NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

A MICRO-COMPUTER BASED EMPLOYEE SCHEDULING SYSTEM
FOR THE
PALO ALTO VETERANS ADMINISTRATION MEDICAL CENTER

by

JAMES M. BUYSKE
and
STEVEN V. CALL

September 1988

Thesis Advisor Tarek Abdel-Hamid

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.
2b. DECLASSIFICATION / DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION NAVAL POSTGRADUATE SCHOOL 6b OFFICE SYMBOL (If applicable) 37	7a NAME OF MONITORING ORGANIZATION NAVAL POSTGRADUATE SCHOOL
6c. ADDRESS (City, State, and ZIP Code) MONTEREY, CA 93943-5100	7b. ADDRESS (City, State, and ZIP Code) MONTEREY, CA 93943-5100
8a NAME OF FUNDING / SPONSORING ORGANIZATION 8b OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (City, State, and ZIP Code)	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO / WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification)
A MICRO-COMPUTER BASED EMPLOYEE SCHEDULING SYSTEM FOR THE PALO ALTO VETERANS ADMINISTRATION MEDICAL CENTER

12 PERSONAL AUTHOR(S)
BUYSKE, JAMES M.; CALL, STEVEN V.

13a TYPE OF REPORT	13b TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
MASTER'S THESIS	FROM _____ TO _____	1988 SEPTEMBER	92

16 SUPPLEMENTARY NOTATION
THE VIEWS EXPRESSED IN THIS THESIS ARE THOSE OF THE AUTHORS AND DO NOT REFLECT THE OFFICIAL POLICY OR POSITION OF THE DEPARTMENT OF DEFENSE OR THE U.S. GOVERNMENT.

17 COSATI CODES		
FIELD	GROUP	SUB-GROUP

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
COMPUTER SCHEDULING SYSTEM; PROTOTYPING; DATABASE; SYSTEM ANALYSIS AND DESIGN, Dietetics, MEDICAL CENTER, VES

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This thesis provides the Palo Alto Veterans Administration Dietetics Division with a micro-computer based employee scheduling system. It includes a discussion of the system analysis and design, program development, and system implementation. The use of a prototyping development methodology and its implications is a central discussion point. theses

20 DISTRIBUTION / AVAILABILITY OF ABSTRACT	21 ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL TAREK ABDEL-HAMID	22b TELEPHONE (Include Area Code) 408-646-2686 22c OFFICE SYMBOL 54Ah

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

* U.S. Government Printing Office 1986—606-24.

Approved for public release; distribution is unlimited.

A Micro-Computer Based Employee
Scheduling System for the Palo Alto
Veterans Administration Medical Center

by

James Michael Buyske
Lieutenant Commander, United States Navy
B.S., University of Idaho, 1976

and

Steven Verdell Call
Lieutenant, United States Navy
B.S., University of New York, 1981

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
September 1988

Authors:

James Michael Buyske

Steven Verdell Call

Approved by:

Tarek Abdel-Hamid, Thesis Advisor

James E. Suchan, Second Reader

David R. Whipple, Chairman,
Department of Administrative Sciences

Kneale T. Marshall,
Dean of Information and Policy Sciences

ABSTRACT

This thesis provides the Palo Alto Veterans Administration Dietetics Division with a micro-computer based employee scheduling system. It includes a discussion of the system analysis and design, program development, and system implementation. The use of a prototyping development methodology and its implications is a central discussion point.

TABLE OF CONTENTS

I. INTRODUCTION

A. DESCRIPTION OF PROBLEM AND ENVIRONMENT

The Chief Dietician's staff at the Palo Alto, California, Veterans Administration Medical Center currently uses a manual system to schedule 137 food service positions on a biweekly basis. This schedule is normally compiled by the Food Production and Service Unit Manager (Primary Scheduler), see Figure 1, who draws from a large personal knowledge base of the employees, positional requirements, and scheduling heuristics to assign personnel to these positions. This individual is the only person within the supervisory positions who has the requisite knowledge of personnel and scheduling heuristics to complete an effective work schedule within a reasonable period of time. The Primary Scheduler's manual process requires approximately three hours to produce a completed schedule.

One weakness of having only a single person capable of completing the schedule was emphasized when the primary scheduler required an unexpected medical leave of absence. He was neither available to complete the schedule himself nor to answer questions which arose when his superior assumed the scheduling function. With limited knowledge of personnel history and scheduling heuristics, the superior found that the scheduling procedure required an extensive amount of valuable time (15 hours) in order to complete a single

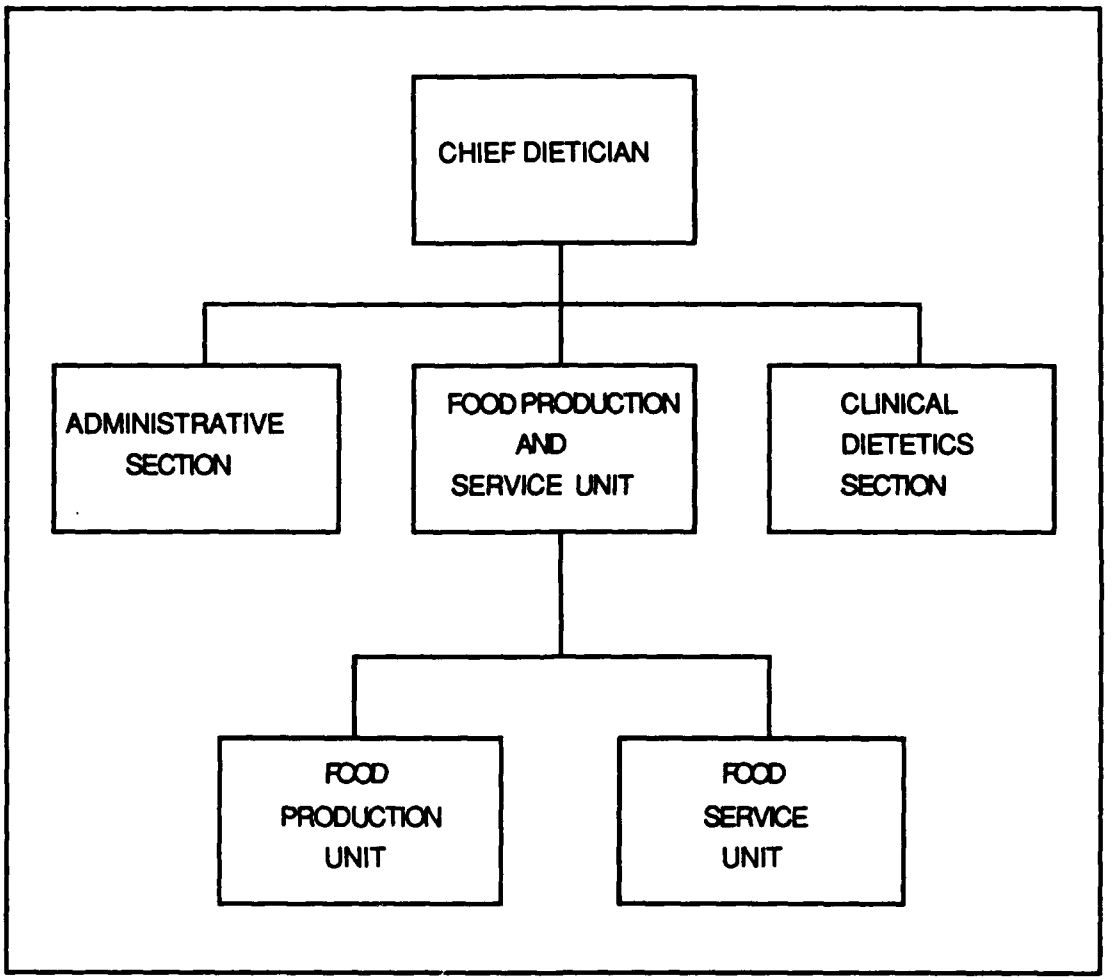


Figure (1) Dietetic Service Organizational Chart

biweekly schedule. The superior surmised that this scheduling process would be further delayed if the task were assigned to a less experienced person.

Another weakness of the current scheduling process is the lack of clear, structured, and written guidelines. The assignment of jobs is a subjective process based upon the Primary Scheduler's personal preferences and heuristics. For example, whenever a primary employee requires time off (for a day off, annual leave, or during other periods of absence) the scheduler assigns a substitute, called a relief. The Food Services Unit maintains a pool of employees from which the Primary Scheduler can assign a relief based upon his personal knowledge of that particular employee's capabilities. This particular practice has caused a few relief employees to always be assigned to certain jobs while other relief employees were seldom allowed into these positions. This has resulted in accusations of favoritism.

Assigning the schedule to an alternate scheduler who is unfamiliar with the Primary Scheduler's personal and informal scheduling processes can and did lead to discontinuity, confusion among the employees, and inefficient work assignments. The alternate scheduler, without knowledge of these "normal" relieving assignments, scheduled relief employees to job positions for which they were not qualified. This caused confusion and tardiness among relief personnel

who were assigned to relieve job positions for which they had no previous experience.

The problems encountered by the alternate scheduler highlighted the need to broaden the relief employee job experience base. The Primary Scheduler would like to accomplish this by instituting a job position rotation which would result in providing cross-training for the primary and relief employees. However, due to the complexity of this type of system and the time needed to manage the rotation cycles, the Primary Scheduler feels it would be too difficult to institute in the current manual scheduling system.

In light of all the above mentioned problems, the Chief Dietician identified a need for a more time efficient and formalized computer-based scheduling system. The system must be capable of being operated by supervisory personnel other than the Primary Scheduler in the event of his absence. These alternate schedulers must be able to create a schedule that is consistent with the previous schedules even though they may have limited knowledge of the work environment and the scheduling process.

The Primary Scheduler has no computer experience and has indicated no desire to become personally involved in using a computerized scheduling program. The Chief Dietician (the Primary Scheduler's immediate supervisor) and her administrative staff (alternate schedulers) have a working knowledge of microcomputers and routinely use word processing

and database applications. The lack of computer literacy on the part of the Primary Scheduler requires special consideration during the system design and implementation in order to increase the likelihood of his acceptance and use of the system.

B. THESIS GOAL

The objective of this thesis is to conduct a system analysis of the current scheduling process and to design an automated system based upon the current general needs and requirements as identified by the Chief Dietician, her Primary Scheduler, and the analysis team. The function of the automated scheduling system will be to produce a 14-day work schedule in which all employees are displayed with their daily job assignments. It will ensure every primary job is filled by a qualified employee (either the primary employee or a qualified relief). The automated system will also provide a consistent method of assigning relief personnel to primary jobs when substitution is required.

By developing an automated scheduling system based upon these initially defined functions, the users will be provided a system with the following benefits:

1. Create a time efficient system which can be used by a primary or alternate scheduler to produce a schedule in a time efficient manner. It is estimated that the proposed system will be able to produce a completed schedule within 15-20 minutes when operated by an experienced scheduler (one who is familiar with employees job related capabilities) and within 30 minutes by an inexperienced scheduler (one who will require inputs from supervisory personnel concerning

employee job related capabilities). This system represents a time savings of 89% (20 min. vs. 3 hrs.) for the Primary Scheduler, and up to 97% (30 min. vs. 15 hrs.) for an inexperienced scheduler. The input required by an inexperienced user consists of the occasional need for a decision on who to assign as a relief when all available reliefs fall outside the normally acceptable parameters. This situation only occurs on average of two to three times during the scheduling process (1932 events).

2. Formalize a structured scheduling system that is not dependent upon the personal techniques or heuristics of differing schedulers with respect to relief assignments. This system will reduce inconsistencies in job assignments, thereby decreasing employee dissatisfaction.

3. Reduce the number of errors that can result from a manually generated schedule. These errors take the form of double scheduling, jobs not being scheduled for reliefs, and inaccuracies in the employee/job position databases.

4. Reduce the reliance upon a single employee for schedule production, thereby providing flexibility in who can prepare the schedule.

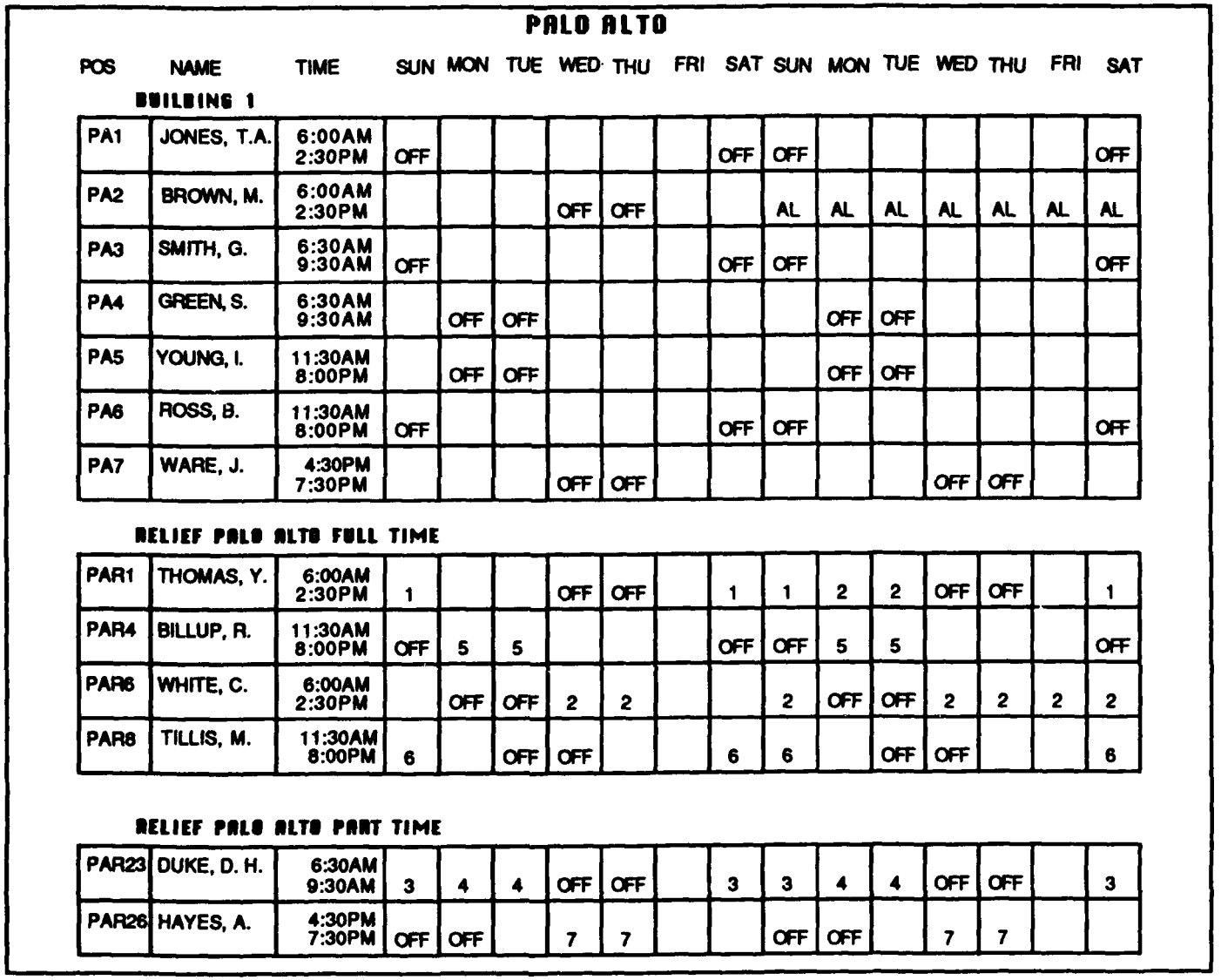
II. SYSTEM ANALYSIS

A. SYSTEM OVERVIEW

The Primary Scheduler, who is also the Food Production and Service Unit Manager, has stated that the primary objective of his job is to ensure the dietary needs of all hospital patients are met on a daily basis. This goal is accomplished through his role as scheduler by ensuring that all dietary job positions are assigned to employees who are qualified to carry out the job responsibilities. This task must be done within the limitations and constraints imposed by federal, hospital, departmental, and union regulations.

The Primary Scheduler has direct authority for assigning employees to job positions as well as for schedule creation. On a biweekly basis a work schedule for the next fourteen-day pay period is produced. An abbreviated sample schedule is provided in Figure 2.

The sample schedule shown represents a portion of the entire schedule that is manually generated to schedule all Food Service employees currently assigned to the Palo Alto division. A similar schedule is also created for the Menlo Park division. Each employee is listed along with their primary job position, work shift times, job location, and daily work assignments. The employees' daily work assignments are coded in the following manner:



PALO ALTO

POS	NAME	TIME	SUN	MON	TUE	WED	THU	FRI	SAT	SUN	MON	TUE	WED	THU	FRI	SAT
BUILDING 1																
PA1	JONES, T.A.	6:00AM 2:30PM	OFF						OFF	OFF						OFF
PA2	BROWN, M.	6:00AM 2:30PM				OFF	OFF			AL	AL	AL	AL	AL	AL	AL
PA3	SMITH, G.	6:30AM 9:30AM	OFF						OFF	OFF						OFF
PA4	GREEN, S.	6:30AM 9:30AM		OFF	OFF						OFF	OFF				
PA5	YOUNG, I.	11:30AM 8:00PM		OFF	OFF						OFF	OFF				
PA6	ROSS, B.	11:30AM 8:00PM	OFF						OFF	OFF						OFF
PA7	WARE, J.	4:30PM 7:30PM				OFF	OFF						OFF	OFF		
RELIEF PALO ALTO FULL TIME																
PAR1	THOMAS, Y.	6:00AM 2:30PM	1			OFF	OFF		1	1	2	2	OFF	OFF		1
PAR4	BILLUP, R.	11:30AM 8:00PM	OFF	5	5				OFF	OFF	5	5				OFF
PAR6	WHITE, C.	6:00AM 2:30PM		OFF	OFF	2	2			2	OFF	OFF	2	2	2	2
PAR8	TILLIS, M.	11:30AM 8:00PM	6		OFF	OFF			6	6		OFF	OFF			6
RELIEF PALO ALTO PART TIME																
PAR23	DUKE, D. H.	6:30AM 9:30AM	3	4	4	OFF	OFF		3	3	4	4	OFF	OFF		3
PAR26	HAYES, A.	4:30PM 7:30PM	OFF	OFF		7	7			OFF	OFF		7	7		

Figure (2) Abbreviated Sample Schedule Format

1. A blank space means a normal work day in their primary position.

2. OFF indicates a normally scheduled day off.

3. AL indicates that the employee is on annual leave during this time (e.g., PA2 in the sample schedule).

4. A number indicates that this employee is to relieve the primary position identified by that number. For example, a number 5 on the Palo Alto schedule means that the primary position to be relieved is PA5.

The Food Production and Service Unit is limited to a specific maximum work force level of 137 employees by Federal allocation. The allocation is further broken down by the number of employees for each of the Federal Civil Service Wage Grades. Each job position is assigned a minimum recommended employee wage grade based on the required skill level. This wage grade assignment is determined by the Dietetic Services management and ranges from Wage Grade 1 (WG-1) through Wage Grade 4 (WG-4), with the WG-4 position requiring the senior, more experienced employee.

For each job position a set of duties and responsibilities are delineated by hospital regulations. Figure 3 provides a sample job description.

Job positions are divided into two major categories: primary positions and relief positions. Primary positions are those which are required to meet the minimum hospital needs on a day-to-day basis. They are also assigned to a particular location (building and work area) and perform the same daily work assignments. Primary positions are shown in

POSITION: PA9
LOCATION: BUILDING 2
JOB TITLE: CHARGE PERSON
SHIFT TIME: 6:00AM-2:30PM

DUTIES:

6:00	Sign on duty in proper uniform. Read Bulletin Board. Sign for key. Go to assigned area.
6:40	Turn on all conveyors, toaster, and diet warmer. Do diet changes. Check hot cart for foods needed. Check for polycose and bran cereal. 1. Put up menu. 2. Set up hot cart.
7:00	Check and load trays. Continue until service is finished.
7:50	Break down first two carts of dishes. Continue to break down soiled dishes until all wards are finished. Take out garbage. Secure breaking-down area. Help where needed until break time.
9:00	Break - 15 minutes.
9:15	Do special cleaning.
9:45	Check menu for items needed for noon meal.
10:00	Put up special items.
10:25	Put cards in order of service. Change soiled cards.
10:35	Assist with dishing up cold food for noon meal and refrigerate.
10:50	Lunch.
11:20	Return from lunch.
11:30	Prepare to check trays until all trays are served.
12:45	Prepare for dish washing. Continue until all dishes are finished.
1:45	Break - 15 minutes.
2:00	Scrub floor and do any special cleaning as time permits. Help put up dishes for supper meal.
2:30	Sign off duty in proper uniform.

NOTE: Perform any other duties as assigned.
Work safely at all times.

Figure (3) Sample Job Description

Figure 2 as PA1 through PA7. Relief positions are indicated in Figure 2 by having the letter "R" in the position identifier (e.g., PAR8). Relief positions are used mainly to relieve primary positions during periods of absence by the primary employee. Those relief employees not assigned to relieve a primary position on a particular day are assigned on a daily basis to assist in areas as designated by the local shift supervisor. Primary and relief positions are divided into full-time (8 hours) and part-time (3 hours). Part-time employees are used during peak meal hours to supplement full-time employees. These positions are further split into early (AM) and late (PM) shifts. There are four full-time AM shifts, two full-time PM shifts, one part-time AM shift, and three part-time PM shifts (see Figure 4).

FULL-TIME		PART-TIME	
AM	PM	AM	PM
5:30AM-2:00PM	10:30AM-7:00PM	6:30AM-9:30AM	4:00PM-7:00PM
6:00AM-2:30PM	11:30AM-8:00PM		4:30PM-7:30PM
7:30AM-4:00PM			5:00PM-8:00PM
8:00AM-4:30PM			

Figure (4) Work Shifts Designations

To produce a schedule, the scheduler executes the following steps:

1. Obtains a schedule format listing all job positions, the current assignment of employees to those positions, and a blank 14-day matrix. This matrix is maintained and updated as to employee changes by the administrative office.

2. Assigns leave days corresponding to a previously arranged leave schedule, as shown by PA2 in Figure 2.

3. Rotates each job position's days-off one day earlier in relation to the previous schedule. For example, in Figure 2 PA1's next days-off will rotate to become Friday/Saturday, PA2's next days-off will become Tuesday/Wednesday, etc.

4. Assigns a relief employee for each primary position that requires a relief. To do this he must: access the pool of relief employees, match employee qualifications with job requirements, check relief employee availability, and assign the relief employee.

B. SYSTEM LIMITS AND CONSTRAINTS

The following have been identified as the limits and constraints in the scheduling process:

1. All primary positions must be filled every day of every week.

2. Each of the primary job positions should be relieved by a relief employee of the same or higher wage grade. If this is not possible, then a relief employee should be selected based upon the scheduler's knowledge of available employees' experience levels. If the scheduler lacks first-hand knowledge of employee experience levels, then assistance from the employee's immediate supervisor will be required.

3. Union contract terms stipulate that employees can not be scheduled to work both AM and PM shifts within a 14-day schedule period. Union terms also state that employees can not be assigned to work both full-time and part-time positions.

4. Positions are located in the two California cities, Palo Alto (designated as "PA") and Menlo Park (designated as "MP"). Employees will not be alternated between cities during a schedule period. That is, Palo Alto reliefs will only relieve positions in Palo Alto. This represents a policy established by the Dietetic Services Department so that employees do not have to change their commuting arrangements on short notice.

5. Each position is assigned four days-off per 14-day schedule period. When an employee is assigned a primary position, he or she assumes that position's designated days-off. For example if Ware, J., currently assigned to PA7 in Figure 2, were to be reassigned to PA5, his new days-off would become Monday/Tuesday.

6. Employees are not to work more than six consecutive days without a day off. This six-day limit applies to the transition between previous and subsequent schedules as well. The last non-work day, either scheduled day off or leave, is considered the start date for the six consecutive work day constraint.

7. By contractual agreement with the employee union, days off shift one day earlier in the week, with each new schedule period. Departmental policy requires that days-off be two consecutive days. Thus a Tuesday/Wednesday this period shifts to Monday/Tuesday on the following schedule. An exception to the two consecutive days-off policy occurs during a schedule period in which an employee is assigned Saturday/Sunday as days-off. As can be seen in Figure 2, PA1 has only a single day off at the beginning of the schedule, two consecutive days-off in the middle, and a single day off at the end. (This is a phenomena of the days-off shifting requirement).

8. Each primary position currently has a designated relief employee who is normally used to relieve the primary employee on his/her days-off and periods of leave. If available, this designated relief employee is always chosen when assigning a relief for the primary employee. If, however, the designated relief employee is not available, Departmental policy allows any relief employee who meets the job requirements to be assigned to the primary position. These job requirements include, wage grade, same shift time (AM/PM), job status (Full-time or Part-time), and job location (Palo Alto/Menlo Park).

9. In order to provide some job continuity during long periods of absences (leave) by primary employees, the scheduler may assign specific relief employees (Leave Reliefs), other than the designated relief, to relieve these positions for the entire period. Currently these Leave Relief positions are unique to Palo Alto and are identified as PAR18, PAR19, PAR20, and PAR21. That is, PAR18 could, for example, be assigned to relieve PA1 during periods of annual leave. In these

cases the Leave Relief employee will assume the days-off of the position he is relieving.

10. In order to provide a broader base of experience, the Dietetic Department has determined that employees should be rotated among different positions on a periodic basis. The Primary Scheduler has been given direct responsibility for implementing this rotation plan.

    a. The rotation should take place within groups defined by job positions with matching shifts and wage grades. (Figure 5 shows the proposed groupings of positions). With each new schedule, one group (e.g., AM WG-4) will rotate all employees within that group down one position (e.g., the employee assigned to PA1 will now be assigned to PA25). This results in all employees changing job positions once every 16 weeks.

    b. There are some employees who are limited in their ability to perform particular tasks due to physical handicaps. They are, therefore, required to remain in their present position. For example, presently there are two employees with reading disabilities which preclude them from moving to a job position which requires the reading of a menu. These particular individuals should not be included in the job rotation cycle.

AM WG-4	PM WG-4	AM WG-3	PM WG-3	AM WG-2	PM WG-2	AM PT	PM PT
PA1	PAR4	PA30	PA6	PAR13	PA38	PA3	PA46
PA25	PA21	PA9	PAR8	PAR11	PA24	PA16	PA8
PA51	PAR5	PAR10	PAR21	PA39	PAR16	PAR25	PAR27
PAR1	PA17	PA29	PA12	PA20	PA13	PAR23	PA47
PA28	PA5	PAR7	MP6	PA35	PA45	PA11	PA7
PA15	PAR19	PAR20	MPR7	PA10	PA50	PA42	PA18
PAR2	PA26	PA32	MP28	PA40	PA43	PA4	PAR28
MP21	MPR4	PA2		PA33	PA36	PAR24	PAR30
MP1	MP13	PA48		PA49	PAR15	PA41	PA14
MPR3	MPR5	MPR6		PA37	PAR14	PAR22	PAR26
MP9	MP5	PAR6		PAR12	PA44	MPR13	PAR29
MPR1	MP27	PAR9		PAR17	MPR10	MPR12	PA22
MP22	MP14	PA31		PA34	MP29	MPR15	MPR18
MP2		PA23		MP33	MP19	MP18	MPR21
PAR18		MP17		MP3	MPR11	MP23	MP32
PAR3				MPR9	MP7	MP4	MP30
PA19				MPR8	MP34	MP25	MP16
PA27						MP24	MPR20
PA52						MP12	MP8
MP10						MPR14	MP15
MPR2						MP11	MPR16
						MP26	MPR17
							MP31
							MPR19
							MPR22
							MP20

AM -- AM work shift
PM -- PM work shift
WG -- wage grade
PA -- Palo Alto
MP -- Menlo Park

Figure (5) Proposed Job Rotation Groups

C. SYSTEM ANALYSIS SUMMARY

A graphical analysis of the scheduling system is depicted in the data flow diagrams of Figures 6, 7, and 8. These figures are a hierarchical representation of the scheduling system.

1. Scheduling System Context Diagram.

Figure 6 shows a simplified, overall view of the interactions that have an impact on the scheduling system. This overall view is called a Context Diagram. The circle labeled 'SCHEDULING SYSTEM' depicts the process of all activities, both manual and automated, necessary in the production of a new schedule. The three rectangles represent entities which contribute and receive data from the scheduling activity. The arrows represent both the data and direction of data flow within the system.

The ADMINISTRATIVE SECTION provides NEW POSITION DATA and NEW EMPLOYEE DATA. NEW POSITION DATA contains information concerning any changes to current job positions (i.e., the addition or deletion of positions). NEW EMPLOYEE DATA contains information about newly hired, terminated, or promoted employees. The FOOD PRODUCTION & SERVICE UNIT MANAGER provides SCHEDULE REQUEST and SPECIFIC RELIEF data. The SCHEDULE REQUEST is an input that simply triggers the start of the scheduling system activity and consists of either a recurring date function or an ad hoc request from the scheduler. The SPECIFIC RELIEF data represents a manual

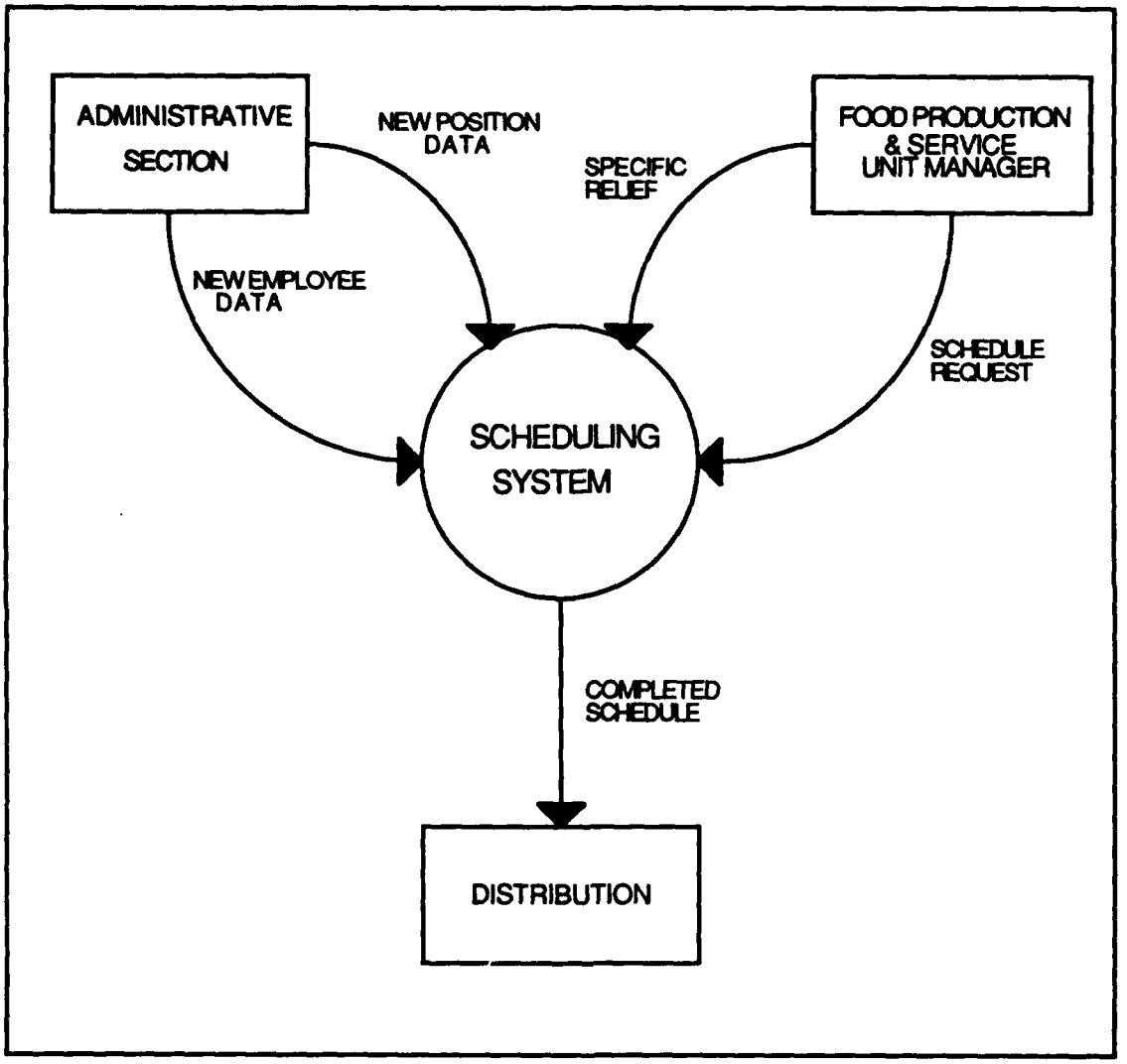


Figure (6) Scheduling System Context Diagram

intervention into the automated portion of the SCHEDULING SYSTEM that occurs when there is a desire to relieve an employee with a particular employee.

The SCHEDULING SYSTEM incorporates these data inputs into its internal data stores, activates its scheduling processes to manipulate the data, and outputs a COMPLETED SCHEDULE. This COMPLETED SCHEDULE is then available for DISTRIBUTION.

2. Scheduling System Level 1 Diagram.

Figure 7 is a lower level aggregation of all those activities, data flows, and data stores which make up the SCHEDULING SYSTEM activity depicted in the Context Diagram.

The UPDATE EMPLOYEE DATA activity receives NEW EMPLOYEE DATA and updates the EMPLOYEE data store. The UPDATE POSITION DATA activity receives NEW POSITION DATA and updates the POSITION data store. The SCHEDULE REQUEST triggers the GENERATE SCHEDULE activity which accesses the EMPLOYEE, POSITION, and SCHEDULE data stores. GENERATE SCHEDULE sorts and manipulates the available data, allows for scheduler manual insertion of SPECIFIC RELIEF data, creates a 14-day schedule, and writes this schedule to the SCHEDULE data store. The OUTPUT SCHEDULE activity reads the SCHEDULE data store and prints the selected schedule. (A more detailed explanation of this diagram is contained in the minispecifications of Appendix A and the data dictionary of Appendix B).

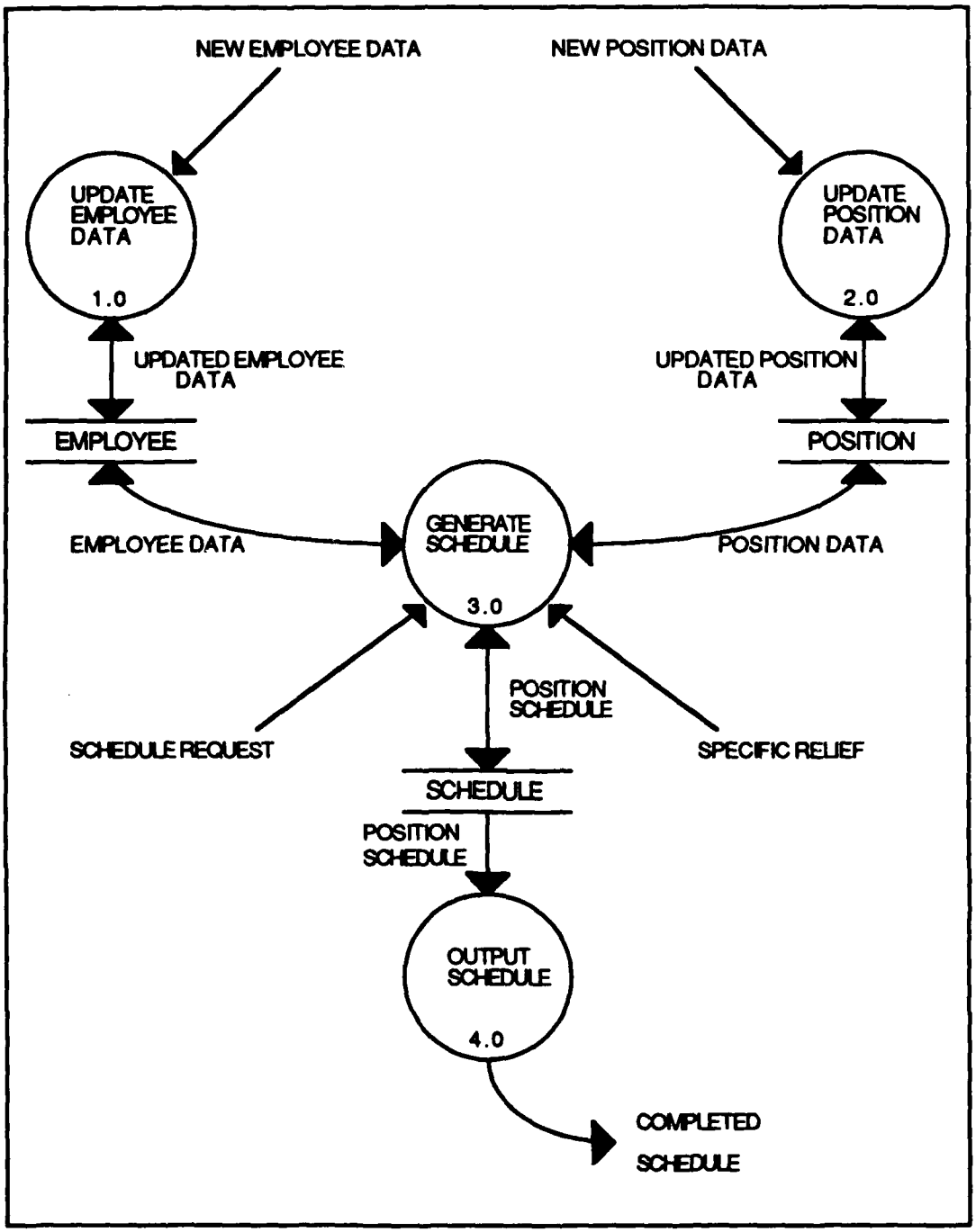


Figure (7) Scheduling System Level 1 Diagram

3. Scheduling System Level 2 Diagram.

Figure 8, is a breakdown of the activities contained within the GENERATE SCHEDULE activity shown in Figure 7. Upon receiving the SCHEDULE REQUEST input, SCHEDULE PRIMARY POSITION obtains PRIMARY POSITION DATA via the GET PRIMARY POSITION activity. It then obtains EMPLOYEE DATA from the GET AVAILABLE EMPLOYEE activity. The SCHEDULE PRIMARY POSITION then assigns an available employee to the selected primary position and writes this combination to the SCHEDULE data store. This process is repeated until all job positions are filled for a 14-day schedule.

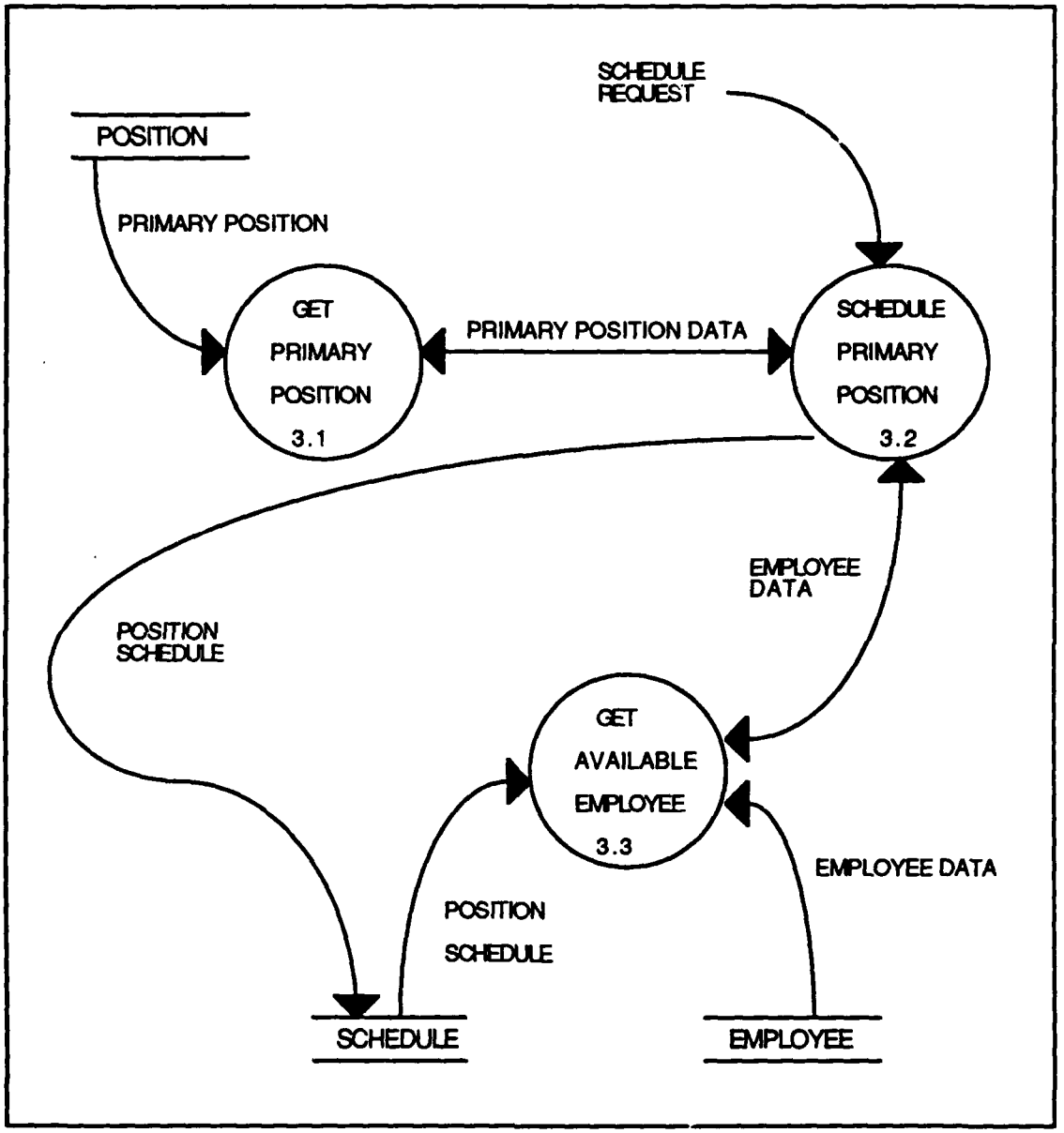


Figure (8) Scheduling System Level 2 Diagram (Generate Schedule)

III. PROTOTYPING

A. CLASSIC SYSTEM DEVELOPMENT LIFE CYCLE

After the decision was made to automate the employee scheduling process for the V.A. Medical Center, the next step was to decide upon a methodology for creating the computer program. Two systems development methodologies were explored. The first is the classic system development life cycle and the second is prototyping.

The classic system development life cycle, also known as the Waterfall Model, has been the standard software development methodology since the early 1970's. Although the model has phases that are known by several names, the basic steps are the same. Occasionally an author will either combine or sub-divide some of these phases. Upon closer inspection of the phase descriptions, one will discover that the processes are essentially the same. Roger Pressman [Ref.1] provides the following model of the classic system development life cycle (CSDLC) phases (see Figure 9):

1. System Engineering
2. Analysis
3. Design
4. Code
5. Testing
6. Maintenance

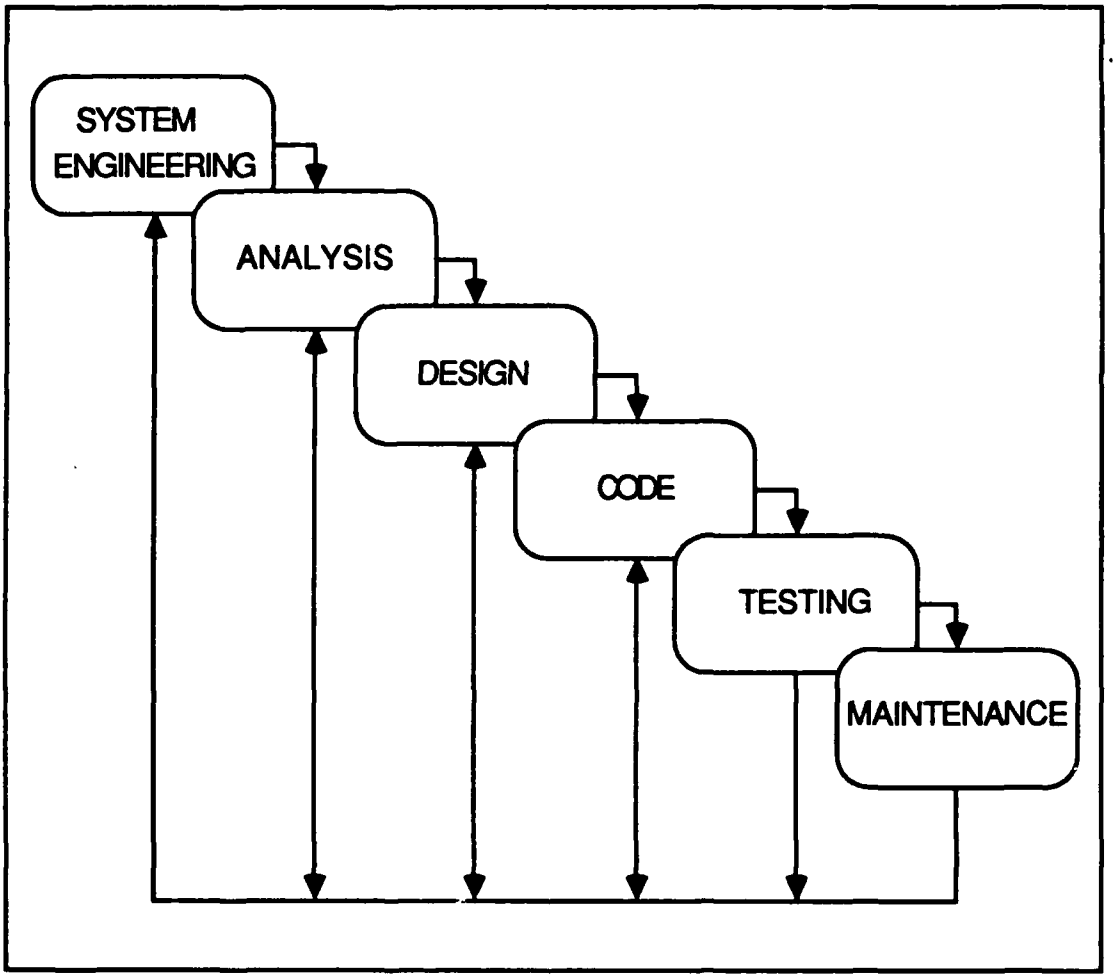


Figure (9) The Classic System Development Life Cycle

Looking at Figure 9, it is easy to see why this model is often called the "Waterfall Model"; each step leads, or "falls," directly onto the next. This is a rigid procedure that became popular because of the disciplined structure this methodology imposed upon the otherwise chaotic and undisciplined world of software development.

B. PROBLEMS WITH THE CLASSIC SOFTWARE DEVELOPMENT LIFE CYCLE

In the late 70's and early 80's software engineers began to question some aspects of the classic software development life cycle. In 1977 Joseph Podolsky [Ref. 2] was one of the first to write about the fact that the rigidity of CSDLC worked well in many cases; but was not flexible enough to fit situations where once the system development effort was started, the project requirements changed. Frequently the project requirements changed after the system was delivered and after the users were able to test the program and gain a feel for what it could do for them. Users often claimed that if they could have foreseen what the system's capabilities were, they could have indicated these unrealized needs at the beginning of the process.

The users depend upon the system designers to know how best to automate a procedure. After all, they are the computer experts, and the system designers must depend on the users to fully express their needs. Both parties are very much dependent upon the other. If the user is unable or

unwilling to express his requirements, then the system designer is handicapped in his ability to build a system that fully meets the user's needs. The same goes for a system designer who fails to adequately obtain all the required information prior to starting the system development process. Having knowledge of all user requirements at the beginning of the development process is an especially important issue when using the CSDLC. The rigidity imposed by the CSDLC does not easily allow for changes to be made after the system development process has begun. Making changes and correcting errors is a very expensive and time consuming process.

C. INTRODUCTION OF PROTOTYPING

It was becoming increasingly obvious to software engineers that the inability to accommodate changes within the software development process was causing severe delays and cost over-runs. These delays and over-runs were giving the software development industry a black eye within a rapidly advancing and highly competitive computer industry.

Podolsky [Ref. 2] recommended that the CSDLC be modified so that the development process be built around the expectation that system requirements will change and probably change often. He called his answer to this problem the "Recursive Development Cycle" which is the basic idea behind what is today called prototyping.

Podolsky stated:

> The user managers will be able to see how the system is working in their area. Then when they come up with suggestions, we will be expecting them and be ready and eager to put their ideas into the next iteration--instead of getting mad because they didn't suggest them during the design process.

The prototyping methodology began to gain a large and enthusiastic following in the early 80's and appears to be still gaining in popularity even today. In one of the classic articles about prototyping, Naumann and Jenkins in 1982 [Ref. 3] stated:

> Prototyping represents and parallels the dynamic process of growth, change, and the evolution existing in any living system. It neither requires nor permits prolonged static specifications in development projects. Since any "freeze points" in the prototype design process are of only a very limited duration, prototyping accommodates changes in both the user and systems environments.

Pressman [Ref. 1] describes the prototyping methodology as shown in Figure 10. Prototyping is a system development process that allows for and expects many "iterations" throughout the entire development cycle. At any time the system may be sent back for redesign, modification or to be completely redone from the beginning.

Prototyping is a software development methodology that can be a very useful alternative to the classic software development life cycle. It is most useful in situations that have the following characteristics:

1. Where user needs and requirements can't be expressed fully or clearly.

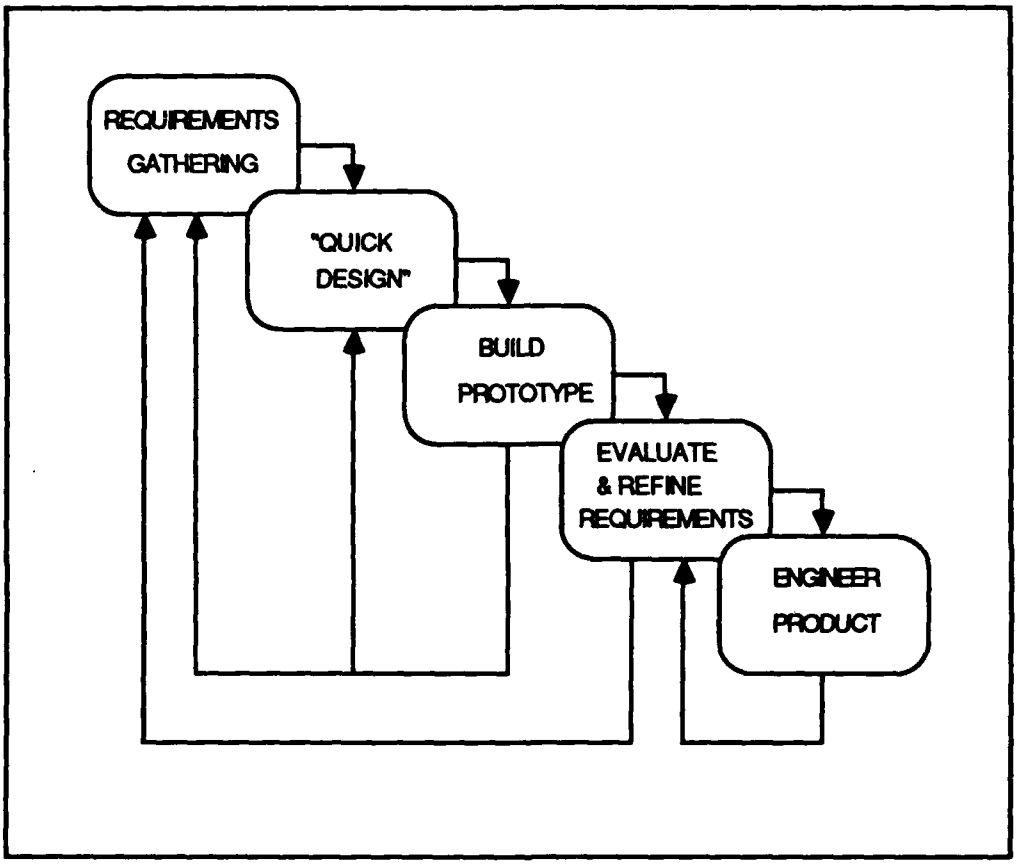


Figure (10) Prototyping

2. Where the analysts lack experience in this area and will need to test the feasibility of a particular design approach.

3. Where the system will be high risk and high cost.

4. Where there is a high probability that future versions of the system will require modification.

Sprague and McNurlin [Ref. 4], give the following description of what prototyping is and what it does:

1. A prototype is a live working system: it is not just an idea on paper.

2. The prototype may or may not become the actual production system.

3. A prototype's purpose is to test out assumptions about users requirements and/or system design architecture, and/or even the logic of the program.

4. It is created very quickly--often within hours, days or weeks--rather than months or years.

5. The prototype is relatively inexpensive to build.

6. Prototyping is an iterative process.

D. USING PROTOTYPING TO DEVELOP THE AUTOMATED SCHEDULING SYSTEM

At the start of this project several factors were felt to be very important as to how the overall project should be developed. Upon close examination of these factors it became clear that prototyping would be very advantageous to this particular situation and would greatly aid in the successful completion of the project.

The first factor was the ability of the users to provide adequate knowledge of the present system during the systems

analysis phase of the project. The Chief Dietician was the individual who was the driving force behind automating the manual scheduling system, but at the same time she was not able to provide answers as to how the system currently works and what capabilities that an automated system should provide. For this information we were directed to the individual we call the Primary Scheduler.

The Primary Scheduler currently creates the manual schedule and would be responsible for overseeing the use of the automated system when it was created. This individual, during preliminary interviews, proved to be somewhat reluctant to provide answers to questions about the system. He stated:

> I don't know anything about computers. If something needs to be done on a computer, I tell someone else to do it.

Because of the user's reluctance to be an active participant in the project, it was never quit clear if all the required information for designing and building the automated system had been obtained. It was often necessary to stumble along the development path until a point was reached that showed the need for further amplifying information. At this point the Primary Scheduler would have to be interviewed again.

The Primary Scheduler also wanted to start a new system of rotating primary employees and asked that this capability be included in the program. Yet he had only a vague idea

about how the procedure should work and was unsure exactly how to go about implementing it.

Because of the lack of definite guidance in the above areas, it would have been extremely difficult, if not impossible, to conduct this project in a timely and efficient manner using the classic system development life cycle. Maryam Alavi [Ref. 5] states:

> Prototyping seems to be effective in coping with undecided users and clarifying "fuzzy" requirements.

Alavi also stated:

> Prototyping is a practical way to cultivate and achieve user participation and commitment to a project.... At the end of the prototyping process the users were very satisfied with the development effort and the prototype. They felt they had some real influence in the design process.

Another factor was the short amount of time that was going to be available for designing and building the system. Approximately six months were scheduled to be available for the entire thesis. During this six month period, the user's current system would have to be analyzed, an automated system designed and coded, the new automated system would have to be installed and tested, and the thesis would have to be written. System development, therefore, would have to be done quickly. By using micro-computers and advanced programming languages with built in data management and screen generator capabilities, it was felt that the system

could be built within the three or four months that were to be available for system production.

The last factor was the level of experience the builders had in systems development and in using the available high level programming languages. Prior to commencing this project, neither of the system developers had any practical knowledge of real world system development other than what was taught in the classroom. This fact along with very limited experience with the available programming languages indicated there was going to be a great deal of trial and error involved with both the system analysis and the actual program construction.

The use of prototyping to develop the scheduling system will help to resolve many of the problems identified above. Prototyping allows inexperienced designers/programmers to develop quick and inexpensive iterations of the program as they increase their knowledge level of the programming language and system analysis process. It also provides the flexibility of allowing for relatively painless modifications to the prototypes as new and different user requirements are realized.

E. PROTOTYPING TOOLS

In the 1980's the computer industry began to see rapid growth in the areas of micro-computers, advanced programming languages, and software development tools. These

technological advances tied in nicely with prototyping as they provided the means by which prototypes could be developed to test assumptions in a relatively inexpensive and rapid manner. The language chosen should also include many of the new user-friendly innovations such as data base management, screen generators, menu builders, etc. McClure [Ref. 6] states:

> Screen generators, report generators, and menu builders are used mainly to prototype the user interface in a quick, friendly way of clarifying user requirements. The prototype provides users with a concrete model of how the system will look from the users' perspective. This is an effective method for identifying and correcting misunderstandings about user expectations of the system.

The need for data base management capabilities is expressed by Naumann and Jenkins [Ref. 3]:

> Database management systems are central to prototyping in two ways. Prototyping without use of database management software prohibits the design and programming of data handling facilities in the desired time frame. In addition, many of the other resources needed for prototyping are being developed as extensions to database management systems.

These above capabilities should all be transparent to the user so as to provide a quick and friendly program interface that doesn't require spending long and tedious amounts of time learning the inner workings of the programming language. This ease of use is especially critical in providing the user a prototype that he can use immediately to see the direction of the project.

Because of the requirement for the prototype program to be developed as rapidly as possible, it was decided that a high level language incorporating as many of the previously discussed features as possible would be chosen. It was also apparent that there would not be time to learn a language from scratch; therefore, whichever language was chosen would require the designers to have at least a small degree of familiarity with it. This last aspect caused virtually all fourth generation languages such as Prolog or C, to be discarded from consideration early on. The only high level languages that fit the project needs and the designers had some experience with were LOTUS 123 and DBASE III PLUS. Deciding which particular program would be best was difficult as both possessed qualities that at the beginning appeared to be well suited for this project.

LOTUS 123 has a limited database management capability, but is relatively user-friendly and has good report and menu generators. DBASE III PLUS has a very strong database management capability, with good screen and menu generators. DBASE III PLUS is not quite as user-friendly when interacting directly with its command language, but the program may be compiled into an easy to use, stand alone (executable) program.

F. THE SYSTEM DEVELOPMENT PROCESS

It was decided that each designer would build a prototype, one using LOTUS 123 and the other using DBASE III PLUS. The prototypes would be built as rapidly as possible, hopefully within a period of a few weeks, then an evaluation would be made to determine which of the prototypes was best suited for implementation.

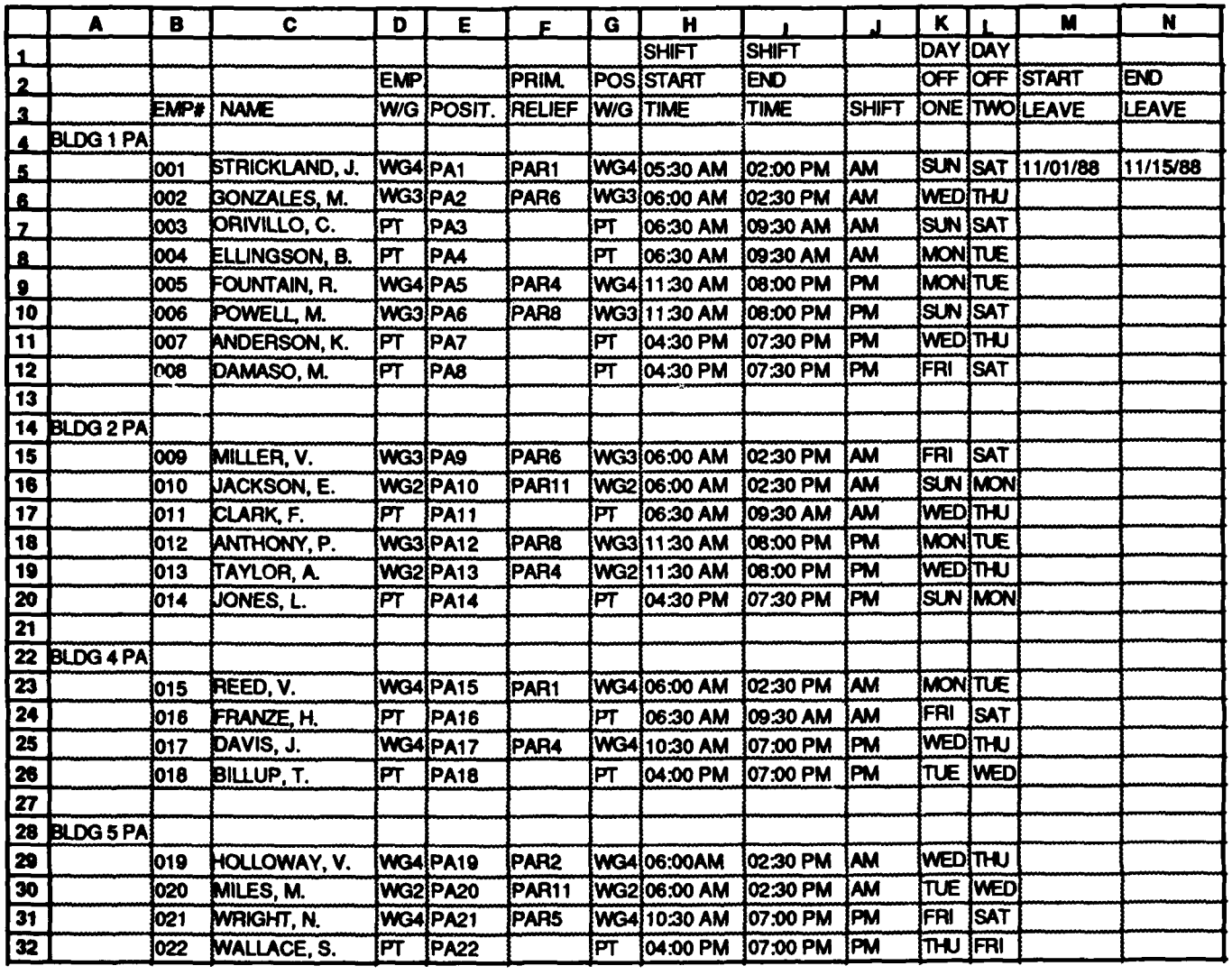
Due to the lack of "in depth" experience with the chosen software packages, it soon became obvious that the prototypes would take somewhat longer than the initially expected few weeks. This delay was further aggravated because as the programs grew, holes in the original system analysis information became apparent. When these gaps in required system knowledge surfaced, further meetings with the Primary Scheduler had to be arranged. Throughout this process the Primary Scheduler never appeared to be totally comfortable with the project and information had to be pulled out rather than coming out in a free exchange of ideas.

After approximately two months of programming effort, the two prototypes were at a point where a decision could be made concerning which program was to be continued and which was to be discarded. The DBASE III PLUS prototype appeared to be very flexible and would be able to produce the desired schedule in under 30 minutes. The LOTUS 123 prototype, on the other hand, was very rigid and would not allow for future changes without major reprogramming. The real death blow to

the LOTUS 123 prototype was that it was going to require somewhere in the range of five to six hours for the production of the schedule when run on an Intel 8088 micro-processor based computer (IBM XT type). This time could be reduced some by using an 80286 based machine (IBM AT type), but not significantly enough to make the LOTUS prototype competitive with the DBASE prototype.

The LOTUS 123 program was based around a spreadsheet that contained all the job, employee and schedule form data in a pre-set format. Figure 11, is an abbreviated sample of this spreadsheet. The spread sheet acts as the database from which the program would extract, manipulate, and input data. The program's rigidity came from the spreadsheet format. This format had to be set up before hand, and had to have all the data current in the proper cells prior to the running of the program. If any changes to the format were to be made, such as the addition, changing, or deletion of jobs, the format had to be changed in the spreadsheet as well as reprogrammed to reflect those changes. Any future changes would require someone familiar with programming in LOTUS 123 to make the required coding modifications. This would not be a realistic requirement to place on any future users of the program.

LOTUS 123's built in database commands are so limited in scope and capability that they were not able to fully execute all the database management functions that were



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1								SHIFT	SHIFT		DAY	DAY		
2				EMP		PRIM.	POS	START	END		OFF	OFF	START	END
3		EMP#	NAME	W/G	POSIT.	RELIEF	W/G	TIME	TIME	SHIFT	ONE	TWO	LEAVE	LEAVE
4	BLDG 1 PA													
5		001	STRICKLAND, J.	WG4	PA1	PAR1	WG4	05:30 AM	02:00 PM	AM	SUN	SAT	11/01/88	11/15/88
6		002	GONZALES, M.	WG3	PA2	PAR6	WG3	06:00 AM	02:30 PM	AM	WED	THU		
7		003	ORIVILLO, C.	PT	PA3		PT	06:30 AM	09:30 AM	AM	SUN	SAT		
8		004	ELLINGSON, B.	PT	PA4		PT	06:30 AM	09:30 AM	AM	MON	TUE		
9		005	FOUNTAIN, R.	WG4	PA5	PAR4	WG4	11:30 AM	08:00 PM	PM	MON	TUE		
10		006	POWELL, M.	WG3	PA6	PAR8	WG3	11:30 AM	08:00 PM	PM	SUN	SAT		
11		007	ANDERSON, K.	PT	PA7		PT	04:30 PM	07:30 PM	PM	WED	THU		
12		008	DAMASO, M.	PT	PA8		PT	04:30 PM	07:30 PM	PM	FRI	SAT		
13														
14	BLDG 2 PA													
15		009	MILLER, V.	WG3	PA9	PAR6	WG3	06:00 AM	02:30 PM	AM	FRI	SAT		
16		010	JACKSON, E.	WG2	PA10	PAR11	WG2	06:00 AM	02:30 PM	AM	SUN	MON		
17		011	CLARK, F.	PT	PA11		PT	06:30 AM	09:30 AM	AM	WED	THU		
18		012	ANTHONY, P.	WG3	PA12	PAR8	WG3	11:30 AM	08:00 PM	PM	MON	TUE		
19		013	TAYLOR, A.	WG2	PA13	PAR4	WG2	11:30 AM	08:00 PM	PM	WED	THU		
20		014	JONES, L.	PT	PA14		PT	04:30 PM	07:30 PM	PM	SUN	MON		
21														
22	BLDG 4 PA													
23		015	REED, V.	WG4	PA15	PAR1	WG4	06:00 AM	02:30 PM	AM	MON	TUE		
24		016	FRANZE, H.	PT	PA16		PT	06:30 AM	09:30 AM	AM	FRI	SAT		
25		017	DAVIS, J.	WG4	PA17	PAR4	WG4	10:30 AM	07:00 PM	PM	WED	THU		
26		018	BILLUP, T.	PT	PA18		PT	04:00 PM	07:00 PM	PM	TUE	WED		
27														
28	BLDG 5 PA													
29		019	HOLLOWAY, V.	WG4	PA19	PAR2	WG4	06:00AM	02:30 PM	AM	WED	THU		
30		020	MILES, M.	WG2	PA20	PAR11	WG2	06:00 AM	02:30 PM	AM	TUE	WED		
31		021	WRIGHT, N.	WG4	PA21	PAR5	WG4	10:30 AM	07:00 PM	PM	FRI	SAT		
32		022	WALLACE, S.	PT	PA22		PT	04:00 PM	07:00 PM	PM	THU	FRI		

Figure (11) Sample Lotus 123 Spreadsheet Database

required of the program. These functions, therefore, had to be performed by a programmer-designed sequence that searched the applicable areas of the spreadsheet cell by cell for the needed information. This code directed positioning, examination of cell contents, and repositioning of the cursor throughout a large spreadsheet is a very slow and time consuming process. The result of which was a program that required an excessive amount of time to produce the desired schedule.

G. PROTOTYPING CONCLUSION

Because of the very long running time, lack of flexibility, and the requirement for the user to become involved with the internal programming, the LOTUS 123 program was dropped. It was decided that the DBASE III PLUS prototype would be continued. The details of the DBASE III PLUS program will be discussed further in the following chapter.

One final benefit of using prototyping was the opportunity that was provided to experiment with two different programming languages. The developers were able to select the best of the two prototypes and discard the other without sacrificing extreme amounts of programming effort. This would certainly not be economically feasible with the classic system development life cycle.

IV. SYSTEM DESIGN AND ARCHITECTURE

A. DESIGN CRITERIA FOR THE AUTOMATED SCHEDULING SYSTEM

Designing the automated Scheduling System to operate within the micro-computer environment will make it possible for an inexperienced scheduler to produce a completed schedule in under 30 minutes. In addition, by structuring the automated system within the defined constants, a framework will be provided which, when operated by different schedulers, will provide consistency in job scheduling and job rotation.

Those decision criteria of the Primary Scheduler that are evaluated as being beneficial to the current system will be captured and implemented in the automated system. This modeling and automation of the Primary Scheduler's scheduling practices results in a program that is familiar in look and feel to the normal user. When this familiarity is captured by the system's method of interaction with the user, a friendly interface is created.

B. APPLICATION LANGUAGE

A relational database application language (DBASE III PLUS) was selected for this prototype for the following reasons:

1. Its ability to establish and maintain relationships between files.
2. The ability of its utility functions to easily recognize and manipulate date and day-of-the-week

variables. For example, given any date input DBASE III *PLUS* can identify on which day of the week it will occur.

3. The developer's working knowledge of the command language which would greatly reduced the programming learning curve.

A database language compiler, Clipper, was used to eliminate the requirement of the user having to run the scheduling program within the database application language, as well as to greatly reduce the speed of execution during schedule generation. A compiler program can be used to separate the database source program from the application language and convert it into an executable program. An executable program avoids having to rely upon the parent application language for the execution of its command instructions and can interface directly with the computer processor. This reduces the overall memory requirements and significantly speeds up the program execution.

C. GENERAL DESIGN

The automated scheduling system will be able to accept additions and deletions of specific data elements of the files used in the scheduling system (see Figure 12), and it will access this updated data during schedule generation.

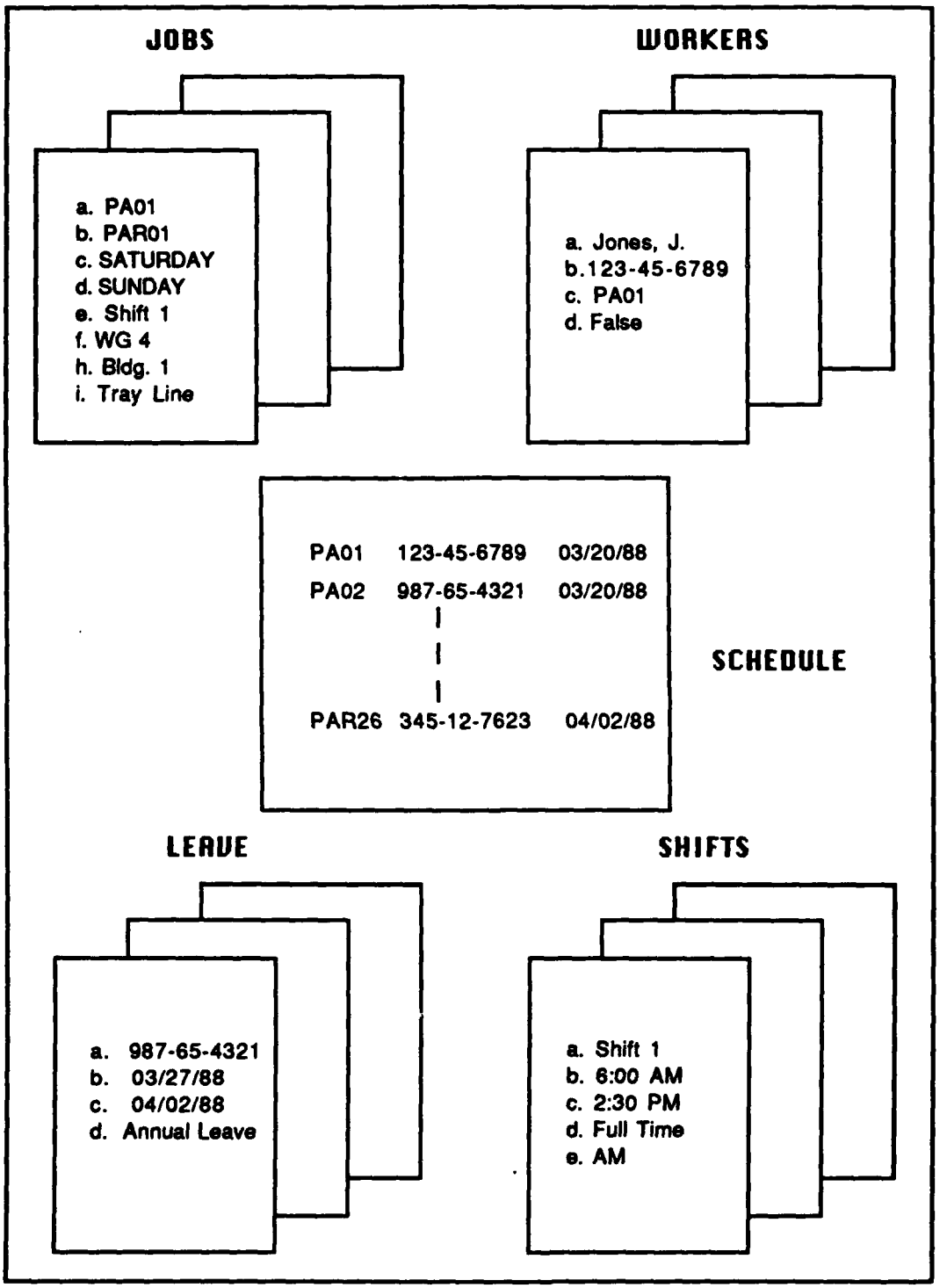


Figure (12) Program Data Files

The design of the days-off framework and the method in which days-off are rotated prevents any employee from exceeding the six consecutive work day limit. For example, an employee that has Tuesday/Wednesday off during the current schedule period will have Monday/Tuesday off during the next schedule period. This allows five consecutive work days during the current schedule and four consecutive work days during the transition to the new schedule. By building this constraint into the days-off framework, the program is able to avoid what would be a time consuming task of checking each employee's schedule so as not to exceed the consecutive work day constraint.

A manual intervention to the automatic scheduling process will be offered in those cases where an exact fit of relief employee to primary position requirements is not available. (This situation may occur due to over scheduling of leave for employees of like skills or due to their unanticipated absences.) This option of overriding the automated scheduling algorithm is provided in order that a scheduler, with personal knowledge of a particular employee's experience, history, and performance level may make a value judgement in that particular job assignment. The Scheduling System will notify the user of this "no-match" situation by displaying the screen in Figure 13, and will offer the scheduler the options of:

1. Manually assigning a relief selected from the displayed list of all available relief employees.

2. Having the system flag this position as having no relief and depict this on the printed output with the letters 'NR'.

```
There are no normal reliefs to fill the following
position:

    PA52   WG4   03/26/88

-------------------------------------------------
The following relief employees are available:


Position       Name           Wage Grade

PAR07          PETERS, T.          3
PAR17          SMITH, J.           2
PAR20          JONES, R.           3


Enter the POSITION of selected relief:   PAR07
(or press ESC to NOT schedule a relief for this
position)
```

Figure (13) *No-Match Relief Screen*

The scheduler will be allowed to manually intervene in those cases where he/she desires to assign a specific alternate employee, other than the designated relief, to a primary position whose normally assigned employee is scheduled for a leave period. In these cases, because the relief employee assumes the days-off of the primary position, the normal safety check of the days-off framework is circumvented. Therefore, the consecutive work day constraint may be violated and the relief employee's schedule must be checked. If this check detects too many consecutive

work days, the scheduling system will display the relief employee's 14-day schedule and offer the scheduler the option of adjusting days-off. Once again, user involvement and experience will better dictate these adjustments.

D. PRIMARY FILE STRUCTURE

There are five primary files from which the schedule is created (Figure 12):

1. Each job position in the JOBS file is defined by:

   a. primary job position
   b. designated relief position
   c. first day off during week
   d. second day off during week
   e. wage grade of job
   f. building where job is located
   g. work area within the building

2. Each employee in the WORKERS file has the following information stored:

   a. employee name
   b. employee social security number
   c. job position assignment
   d. logical flag used to determine if employee can be rotated from the currently assigned position

3. The LEAVE file is defined by:

   a. employee social security number
   b. leave starting date
   c. leave ending date
   d. reason for taking leave

4. The SHIFTS file is defined by:

   a. identifying number
   b. starting time of shift
   c. ending time of shift
   d. shift job status (full-time (FT) or part-time (PT))
   e. shift time status (early (AM) or late (PM))

5. The SCHEDULE file is defined by:

   a. job position identifier
   b. scheduled employee's social security number
   c. schedule date

During the scheduling process, temporary files are created from the main WORKERS and SCHEDULE files and used during all schedule manipulations. Any changes made during the creation of a new schedule, such as initiating a job rotation or introducing a specific relief, will only be transferred to the permanent files with the acknowledgement of the user.

E. DEFINITIONS

The following terms are used extensively throughout the accompanying sections and require amplification.

1. Job Status

A particular job position's status is identified as being either full-time or part-time. This information is stored in the SHIFTS file. Job Status is used in the scheduling program when determining relief qualifications.

2. Job Requirements

The requirements of a particular job position are defined by the facility location (Palo Alto or Menlo Park), shift time status, job status, and wage grade.

3. Relief "Daisy-chain"

The analysis of the current scheduling system determined that two groups of job positions existed: the Primary Job Positions and the Relief Job Positions. Each

primary position has a designated relief position (see Figure 12) which is the first choice when a relief is required. The next choice would be any of those relief job positions which match the primary job requirements. This grouping of relief positions is created by linking together all relief positions with matching job requirements via the designated relief attribute in the JOBS file. The scheduling program establishes this link during the initial creation of the JOBS file and maintains its integrity during subsequent addition and deletions of relief job positions. The designated relief of the primary job position is the entry point into this "Daisy-chain."

4. Specific Relief

The user has identified an occasional need to relieve a primary job position with a relief other than the designated relief and to have this relief assume the days-off of the primary position. This relief is called a Specific Relief.

5. Job Rotation

Job Rotation is the moving of employees among jobs with matching job requirements. There are some employees who are not to be moved, hence the need for the logical field in the WORKERS file.

F. SCHEDULING PROGRAM SEQUENCE

The following sequence of events occurs when a new schedule is created:

1. When the Scheduling Program is first brought on-line, its days-off framework must be matched with the current system. This is done by a menu-selected initialization process which matches the correct days off sequence with a user inserted schedule date and also establishes a reference date which will be used in the automatic days-off rotation in subsequent schedules. Figure 14, represents the menu screen from which the user selects option 2 to begin the initialization process.

   In any given schedule the relationship between the days-off for different jobs remains constant. That is, if PA01 has Monday/Tuesday off and PA03 has Thursday/Friday off, for any new schedule in which PA01 again has Monday/Tuesday off, PA03 will have Thursday/Friday off. Therefore, by matching any given job position's days-off to the scheduler's desired sequence, all other job positions will match.

   In Figure 15, the user has elected to match the days-off framework to the schedule date of March 20, 1988. Menlo Park job position one (MP01) shows a days-off sequence of Saturday/Sunday. If this is not correct, the user can sequence through all the days-off combinations by continuing to answer 'N' to the correct days-off question. Once the correct sequence is attained, the user answers 'Y', the input date is established as the reference point for future days-off rotations, and the user is returned to the main menu of Figure 14.

2. To begin the scheduling process the user inserts a schedule start date. From this date the program will correctly sequence the days-off in the JOBS file based upon the initialized reference date.

3. The user is queried for Carryover Specific Reliefs (Figure 16). This function only applies if a Specific Relief has been assigned from a previous schedule and if the relieving period has extended into the current schedule.

4. The user is queried if Job Rotation is desired for the current schedule and for any employees who are not to

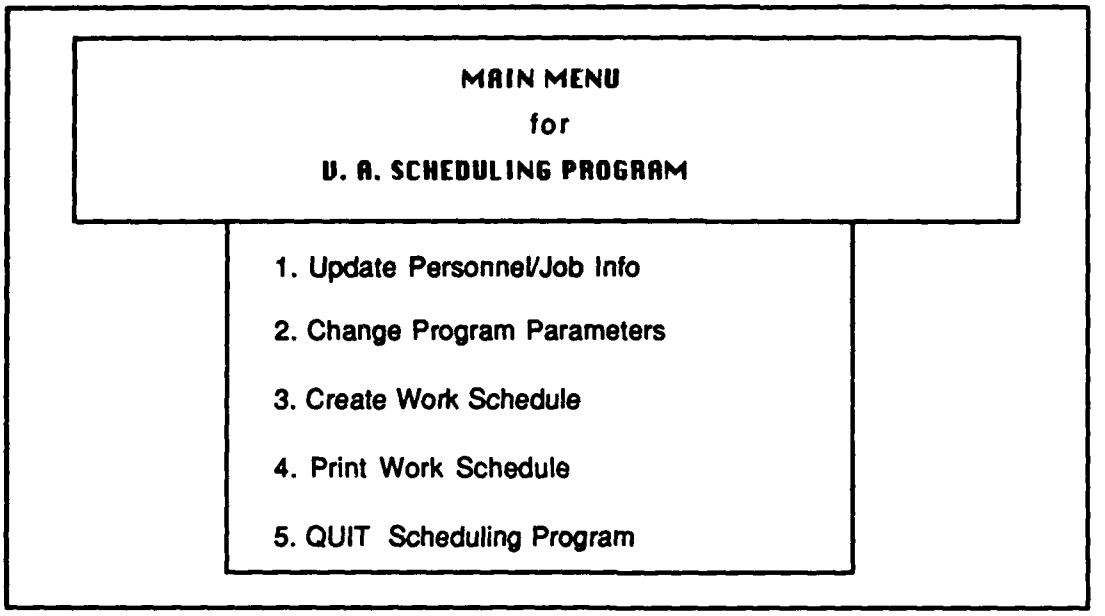


Figure (14) Opening Menu

This part of the program can be used as follows:

1. To change how often days off are rotated.
2. To change the days off associated with a reference date which is used by the program to automatically rotate the days off.

Enter the date the schedule is to start (MM/DD/YY): 03/20/88
(or press ESC to exit)

MPO1: SATURDAY SUNDAY

Are these the correct days off? (Y/N): N

Figure (15) Parameter Change Screen

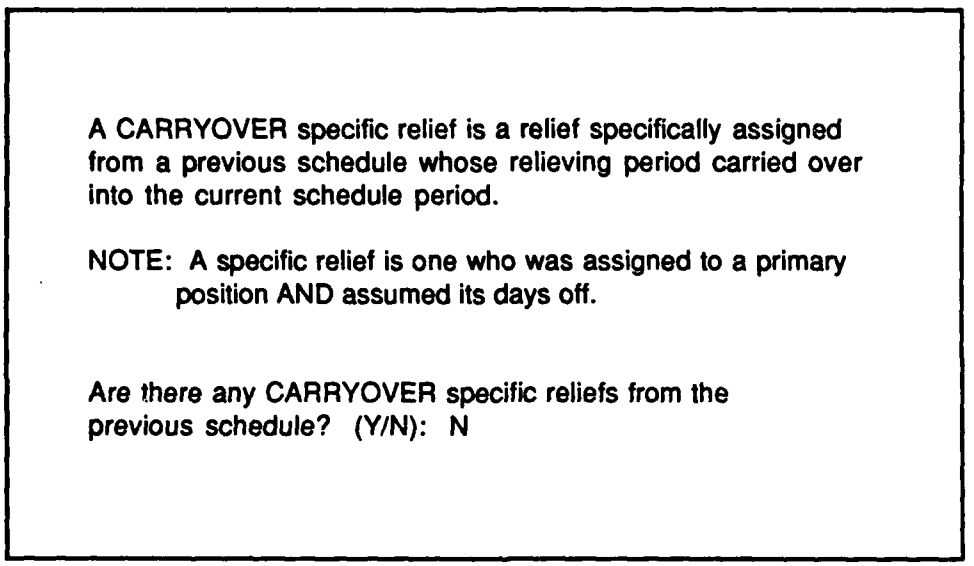


Figure (16) Carryover Specific Relief Screen

be rotated (Figure 17). This portion of the scheduling system is menu-driven, displaying the selected job groupings as well as displaying those employees who are not to rotate.

5. The user is queried for Specific Reliefs for this schedule period (Figure 18). These will only be any new specific reliefs that start during the current schedule.

6. After the above mentioned information has been gathered, the Scheduling Program creates the new schedule using the following algorithm:

   For each Schedule Date:

   For each Primary Job Position that has an employee assigned:

   If Schedule Date is a normal day off or employee assigned is on leave:

   Check availability of employees in the Relief "Daisy-chain" starting with the Designated Relief.

   If available relief employee is found, assign him/her to fill the Primary Job Position on the Schedule Date in the Schedule file.

   Else list all available relief employees whose assigned positions match the job requirements of the Primary Job Position excluding the wage grade restriction. (i.e., Include all wage grades). User selects desired relief.

   If no relief employees are available, assign the letters 'NR' (No Relief) to the Primary Job Position on the Schedule Date in the Schedule file. (See Figure 19, PA50's schedule for an example of this).

7. The completed schedule is kept in a temporary file from which the user can use the screen shown in Figure 19 to view it, delete it, or save it.

8. If a printed copy is desired, the user must first save the newly created schedule. The schedule output format is in sorted order by position identifier, building number, and work area. For each job position, the schedule will show the position identifier, employee name, shift times, and the daily assignments

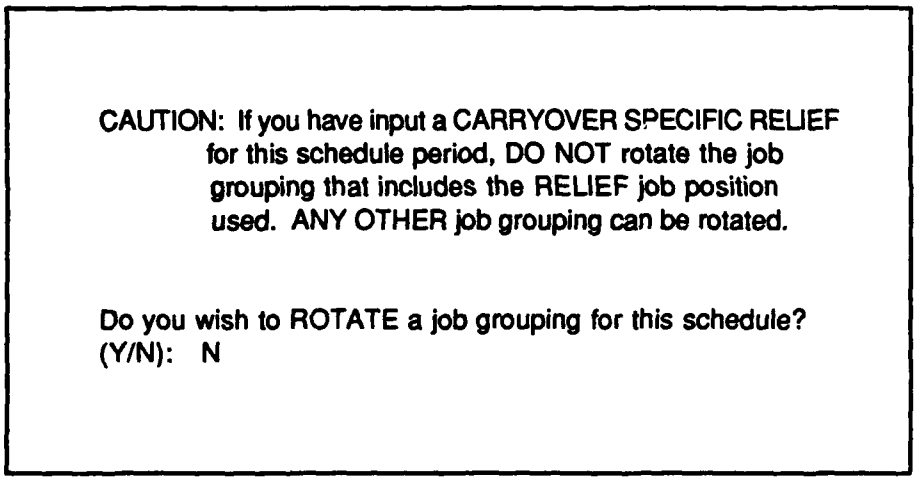


Figure (17) Job Rotation Screen

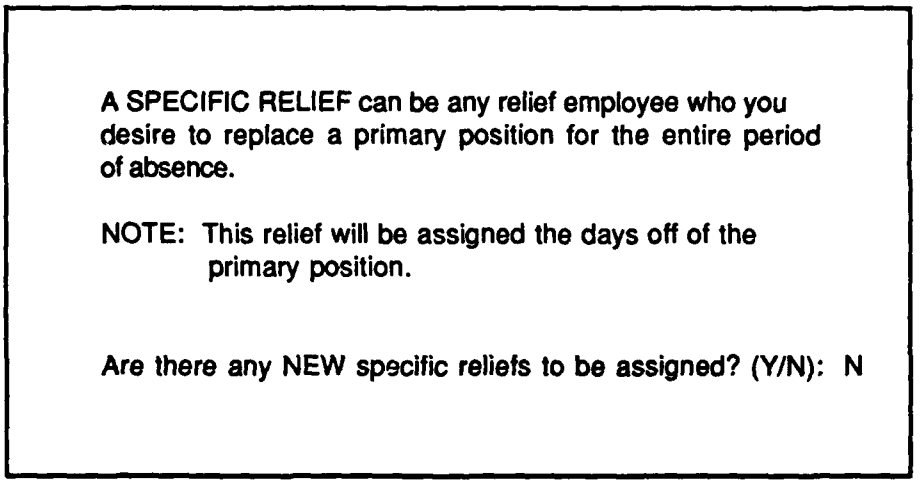


Figure (18) New Specific Relief Screen

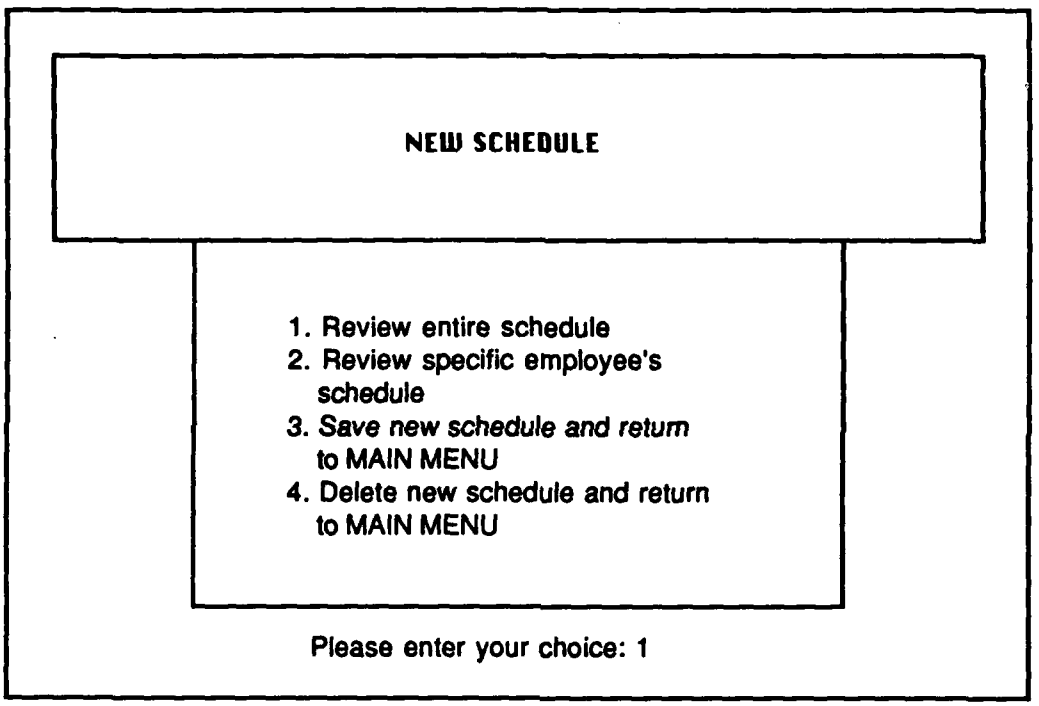


Figure (19) Review Created Schedule

for a two week schedule. A sample seven day portion of a completed schedule is depicted in Figure 20.

PALO ALTO SCHEDULE

Schedule dates: 03/20/88 to 03/26/88

Posit	Name	Time	20 Sun	21 Mon	22 Tue	23 Wed	24 Thu	25 Fri	26 Sat
Building 23 TRAYLINE									
PA46	JONES, T.	5:00PM-8:00PM	OFF						OFF
PA47	THOMAS, J.	5:00PM-8:00PM				OFF	OFF		
PA48	ANDERSON, A.	6:00AM-2:30PM			OFF	OFF			
PA49	JONES, A.	6:00AM-2:30PM					OFF	OFF	
Building 32 COLD FOOD DISHUP									
PA50	SMITH,J.	11:30AM-8:00PM	NR						OFF
PA51	DOUGLAS, B.	8:00AM-4:30PM					OFF	OFF	
PA52	STEVENS, S.	8:00AM-4:30PM	OFF						OFF
RELIEFS PALO ALTO DIVISION									
PAR01	ABORN, F.	6:00AM-2:30PM	01	15	15	OFF	OFF	51	01
PAR02	BROWN, J.	6:00AM-2:30PM	52	OFF	OFF	19	19	25	25
PAR03	BRAY, J.	6:00AM-2:30PM	27	27	28	28	51	OFF	OFF
PAR04	CALLEGO, T.	11:30AM-8:00PM	OFF	05	05	17	17		OFF
PAR05	CAUSEY, T.	11:30AM-8:00PM	26	26		OFF	OFF	21	21

Figure (20) Sample Program Generated Schedule

V. SUMMARY

A. EVALUATION OF THE IMPLEMENTED SYSTEM

During the development and implementation of the Scheduling System program, several important factors concerning the program were realized.

1. As designers of the system and being intimately involved on a day to day basis, it is often difficult to objectively evaluate what has been created. Thus, the development team relies heavily upon user feedback once the system has been implemented, tested, and used over a period of time. In this particular situation, the Primary Scheduler is perfectly happy and comfortable with his manual scheduling process and does not appear to be highly motivated towards adopting the automated system. Therefore, the automated system is not being used to the extent that will generate feedback for the development team. It would appear that until pressure is applied by the Chief Dietician to force its use, the automated system will not be fully implemented. This is a problem for two reasons:

   a. The system requires periodic updating of the databases to ensure maximum efficiency and active user involvement for maximum proficiency. By avoiding use of the automated system until the Primary Scheduler is unavailable will require a great deal of initial set up effort which may lead to user frustration and dissatisfaction with the system.

   b. The development team will be unable or unavailable to correct those deeply imbedded problems that will only be discovered from extensive program use.

2. The time required to create a 14-day work schedule by an inexperienced scheduler (upwards of 15 hours) was the driving force behind the development of the Scheduling System program. The final version of the Automated Scheduling System will enable an inexperienced user to create a complete schedule in under 30 minutes. In benchmark tests the Scheduling System produced an acceptable schedule in 14 minutes, thereby showing the success of the project with respect to this criteria.

3. A secondary concern of the user organization was the dependence upon a single individual, the Primary Scheduler, for the production of a workable schedule. The Scheduling System as implemented is an easy to use ("User-Friendly"), menu-driven program that an inexperienced user can operate with a minimum of supervisory input. Preliminary testing has shown that inexperienced users are able to produce a workable schedule, thus relieving the dependence upon the single individual within the organization.

4. During initial system analysis the Primary Scheduler voiced a desire to implement a method of rotating employees through different jobs, but was unable to devise a manual system for doing this in a consistent and fair manner. This employee job rotation was designed into the Scheduling System and the initial user feedback has been positive.

B. LESSONS LEARNED

The value gained from the analysis, design, and implementation of any software system is not limited to the final user. The development team gains valuable experience throughout the entire process, especially in those areas which do not go smoothly. From these experiences the developers are able to devise better strategies for future projects. The important lessons learned from this project are:

1. Prototyping Lessons

The selection of prototyping to develop, build, and implement the Scheduling System was done for several reasons as discussed in Chapter III. The primary reason for using prototyping was to identify user requirements that the Primary Scheduler was unable or unwilling to share with the design team. As the coding of the program progressed, those

user requirements that were unclear became apparent and were resolved by repeat interviews. This process resulted in a better understanding of the requirements, but did little in the way of improving the Primary Scheduler's acceptance of the automated system.

After the program was completed and implemented, the Primary Scheduler examined the results and suddenly became very relaxed and open about the situation. The Primary Scheduler stated that the automated system validated his method of manually developing a schedule. It suddenly became apparent to the design team that he had been afraid that the purpose of the project had been to prove his method wrong. Once it became clear that this was not the case, he became interested and quizzical about the program.

If earlier prototypes, as is recommended in the literature, had been developed and shown to the Primary Scheduler, perhaps his acceptance of the system could have been cultivated at an earlier stage. This could have reduced the number of interviews required to extract all the required information from him. Earlier acceptance of the program may also have increased the likelihood of the program being more actively used in its final version.

Programming and system design inexperience on the part of the design team were the main reason that earlier versions of the program were not available for user testing. By the time the developers had a firm grip on what the system

requirements were and had developed the expertise to implement them, two months had elapsed. At this point the initial prototype had evolved through several iterations before being shown to the users. If more design and programming experience had been initially available, the prototyping methodology could have been followed more closely.

2. Specific Reliefs

The single most troublesome area that arose during the actual coding of the scheduling program was in dealing with the Specific Reliefs. Under normal conditions a relief employee's days-off are set for a given schedule period and that employee can only be used for relief on his/her scheduled work days.

However, the user defined a specific requirement where, on occasion, the selected relief would assume the days-off sequence of the primary position. By invoking this procedure, the safety check for consecutive work days built into the days-off framework is bypassed. The scheduling program must then determine when a selected relief employee last had a non-work day, determine when the next non-work day will occur, and make the appropriate schedule changes if the consecutive work day constraint is exceeded. The algorithm to accurately accomplish this procedure required approximately 40% of the programming time, yet will be used in less than 5% of an entire year's schedules.

It is recommended that the manual intervention option be dropped for specific reliefs and allow the automated system to determine reliefs. This will not create any problems with generating an acceptable schedule and will allow the system to function in a more effective and time efficient manner.

3. Improving the Program's Efficiency

Programs built using relational database languages, such as DBASE III PLUS, have the inherent problem of requiring access to a large number of files (Input/Output). When these accesses are made to disk memory, the time required for the program to run is greatly increased. Any techniques of moving the location of these files to faster disk memory (i.e., hard disks) or to Random Access Memory for data access has a significant improvement in the time efficiency of these accesses. The use of a database language compiler also greatly improved the speed as well as transferability of the program (see Chapter IV for further discussion on database compilers).

C. AREAS FOR FUTURE ENHANCEMENTS

The following areas have been identified as those which could be incorporated into the Scheduling System to improve the overall scheduling and manning requirements of the Dietetic Services Unit:

1. The present method of manually assigning annual leave cycles to employees could be automated and incorporated into the Scheduling System. A system could be

developed which allows direct access by the employees to the leave database, allowing them to input their leave requests. The system could then give immediate feedback as to which leave periods are available based on the employee's seniority and as to the number of leave days earned versus the number of leave days taken. This would alleviate the requirement of the Primary Scheduler to update the leave periods. The result of this would be to provide a more accurate picture of future manning levels and reduce the need for changes to the leave periods.

2. A method of long-term tracking of employees and jobs could be developed which would support the managerial side of the scheduling process. Statistics of absenteeism and related frequency comparisons could point the supervisors to problem areas which presently are difficult to determine. In addition, the automation of these statistics can easily lend itself to the techniques of work load forecasting which could provide management with the means of predicting a daily work force level (the number of available employees for any given day). This information can lead to a more efficient scheduling process.

REFERENCES

1. Pressman, Roger S., An Assessment of the Prototyping Approach, McGraw-Hill, Inc., New York, NY, 1987.

2. Podolsky, Joseph L., "Horace Builds a Cycle," Datamation, Vol. 23, No. 11, Nov. 1977.

3. Naumann, Justus D., and Jenkins, Milton A., "Prototyping: The New Paradigm for Systems Development," MIS Quarterly, MISRC, 269 19 Ave. S., University of Minnesota, Minneapolis, MN, Sep. 1982.

4. Sprague, Ralph H., and McNurlin, Barbara C., "New Application Development Tools and Approaches," Information Systems Management in Practice, Prentice-Hall, Englewood Cliffs, NJ, 1986, pp 231-257.

5. Alavi, Maryam, "An Assessment of the Prototyping Approach to Information Systems Development," Communications of the ACM, Vol. 27, No. 6, Jun. 1984.

6. McClure, Carma L., "Software Workbenches: The New Software Development Environment," National Computer Conference, 1987.

APPENDIX A

MINISPECS

1.0 UPDATE EMPLOYEE DATA

For each update of employee data:

1.1 If adding new employee

1.1.1 Search Employee data store for indicated employee

1.1.2 If not found then add employee to data store

1.1.3 If found check employee data for correctness

1.2 If deleting employee

1.2.1 Search Employee data store for indicated employee

1.2.2 If found then delete employee data

1.2.3 If not found then end procedure

2.0 UPDATE POSITION DATA

For each update of Job Position:

2.1 If adding new Primary/Relief Position

2.1.1 Search Position data store for indicated position

2.1.2 If found check for correctness

2.1.3 If not found insert new position data

2.2 If deleting an existing Position

2.2.1 Search Position data store for indicated position

2.2.2 If found delete position data

2.2.3 If not found end procedure

3.0 GENERATE SCHEDULE

For each date:

For each Primary Position without Specific Relief:

3.1 Obtain position data from Position data store

3.2 Check employee for:
a. Day off
b. Leave
c. Consecutive work days

3.2.1 If no conflict found schedule employee in Schedule data store.

3.2.2 If conflict found then obtain position data for designated alternate employee.

3.2.2.1 Check employee for:
a. Day off
b. Leave
c. Consecutive work days
d. Previously scheduled

3.2.2.2 If no conflict schedule employee in Schedule data store

3.2.2.3 If conflict found then select next replacement employee within wage grade and shift criteria and loop to 3.2.2.1

3.2.2.4 If no replacement found give error message

For each Specific Relief requested:

3.3 Assign relief to primary position's days-off

3.4 Check consecutive work days

3.5 If conflict delete days-off until no conflict

3.6 Schedule employee in Schedule data store

4.0 OUTPUT SCHEDULE

Upon receiving request for work schedule do the following:

4.1 Read position schedule from Schedule data store

4.2 Compile position schedules by location and output completed work schedule to printer or screen as requested by user.

APPENDIX B

DATA DICTIONARY

Data Stores

EMPLOYEE = *semantic definition
>listing of employees which includes leave periods.
{employee number + employee name + position identifier}

POSITION = *semantic definition
>listing of all food service positions to be scheduled.
{position identifier + designated relief position + days-off + shift + wage grade}

SCHEDULE = *semantic definition
>biweekly listing of all positions with scheduled employees, location, shifts and when scheduled employee is a relief position the primary position numerical identifier that will be worked that day, leave and days-off.
{position identifier + employee name + shift times + 14-day work schedule}

Data Elements

Date = *semantic definition
>combination of day of month, month of year, and year.
{month + day + year}

Days-off = *semantic definition
>days of the week which are designated as non-work days for the individual assigned to a particular position.
{first day-of-the-week + second day-of-the-week}

Designated Relief = *semantic definition
>first choice to relieve each primary position.
{position identifier + employee number}

Employee Data = *semantic definition
>listing of individual employee data.
{employee number + employee name + position identifier}

```
Employee Name = *semantic definition
                 >first and last name of employee plus middle
                  initial.
   {last name + first name + middle initial}


Employee Number = *semantic definition
                   >any number which uniquely identifies each
                    employee.
   {number}


Location = *semantic definition
           >two letter identifier of the position location.
            (PA = Palo Alto, MP = Menlo Park)
   {PA | MP}

New Employee Data = *semantic definition
                     >list of data to be edited to an
                      employee record. (Additions, deletions,
                      changes).
   {employee number + {items to be edited}}


New Position Data = *semantic definition
                     >list of data to be edited to a
                      position.(Additions, deletions,
                      changes).
   {position identifier + {items to be edited}}


Relief Position = *semantic definition
                  >combination of the location, relief
                   designation (R), and numerical identifier
                   within that location.
   {location + 'R' + numerical identifier}


Position Schedule = *semantic definition
                     >position with employee assigned for a
                      given date.
   {position identifier + employee number + date}


Schedule Request = *semantic definition
                    >request for a biweekly schedule with
                     start date.
   {date}

Shifts = *semantic definition
         >start and end times of work period.
   {start time + end time + job status + time status}
```

```
Specific Relief = *semantic definition
                  >a relief employee that scheduler desires
                   to relieve a particular primary position
                   for a given period of time.
   {primary position + relief position + leave period}


Updated Employee Data = *semantic definition
                         >list of data to be edited to a
                          employee record. (Additions,
                          deletions, changes).
   {employee number + {items to be edited}}


Updated Position Data = *semantic definition
                         >list of data to be edited to a
                          position. (Additions, deletions,
                          changes).
   {position identifier + {items to be edited}}


Wage Grade = *semantic definition
             >two letters, 'WG', plus the numerical pay
              level identifier (1-4) for full-time workers
              or 'PT' for all wage grades of part-time
              workers.
   {(WG + 1|2|3|4)|(PT)}
```

APPENDIX C

USER GUIDE

A. INTRODUCTION

The Food Services Scheduling Program has been developed to maintain a database of Personnel and Job Position information. From this database the program can produce and display on the screen (or as a printed output) a 14-day schedule, the Personnel Roster, and the Job Roster.

The database application language used to develop this program is transparent to the user, making it unnecessary to understand the command language itself. This transparency has been accomplished by designing a menu-driven system in which the user, armed only with an understanding of how the normal scheduling process works, responds to program generated questions in order to create a database of files from which the fourteen-day work schedule will be produced.

This manual's intent is to first provide a general outline in the proper use of the Food Services Scheduling Program, then to provide a more in-depth explanation of each menu option available to the user. It will also make note of pitfalls that can be avoided when using the scheduling program. (See Figure C-1 for a graphic representation of the menus available).

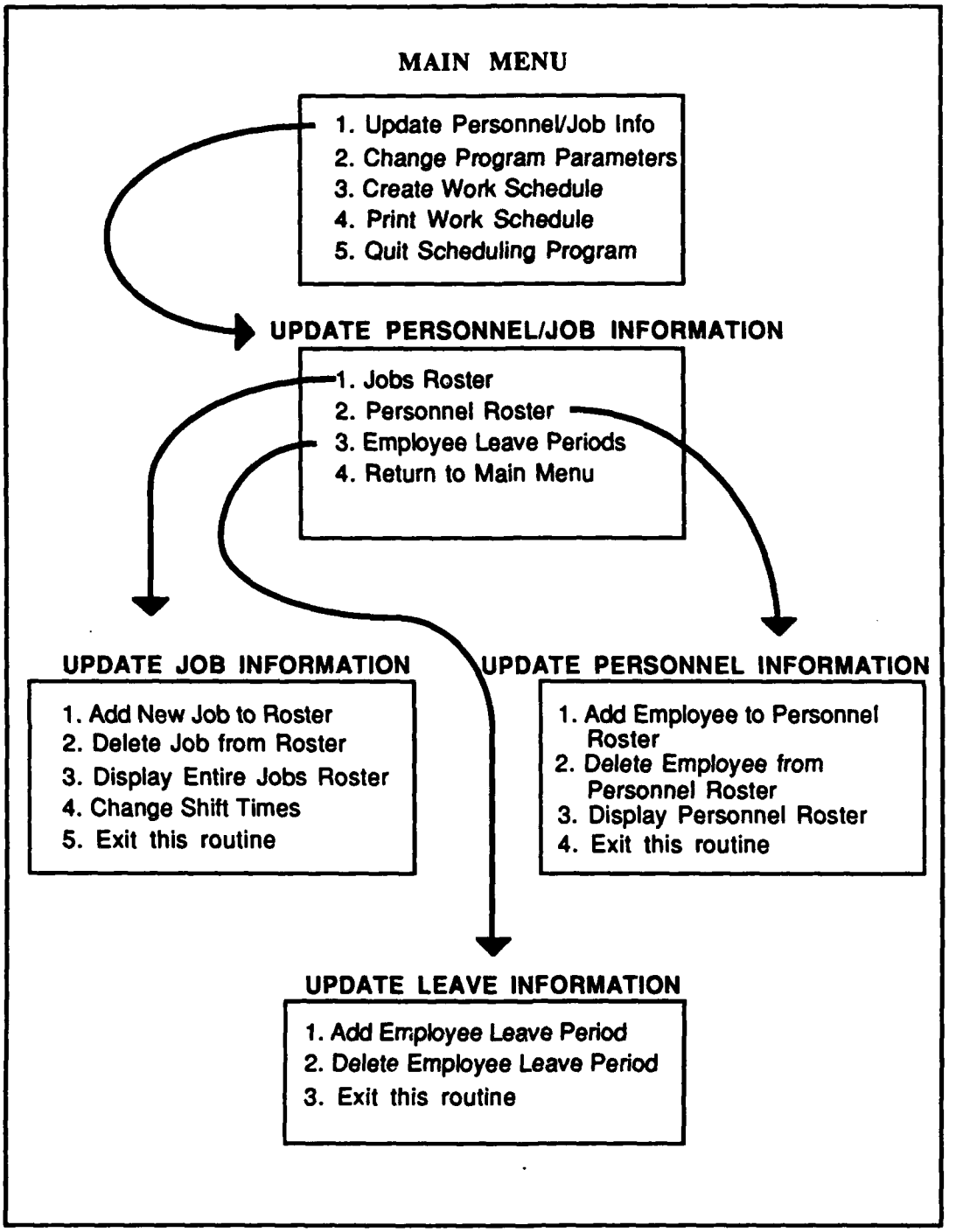


Figure (C-1) Menu Hierarchy

B. COMPUTER HARDWARE/SOFTWARE REQUIREMENTS

The Scheduling Program has been developed using the database application language dBASE III+ and compiled into an executable DOS program. The memory requirements are at least 256K of Random Access Memory and the directory location of the program must have at least 512K of usable memory. The booted CONFIG.SYS must have the parameter 'FILES = 20' included.

The program is designed to run on an IBM-PC, XT, AT, or compatible and consists of the following files which must be in the working directory:

1. MAIN.EXE
2. WORKERS.DBF
3. JOBS.DBF
4. SHIFTS.DBF
5. LEAVE.DBF
6. TEMPSKD.DBF
7. AVAIL.DBF
8. REL_LIST.DBF
9. SUBST.DBF
10. SKDLIST.DBF
11. TEMPFILE.DBF
12. TWORKERS.DBF
13. MEMVARS.MEM

NOTE: With the program installed on the hard disk, it takes approximately 20 minutes to complete a schedule. By running the program from the computer's Random Access Memory rather than from disk memory, the run time can be reduced to approximately 7 minutes.

C. PROGRAM STRUCTURE

The Scheduling Program has been custom designed for a scheduling system which consists of two groups of job positions:

1. Primary Job Positions which must be filled for each working day.
2. Relief Job Positions whose assigned personnel are used to relieve the Primary Job Positions as necessary.

In addition, each Job Position has two specific days-off assigned to it per schedule cycle. When the program encounters a Primary Job Position which has a day off or in which the assigned personnel has been scheduled for leave, it will first search the Relief Job Positions which match the same job requirements (wage grade, shift time, facility location, full-time or part-time).

If all the personnel assigned to these relief jobs are not available, the program will compile a list of all other relief jobs which match the facility location, shift times, full-time or part-time, and are one wage grade above and below the primary job in question. The user will then be prompted to select the required relief. (Amplification on

this process can be found in the menu section dealing with creating the program).

If a job position is designated as part-time, it is considered to be either a Wage Grade 1 or Wage Grade 2. The program evaluates these jobs separately from those full-time job positions with a Wage Grade 2 designation.

The built-in structure of assigning days-off and the shifting of days-off to one day earlier with each rotation prevents any employee from working more than six consecutive days without a day off. With this structure as a foundation, the program does not normally check for employees exceeding the six work day restriction.

D. HOW TO USE THE SCHEDULING PROGRAM

The Scheduling Program has been initially installed on the IBM AT computer in the Palo Alto Diet Center and has been placed in the subdirectory C:\DBASE. To start the program:

1. Turn on the IBM AT computer.
2. When the 'C:\>' symbol appears, type 'SCHEDULE' and press ENTER.
3. Follow the screen menus' directions to create a schedule.

NOTES:

a. If at any time the program fails or there is a loss of power to the computer, simply re-boot the computer by pressing the keys marked 'Ctrl', 'Alt', and 'Del' simultaneously (an alternative method is to turn the computer

OFF then ON again) and begin the process from step #2 above. All JOB, LEAVE, and PERSONNEL information updates will be stored, however, any schedule information that was currently being worked on will be lost and have to be reentered.

b. If typing the command 'SCHEDULE' does not work, then type 'CD DBASE', press ENTER, then type 'MAIN' and press ENTER.

c. If at any time during the program execution you wish to pause, press the keys marked 'Alt' and 'C' simultaneously. This will display the message 'QUIT? (Q/A/I)' in the upper right corner of the screen. Selecting 'Q'(quit) or 'A'(abort) will stop program execution. Selecting 'I'(ignore) will continue program execution from the point the pause was initiated.

A sample outline to be followed in the initial database and schedule creation is presented below. For subsequent schedules that require no changes to the Jobs Roster or Personnel Roster, simply enter the process at step #8.

1. Go to the UPDATE JOB INFORMATION menu by:

   a. displayed menu: MAIN MENU
      select: 1. Update Personnel/Job Info

   b. displayed menu: UPDATE PERSONNEL/JOB INFORMATION
      select: 1. Jobs Roster

2. Ensure all SHIFTS times have been created.

3. Add all RELIEF job positions.

4. Add all PRIMARY job positions.

5. Exit to UPDATE PERSONNEL/JOB INFORMATION menu.

6. Input new personnel to Personnel Roster.

   a. select: 2. Personnel Roster

   b. displayed menu: UPDATE PERSONNEL INFORMATION
      select: 1. Add Employee to Personnel Roster

7. Exit to UPDATE PERSONNEL/JOB INFORMATION menu.

8. Input any LEAVE periods for employees.
   select: 3. Employee Leave Periods

9. Return to MAIN MENU

10. Ensure how often days-off are rotated is correct and match the days-off to a starting date.
    select: 2. Change Program Parameters

11. Create a two week work schedule.

12. Print a created work schedule.

13. Quit the scheduling program.

E. HOW TO STOP THE PROGRAM

The preferred method of stopping the scheduling program is to select option #5 from the MAIN MENU, 'Quit Scheduling Program'. However, the user may simultaneously press the keys marked 'Alt' and 'C' and enter 'Q' to quit the program execution. As a last resort the user can turn off power to the computer, but this method is not recommended.

F. MAIN MENU

1. Update Personnel/Job Info:

Selecting this option takes the user to the UPDATE PERSONNEL/JOB INFORMATION menu. From here the user can access a series of menus in order to add/delete personnel leave periods, add/delete/display all personnel currently

used in the scheduling process, and add/delete/display all currently listed jobs.

2. **Change Program Parameters**:

Selecting this option will allow the user to modify how often the employees' days-off will rotate. It will also allow the user to match the currently listed days-off to any given schedule date. Once these two parameters have been initialized, the program will automatically rotate the days-off, even if the schedules are not sequential.

3. **Create Work Schedule**:

By selecting this option the user can begin creating a schedule that spans 14 days, beginning on a Sunday. The program will prompt for a schedule START DATE and double check to ensure it is indeed a Sunday. It will ask for CARRYOVER SPECIFIC RELIEFS which are only applicable if the user has overridden the automatic relief assigning function in the schedule just previous to this one (see Special Functions section for more details on this). The user is queried if JOB ROTATION is to occur during this schedule. If so, the program prompts for which job grouping to rotate and which employees within that job grouping NOT to rotate.

NOTE: This job rotation is temporary until the schedule is saved. It then becomes a permanent change to the jobs assigned to the employees in the Personnel Roster.

The next prompt will be for SPECIFIC RELIEFS which again can be used to manually override the automatic relief assigning function to assign a specific relief for a specific period of time. If this manual override option is used, the relief will be assigned the days-off of the primary job. If the transition to these new days-off causes the employee to exceed the six consecutive work days restriction, the program will display the two week schedule for the employee and prompt for the necessary adjustments.

Once the above inputs are complete, the program commences to create the schedule. When it finds a primary job whose assigned employee is not available to work due to a day off, leave, etc., it will first check the designated relief position assigned for that job. If the employee assigned to the designated relief position is not available, the program will then check all relief positions that match the wage grade, shift, job status (i.e., full-time or part-time) and facility location. If a relief is still not found, the program will create a list of available relief employees whose jobs match the shift, facility location, and job status of the primary position and it will then prompt the user to select the relief. If there are no reliefs available, the letters 'NR' (No Relief) will be displayed on the schedule on the appropriate date for the primary job position.

When all positions have been scheduled, the user will be given the option of reviewing an individual position's

schedule on the screen, reviewing the entire schedule on the screen, saving the new schedule, or deleting the new schedule.

4. Print Work Schedule:

From this option the user can printout any given schedule that is on file. The maximum number of schedules that can be saved is six (6). The printed schedule will be listed by LOCATION (i.e., Palo Alto, Menlo Park), with one week per printed page. (An 8 X 11 inch paper size should be used).

5. Quit Scheduling Program:

Selecting this option will complete the scheduling program and exit the user to the DOS prompt. This is the preferred way to complete the program usage; however, the program SHOULD not be damaged if the computer is turned off prior to terminating the program.

G. UPDATE PERSONNEL/JOB INFORMATION MENU

1. Jobs Roster:

Selecting this option takes the user to the UPDATE JOB INFORMATION menu. From here the user can add/delete/display the information in the current Jobs Roster as well as the SHIFTS file. The user will be prompted for necessary job information and will display the entered data for confirmation before saving.

NOTE: The Relief Job Positions must be entered BEFORE the Primary Job Positions because each primary position requires an existing relief position to be assigned the designated relief.

2. Personnel Roster:

Selecting this option takes the user to the UPDATE PERSONNEL INFORMATION menu. From here the user can add/delete/display the information in the current Personnel Roster. The user is prompted for Employee SSN, Employee Name, and the Job Position assigned.

NOTES:

a. The Job Position must exist BEFORE it can be assigned to an employee.

b. The employee SSN is the key element and must be accurate.

3. Employee Leave Periods:

Selecting this option takes the user to the UPDATE LEAVE INFORMATION menu. From here the user can add/delete employee leave periods. The user will be prompted for Employee SSN, the First day of leave, the Last day of leave, and the Reason for leave.

NOTES:

a. This option can also be used to give an employee an arbitrary day off simply by entering the Reason as 'OFF'.

b. The Employee SSN is the key element and must be accurate.

4. Return to Main Menu:

Selecting this option returns the user to the MAIN MENU described above.

H. UPDATE JOB INFORMATION MENU

NOTE: In order to change the information of a currently listed Job Position, it must first be deleted, then re-added with the necessary change information.

1. Add New Job to Roster:

This option allows the user to add a new job position to the Jobs Roster. The program prompts for the following data inputs:

a. The job position identifier.

b. The designated RELIEF position if the primary position is not a relief position as identified by the third letter being a 'R'. (i.e., PAR23).

c. The first day off during the week (the second day off is automatically calculated).

d. The shift number from a displayed list of available shift times.

e. The wage grade of the position.

f. The building the position is located in.

g. The work area the position is assigned to.

NOTES:

a. The RELIEF position MUST exist in the Jobs Roster BEFORE it can be assigned as a relief for a primary position.

b. The RELIEF positions of matching wage grade, shift times, job status (i.e., full-time or part-time) and facility location are "daisy-chained" together. This "chain" is then searched when a primary position of matching requirements needs a relief. (See Figure C-2). If the user desires that the job reliefs be evenly spread throughout the relief positions, then the designated relief positions assigned to a primary position should vary, resulting in different entry points into the "daisy-chain". If the user desires that the first available relief be used until it is completely scheduled, then the designated relief position assigned to all primary positions of like requirements should be the same.

c. The first and second days-off will be assigned sequentially (i.e., SATURDAY/SUNDAY, SUNDAY/MONDAY, FRIDAY/SATURDAY).

d. The Building and Work Area inputs are optional and can be left blank; however, doing this may cause the output to appear strange because the jobs are sorted and printed in the following sorted order:

1. LOCATION

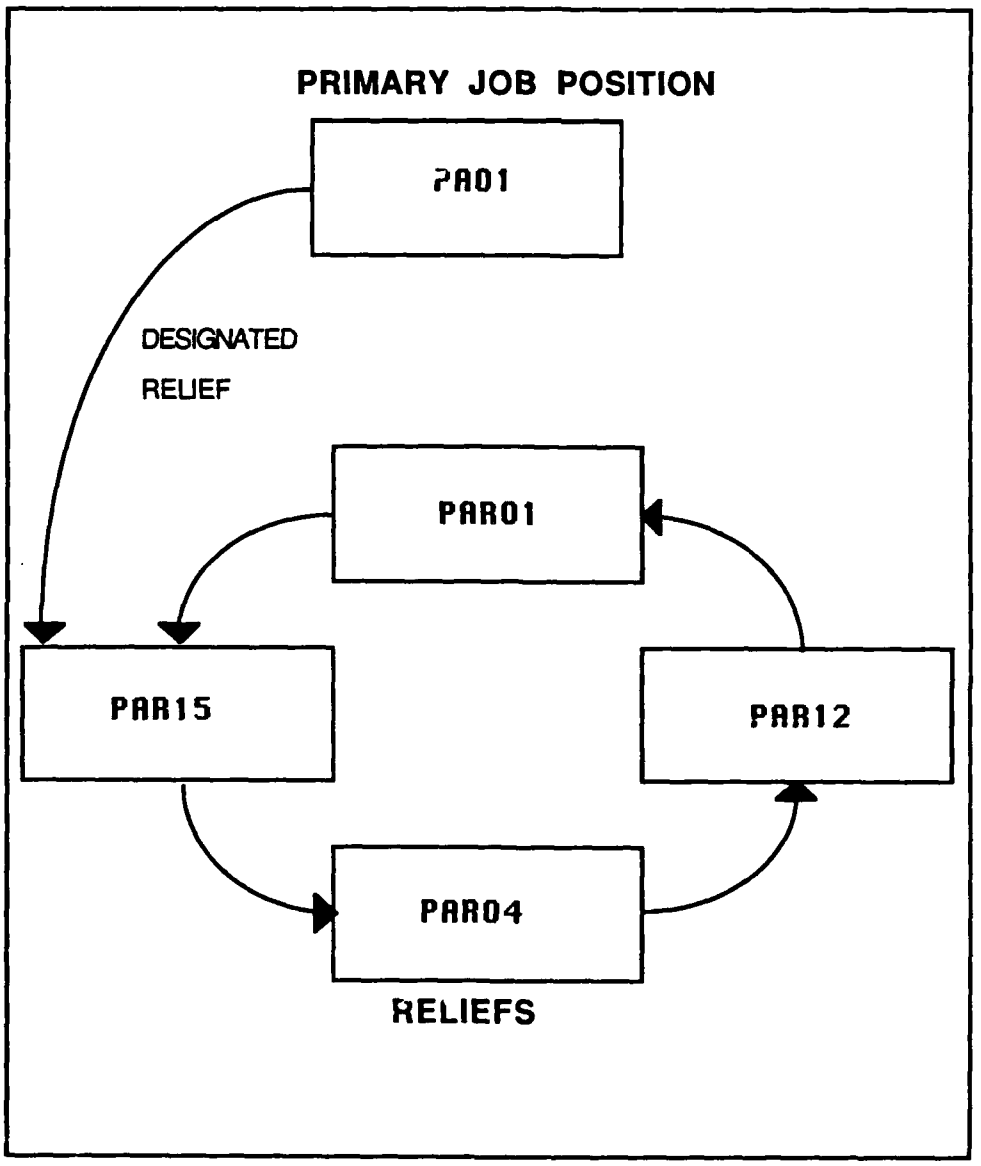


Figure (C-2) Relief Daisy-Chain

2. POSITION NUMBER

3. BUILDING

4. WORK AREA

e. It is important that all relief positions have a 'R' as the third letter and that all single digit position numbers have a preceding number zero (0) NOT an alphabetic Oh (O) (e.g., PAR01, MP01).

2. Delete Job from Roster:

Using this option, the user can delete jobs from the Jobs Roster. The user is prompted for the position identifier. If it is a relief position (i.e., PAR22), the program will ensure the "daisy-chain" continuity is maintained as well as assigning a new designated relief to all primary job positions which had previously been assigned the deleted position as a relief. If the deleted position is a relief that has no other "daisy-chain" reliefs, then the program will REQUIRE the user to input a new designated relief for the primary positions affected.

3. Display Entire Jobs Roster:

Selecting this option allows the user to view on the screen or output to a printer the Jobs Roster.

4. Change Shift Times:

This option allows the user to view the currently stored shift times and to add/delete shifts as necessary. The program prompts for Shift Number (an arbitrary number used in cross-referencing this file with the Jobs Roster),

Start Time, End Time, whether the shift is an early (AM) or late (PM) shift, and the shift status (i.e., a full-time (FT) or part-time (PT) shift).

5. Exit this routine:

Selecting this option returns the user to the UPDATE PERSONNEL/JOB INFORMATION menu.

I. UPDATE PERSONNEL INFORMATION MENU

NOTE: If the information of a currently listed employee needs to be changed, that employee must first be deleted then re-added.

1. Add Employee to Personnel Roster:

This option is used to add new employee information to the Personnel Roster. The program will prompt for Employee SSN, Employee Name, and the assigned job position. The Employee SSN is the key element and must be accurate.

NOTE: The assigned Job Position must exist in the Jobs Roster BEFORE it can be assigned to the employee.

2. Delete Employee from Personnel Roster:

This option is used to delete employees from the Personnel Roster. The program will prompt for the Employee SSN and will then delete all information associated with that SSN from the Personnel Roster as well as from the Leave file.

3. Display Personnel Roster:

Selecting this option allows the user to output the Personnel Roster to either the screen or the printer. This output can be listed in order of Employee Name, Employee Job Position, or Employee Entrance on Duty date.

4. Exit this routine:

Selecting this option will return the user to the UPDATE PERSONNEL/JOB INFORMATION menu.

J. UPDATE LEAVE INFORMATION MENU

NOTE: The Employee SSN is the key element in this menu and must be accurate.

1. Add Employee Leave Period:

This option is used to add employee leave periods to the Leave List. The program will prompt for an Employee SSN, check the Personnel Roster to ensure the SSN matches one on file, display the employee name associated with the entered SSN, and request user confirmation. The program will display all leave on file associated with the selected employee and prompt the user for a starting date, ending date, and reason for the leave period being added.

NOTE: The Employee SSN is the key element and must be accurate.

2. Delete Employee Leave Period:

This option is used to delete an employee leave period already on file. The program will prompt for an Employee SSN, check the Personnel Roster to ensure the SSN matches one on file, display the employee name associated with the entered SSN, and request user confirmation. The program will display all leave on file associated with the selected employee and prompt the user for a starting date and ending date from which ALL leave entries will be deleted.

3. Exit this routine:

Selecting this option will return the user to the UPDATE PERSONNEL/JOB INFORMATION menu.

K. SPECIAL FUNCTIONS

1. Specific Reliefs/Carryover Specific Reliefs:

This function can be used to override the automatic assigning of reliefs to a specific primary job position. The user will be prompted for the PRIMARY job position to be relieved, the RELIEF job position, the First date of relief, and the Last date of relief. It will then transition the relief position to the primary position's days-off and inform the user if the six consecutive work day (or more than four scheduled days-off) restriction has been exceeded. It will also transition the relief position back to its normal days-off at the end of the relieving period, taking into account the above restrictions.

This relieving period may be of any length. If it extends into the next schedule period, the user will be prompted to input the necessary information via the CARRYOVER SPECIFIC RELIEF routine. If the user decides NOT to carryover the specific relief, the program will then transition the relief position to its normal days-off.

CAUTION: This routine assigns the relief to the days-off of the primary position, thereby creating the possibility of the user being required to decide which day off to delete if there are too many, or which day off to add if there are not enough.

2. **Job Rotation**:

This routine allows the user to rotate employees through specific job groupings based on facility location, position wage grade requirements, position shift times (i.e., early or late), and job status (i.e., full-time or part-time). Once a job group has been identified, it will be displayed to the user along with the present employees assigned to it. The user will then be asked to input any employees of this group that are NOT to be rotated. The program then disregards any jobs that have employees not to be rotated, sorts the jobs according to days-off starting at the top with SATURDAY/SUNDAY, FRIDAY/SATURDAY, etc., and ending with SUNDAY/MONDAY. The employees are then rotated down one job. If the program finds an employee who will

exceed the six consecutive work day restriction as a result of this rotation, it will add a day off on the first day of the new schedule and delete the next normal day off.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Library, Code 0142 Naval Postgraduate School Monterey, California 93943-5002	2
2. Superintendent, Naval Postgraduate School Computer Technology Programs, Code 37 Monterey, California 93943-5000	1
3. Professor Tarek Abdel-Hamid, Code 54Ah Department of Administrative Sciences Naval Postgraduate School Monterey, California 93943-5000	1
4. LCDR James M. Buyske 1048 Summersong Ct. San Jose, California 95132	1
5. Fleet Air Reconnaissance Squadron Three Attn: LT Steven V. Call NAS Barbers Pt., Hawaii 96862-6517	2
6. Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
7. Director, Information Systems (OP-945) Office of the Chief of Naval Operations Navy Department Washington, D. C. 20350-2000	1