges of Managing Expert Systems." IM. (January): 14-16.

Waterman, Donald A. 1986. A Guide to Expert Systems. Reading (PA): Addison-Wesley.

Waterman, Donald A. and Peterson, Mark A. 1984. "Evaluating Civil Claims: an Expert Systems Approach." Expert Systems. 1(1): 65-76.

Williams, J. E. and Donaldson, B. 1969. "SCORE: A Management Evaluation Program for Dietary Departments." Journal of The American Dietetic Association. 54(4): 283-289.

Zemel, P. C. and Matthews, M.E. 1982. "Determining Labor Productivity Time for Roast Entree in Hospital Foodservice." Journal of The American Dietetic Association. 81(12): 709-714.