FLIGHT TRANSPORTATION LABORATORY

REPORT R 84-5

INTERACTIVE DYNAMIC AIRCRAFT SCHEDULING

Thomas A. Deckwitz

June 1984

ACKNOWLEDGEMENTS

I am indebted to Professor Robert W Simpson as my advisor on this project for introducing me to the topic and his constant assistance and encouragement.

I would also like to thank the following special people for their friendship and support: the faculty and staff of the Flight Transportation Laboratory and the Wright Brothers' Wind Tunnel, the members of the MIT Ice Dance and Figure Skating clubs, my pairs skating partner, and my close friends in Seattle.

I would like to dedicate this thesis to my parents and brother for their inspirational love and friendship and their unfailing support and encouragement.

FLIGHT TRANSPORTATION LABORATORY
REPORT R84-5

INTERACTIVE DYNAMIC AIRCRAFT SCHEDULING

BY: THOMAS A. DECKWITZ

FLIGHT TRANSPORTATION LABORATORY REPORT R84-5

INTERACTIVE DYNAMIC AIRCRAFT SCHEDULING

by

Thomas Anthony Deckwitz

ABSTRACT

Introducing recent advances in computer technology to improve aircraft scheduling is investigated. Incorporating interactive graphics, modern database manipulation techniques, and decision support algorithms, the computer is proposed as a tool for the schedule development process, replacing present manual methods.

A detailed set of graphics representations of schedule data are presented based on the sequence chart and station activity chart. The interactive manipulation of these displays by the scheduler results in an immediate appropriate update of the schedule database. Quick graphics response and automatic constraint violation alerts speed the search for feasible schedules.

The execution of complex aircraft scheduling operations by the proposed system is presented. Schedule display and database structures are designed for implementation on computers with modern high resolution graphics and pointer directed list capabilities.

Interactive Dynamic Aircraft Scheduling
Table of Contents

Interactive Dynamic Aircraft Scheduling

List of Figures

List of Figures

(continued)

FIGURE PAGE

List of Figures

(continued)

1 INTRODUCTION

Scheduling plays a central role in airline and airlift planning and operations. No operational activities can be performed independently of the schedule and overall operational efficiency is determined by the specific scheduling of aircraft and crews. Improved scheduling efficiency can have the same impact as adding aircraft to the fleet. Scheduling is a complex function which routes aircraft, payloads, and crews in both time and space subject to a multiplicity of operational constraints.

Airlines and airlift operators have the opportunity to purchase extremely fuel efficient, advanced technology aircraft and most employ modern communications systems. Aircraft scheduling, however, is one function to which modern techniques and technologies have not been applied. Scheduling today is still done manually using wall charts, paper slips, reference books and colored pencils. The introduction of computerized scheduling processes has been limited, and schedule generation algorithms have not been able to solve even moderately sized scheduling problems.

This report introduces the modern graphics computer as the sole tool necessary to handle the complexities of airline and airlift scheduling. The scheduling system uses graphics displays to represent scheduling information.

Manipulation of the screen's symbols is interpreted as scheduling commands by the system which automatically updates the database, tests schedule constraints for feasibility, and redisplays the newly modified information immediately. Decision support algorithms and numerous automated scheduling subprocesses are made interactive with the scheduler who controls all final scheduling decisions.

This system is intended to speed and simplify schedule development over present manual methods, but not to automate the schedule process. This design recognizes that the human role in the complex scheduling processes is indispensable. The computer system will be used as a tool to present scheduling information in a convenient format, allow for quick and easy manipulation of the schedule, and free the scheduler for many tedious scheduling tasks through helpful automation of sub-processes. By increasing the speed of the scheduling processes, more scheduling and rescheduling alternatives can be investigated in each scheduling cycle, resulting in a better final product (the schedule) for the airline or airlift operator.

The proposed interactive dynamic aircraft scheduling system has been designed to handle a most difficult scenario: the dynamic overload case in airlift. In this scenario, demands, resources, and constraints are dynamically changing and demands always exceed airlift capabilities. Frequent rescheduling of aircraft and crews occurs constantly in this worst case scenario.

Chapter 2 introduces the airlift scheduling problem and defines terms and processes needed to understand the more detailed following chapters. The designs and options for graphics displays of scheduling information are outlined in Chapter 3. Chapter 4 discusses the scheduling operations which will be used to build and modify complete schedules. Automated scheduling subfunctions, decision support algorithms and sample scheduling problems are reviewed. Advanced scheduling topics are detailed in Chapter 5.

2. SYSTEM DESIGN AND DEFINITIONS

2.1 Design Scenario

The main purpose of designing an interactive dynamic aircraft scheduling system is to improve the scheduling of multiple interdependent resources for efficient flow of commodities across a network. The application of the basic design philosophy is not limited to aircraft scheduling, rather any resource and commodity combination that must be schedule in both time and space. This design concentrates on scheduling in an airlift environment: the most dynamic of aircraft scheduling scenarios.

The scheduling system design criteria is to have the capability of handling the 'worst case' airlift scenario: the dynamic overload scenario. During dynamic overload, airlift resources are inadequate in satisfying the demands, thereby causing tradeoffs in accepting newer higher priority and rejecting older lower priority requests. In airlift, demands are dynamic in the sense that they are continuously arriving and changing. Available resources of aircraft, crews, stations are also assumed to be dynamically changing.

At this point, it is assumed that the reader has a basic understanding of aircraft scheduling and familiarity with the present status of scheduling tools and processes. A clear discussion of these topics can be found in Mr. E.

Lubow's Master's Thesis (see References). He dicusses the very basic elements of scheduling and includes an extensive review of present scheduling methods.

Although there are a number of small differences, airlift and airline scheduling are very similar. Airline scheduling could be considered an easier subset of airlift scheduling. Airline scheduling, by its nature, is less dynamic than airlift scheduling. Demands in airline scheduling are mostly symmetrical; a helpful scheduling characteristic not found in the airlift case.

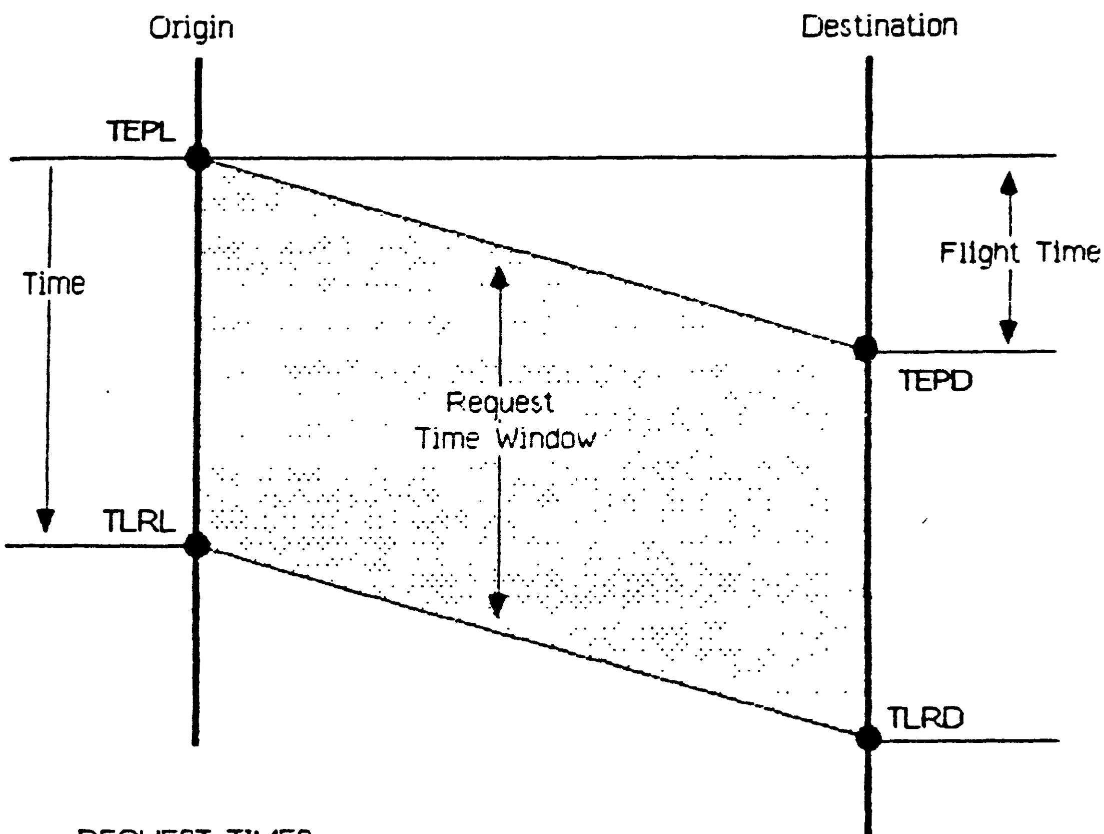
This chapter introduces the basic airlift scheduling processes and elements. A general outline for the software and database design is also discussed.

2.2 Airlift Demand (Requests and Tasks)

Demand for airlift comes in the form of individual user's requests. Each request calls for the transportation of a certain weight/volume of cargo (or number of passengers) from an origin to a destination. The request also has some timing requirements which can be very specific ("between noon and dusk") or very vague ("when can it be done soonest"). To quantify this request identifier, we establish an earliest and latest time for the transport thereby creating a "time window" for execution of the request. Figure 2.2.1 shows the set of times assoc-

FIGURE 2.2.1

Airlift Request Time Window



REQUEST TIMES

TREQ - time of request initiation/modification
TEPL - time of earliest possible load at origin
TLRL = time of latest required load at origin
TEPD - time of earliest possible delivery at destination
TLRD - time of latest required delivery at destination

REQUEST DATA - priority,user,load,special handling,specific aircraft type,etc.

iated with a time window. The time of earliest possible loading, TEPL, and the time of latest required delivery, TLRD, alone can identify the entire window.

The load sizes for airlift requests can vary greatly. For small loads the space onboard the aircraft can be shared, and requests that overlap in time and space can be aggregated. This is part of the mission generation process described in the next section. On the other hand, a single request could also specify multiple plane loads from various origins/destinations with interrelated delivery times.

Each request is given a priority ranking which will determine the order of eligibility for a given mission operating at a certain time and space. In the dynamic overload scenario, it is the objective of the scheduler to accept as many of the higher priority requests as possible through efficient scheduling and routing aircraft. Requests will thus compete with each other in time and space as well as priority. Lower priority requests may be fortunate enough to piggyback on empty positioning flight segments as aircraft are routed to higher priority requests.

Airlift requests are dynamic and are continuously arriving for processing by the scheduler. The users may submit the requests anywhere from weeks to hours before the desired execution time. The user may also subsequently modify the request by changing the origin, destination, priority, etc. or even delete the request completely. In the overload scenario, it is the schedule which defines which requests can be handled, allowing confirmation messages to be sent to those users. A confirmed request shall be defined as a "requirement". Unconfirmed requests are retained for consideration since it may be possible to accept them later due to the dynamically changing scenario.

2.3 Mission Generation

The mission is the basic scheduling element in airlift. An Airlift Mission is defined as the movement of a given type of aircraft with sufficient capacity to carry its assigned requests between their origins and destinations along a specified routing. Additionally, the mission is described by a priority and time window derived from the assigned requests. Initially, the mission does not have scheduled departure or arrival times, although specified segment flying time, loading and unloading times are calculated. The mission ends when all cargo is unloaded at any point. Positioning and depositioning flights are not included in the mission.

Mission Generation is the process of converting the user requests into airlift missions. It includes the selection of an aircraft type and an efficient routing to serve one or more requests. The simplest case is the "Single

Mission Request" where one request becomes one mission, and the smallest aircraft capable of transporting the payload is selected. The mission time window and priority are exactly those of the request's.

If their locations in both time and space are compatible, a number of small payload requests may be aggregated in order to efficiently use the capacity of an aircraft along a routing which serves those requests. This is defined as a "Multiple Request Mission". Figure 2.3.1 shows an example of the aggregation of four compatible requests into one two segment mission. The mission time window is derived from the intersection of the four requests' time windows. The scheduler may want to create an airlift mission on the expectation of future requests which will fill the selected capacity. When an appropriate request arrives, it is assigned to this mission if there is still space available onboard. Available is defined as there being not enough volume of user requests of equal or higher priority to fill the selected capacity. To accommodate a new request, the scheduler may "bump" other previously assigned requests of lower priority. This method of bumping avoids the generation of another mission and may save the use of one or more aircraft for other missions.

The final case of mission generation is the "Multiple Mission Request". This is where a single request must be decomposed into a multiple set of missions, all having the

FIGURE 2.3.1

Single Mission from Multiple Requests

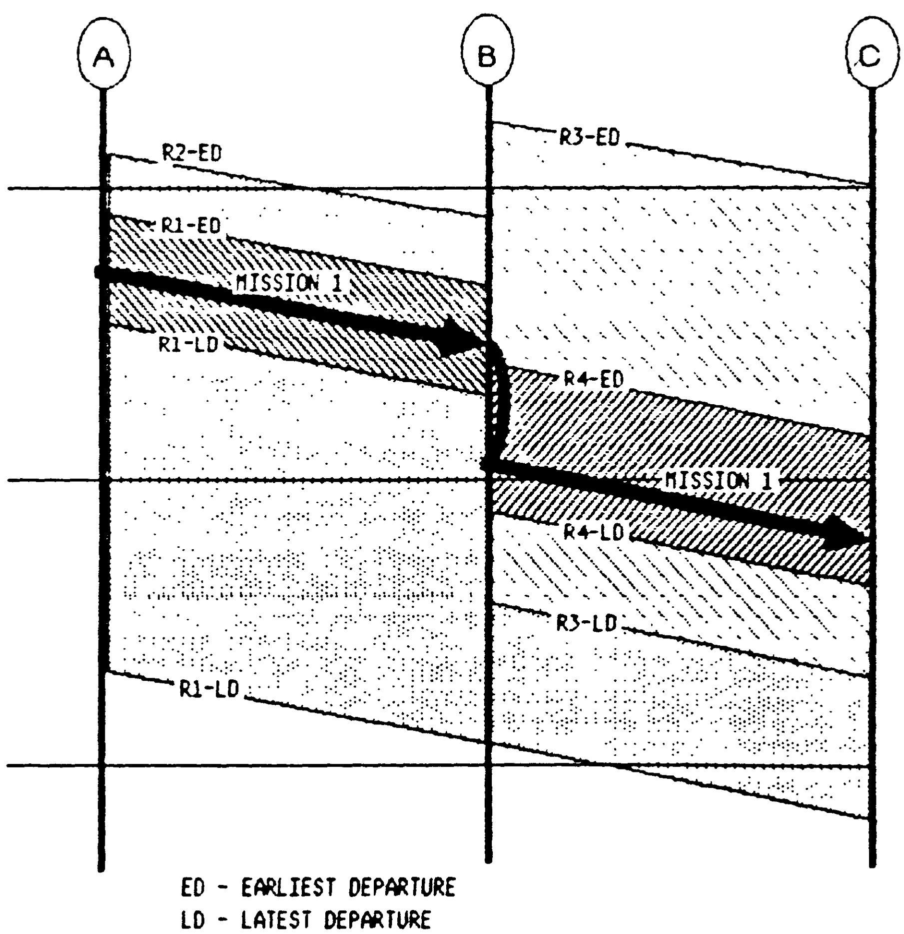
Request R1, W1 tons from A to B, priority β, time window TW1

Request R2, W2 tons from A to C, priority β, time window TW2

Request R3, W3 tons from B to C, priority α, time window TW3

Request R4, W4 tons from B to C, priority β, time window TW4

Mission 1, priority α , C-141, serves all 4 requests



Mission 1 -- C-141 from A to B to C within reduced time window

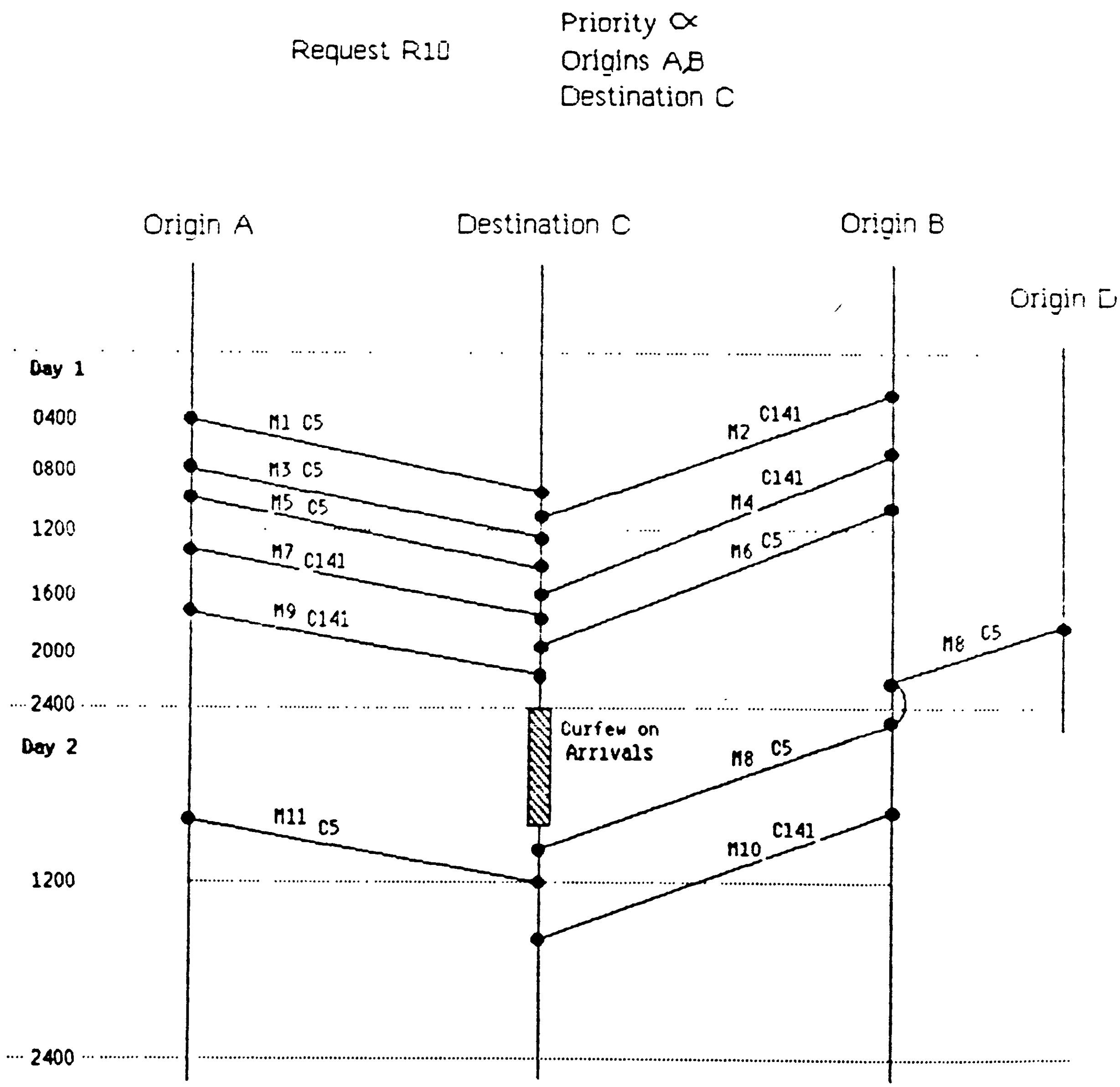
same priority as the request and flown by one or more aircraft types between multiple origins/destinations.
Figure 2.3.2 shows the decomposition of a single request that generated eleven missions. Two different aircraft types are used to transport payloads from three different origins to a single destination.

2.4 Mission Scheduling an Routing

The next major function of the airlift scheduling process is "Mission Scheduling and Routing". This function is undoubtedly the core of the scheduling process. It is where actual departure/arrival times are given to missions, where aircraft are routed, and where crews are assigned and routed, given a list of currently assigned missions. The scheduler's objective, in the dynamic overload scenario, is to find an efficient routing of aircraft and crews so as to minimize the number of higher priority missions that cannot be flown.

The routing of aircraft and crews introduces the idea of sequencing. An "Aircraft Mission Sequence" is a linked set of missions (and their appropiate positioning flights) to be flown successively by a given aircraft type. Similarly, the "Crew Mission Sequence" is defined as a set of flight segments to be flown by a crew qualified in that type of aircraft. These sequences must satisfy the given

FIGURE 2.3.2 **Multiple Missions from a Single Request**



aircraft and crew performance restrictions. Note that these sequences have not yet specified an exact aircraft tail number or named crew at this point, similar to generated missions not having specific times.

Once the missions, positioning flights, and mission sequences have been given specific times and the crew and aircraft routings are feasible the schedule is basically complete. Airlift demand being dynamic, there will be a constant need to reschedule the aircraft and crews as late arriving requests are added to the schedule, bumping lower priority missions. The rescheduling process can be taken back to the mission generation function as well. Missions can be regenerated from the new set of user requests for more efficient use of airlift resources.

Satisfying aircraft and crew routing restrictions are not the only constraints to airlift scheduling. Station constraints can be also very restrictive. Stations have finite resources for handling airlift activities. Each station has, for example, a limited area for aircraft parking and hence there is a maximum number of aircraft that can be on the ground at the station at one time. This maximum is called a MOG (Maximum on Ground). The scheduler must avoid exceeding the local MOG when scheduling missions. Another important station constraint concerns the loading and unloading of aircraft. At the Military Airlift Command, the palletized payloads are loaded and unloaded with K-loaders. A station has a finite number of these. The scheduler must plan aircraft ground times at a specific station for loading and unloading of cargo. Aircraft may be delayed on the ground if there are no K-loaders available upon arrival.

This section has briefly introduced the scheduling process outlines its basic functions and constraints. Chapter 4 describes this process in far greater detail. However, before continuing with a discussion of what the proposed interactive dynamic aircraft scheduling system will handle, let us review the requirements this complex scheduling process places on the design of the computer software.

2.5 Database General Design

The interactive dynamic aircraft scheduling system uses the computer as the sole tool for scheduling. The computer is asked to do the following tasks: 1) maintain a complete schedule database including all airlift resources and all demands. 2) be able to draw numerous representative graphics displays of the schedule. 3) execute a number of complex schedule modification operations. 4) run subfunctions and optimization algorithms on the schedule data.

The two main activities that the computer software must handle are data retrieval and data management. The data retrieval activity is needed for drawing displays, and searching for specific data. Data management is needed during data modification operations. In designing an interactive system, a critical software design criteria is running speed. If we wish to optimize the schedule database for quick data retrieval, multiple storage of data shortens search time. For each different proposed display a data structure would be created that would search through that particular display data in the quickest manner. Thus, with a number of displays, there would be a number of data structures each representing the same data in a different way. Therefore, there is multiple storage of data. However, if we wish to speed data management, it is important to design the database with as little data redundancy as possible. This will minimize the number of changes to the database during a data modification operation. In trying to minimize running time of scheduling software, there is a design tradeoff when considering the activities of both the data retrieval and the data management.

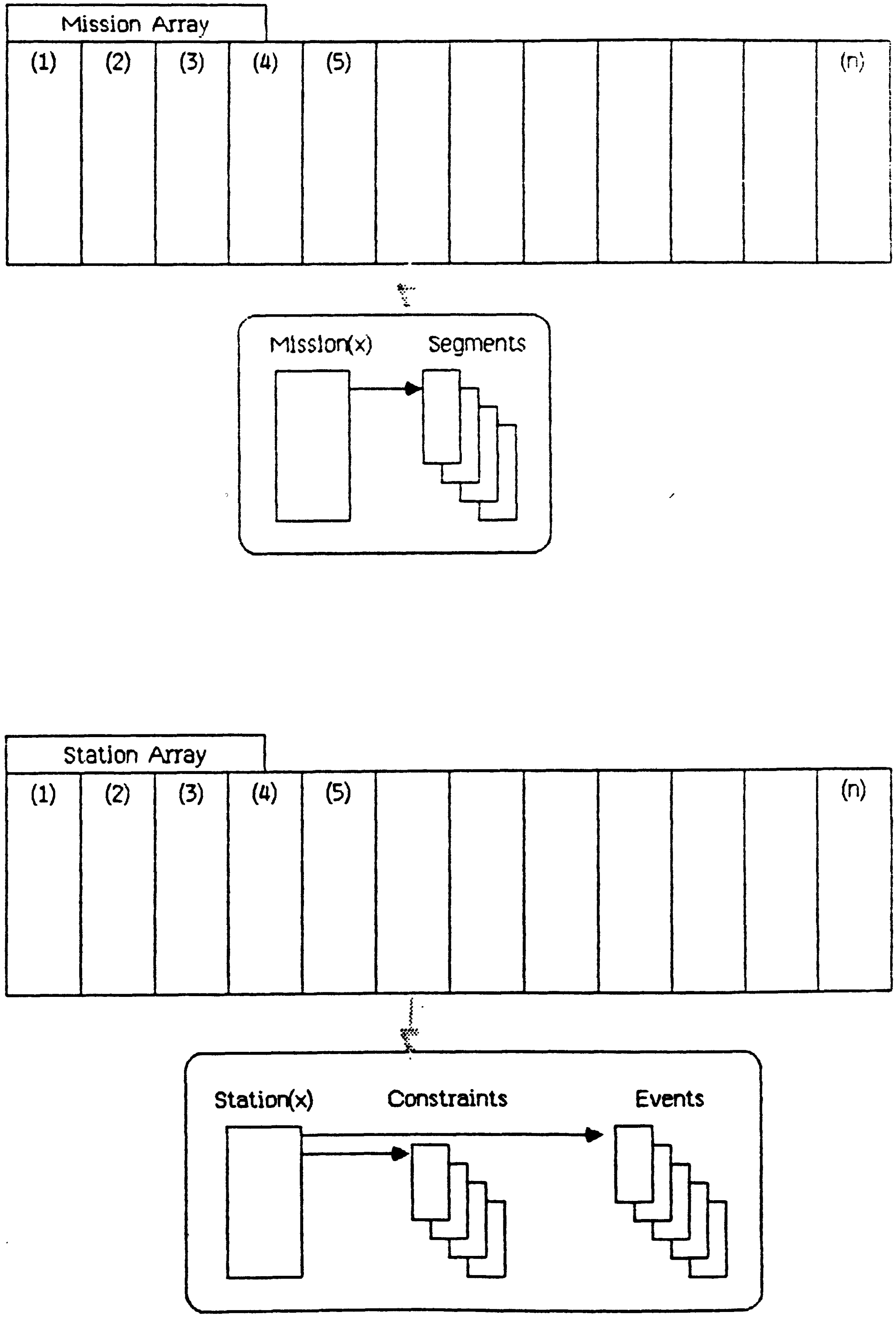
Rescheduling is the major function in the overall scheduling process. This requires a great deal of data modification and management. Yet for each modification the data, the system must retrieve schedule information for an updated display. Thus, the activities of data retrieval and data management are evenly distributed during the scheduling process.

The general design of the schedule database should be a compromise between absolute non-redundance of data storage and wasteful multiple storage of all data. A general database design strategy would be as follows: 1) Minimize the redundancy of data. Only important descriptive data elements should be allowed to be stored more than once. Due to the large amount of resource, demand and constraint data in a large scale airlift scenario, it is imperative to minimize redundancy. 2) In order not to neglect the importance of data retrieval, a modern list-based data structure using pointers is introduced to speed the search and retrieval of data without requiring extra data storage. This general design strategy allows for advanced software speed in both data retrieval and management.

The following are a more specific design principles for the scheduling database. The foundation of the database should lie with one or more of the major scheduling elements: missions, aircraft or stations. Separating the database between demand related information and scheduled times related information would be logical. The demand related data of the schedule database should use the mission as the basic element since demand information comes in terms of user requests which are converted by the scheduler to missions.The scheduled times information should be stored in a structure based on the station in the form of station event schedules. See Figure 2.5.1.

The mission database structure 'holds' the demand related information. User requests for airlift demand are initially stored in a task database. The scheduler, when

FIGURE 2.5.1 **Scheduling Database Structures**



building a schedule, generates candidate missions from this data. The generated missions and positioning flight information is then stored in elements identified by the mission or positioning flight number/identifier. Figure 2.5.2 shows the proposed structure of the mission database. Each mission element holds information regarding the specific mission such as priority, aircraft type, mission number, time window. From the main mission description record a pointer directs one to the list of mission segments. Specific segment duration times and characteristics are found in these segment records. Note that those are times still unscheduled at present. The only data stored in the mission database structure that is dependent on specific scheduling are the routing sequence and assignment (crew, tail number, division) data.

Different from the mission database structure, the station database structure hold most the dynamic scheduled events and constraints information. Each station has a few unchanging pieces of information, such as identifier, location, region, etc. The dynamically changing station information is stored in two time series record lists with pointerts. The first list holds a record for each event occuring at the station. Each new record updates the station's status as well as describes the event. The second list holds the dynamic station and scheduling constraints ordered by their effective times (Figure 2.5.3). During rescheduling, arrival

FIGURE 2.5.2 **Mission Database Structure**

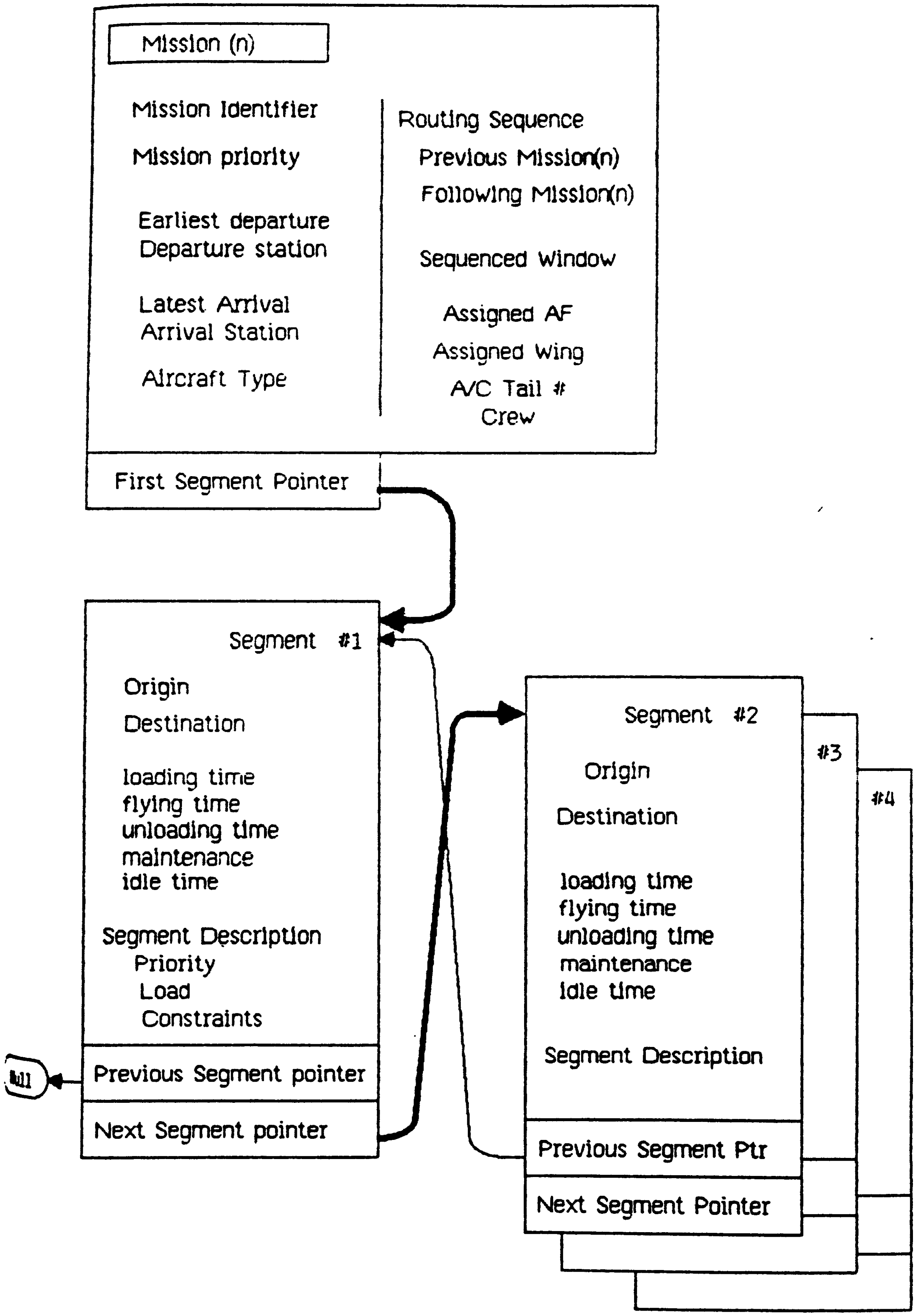
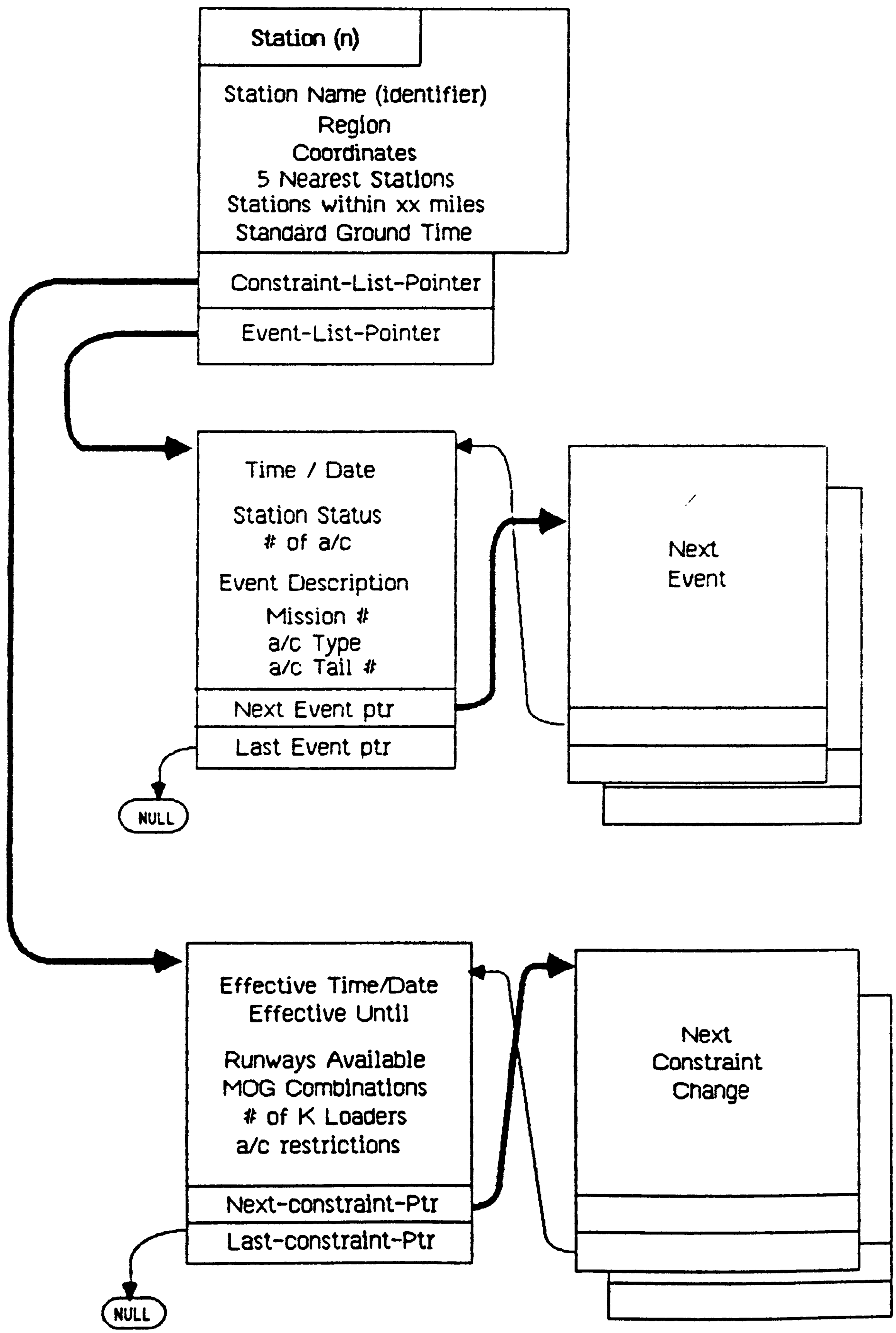


FIGURE 2.5.3 **Station Database Structure**



and departure times are modified most often. These changes will occur in the first list which must be updated each time. For every update of the scheduled events list, the station constraint list is consulted and schedule feasibility is checked. Since the demand remains unchanged in rescheduling so too does much of the mission database remain unchanged.

The advantage of this general database design will become clearer in later chapters. The use of pointers to direct the search for specific data in records (rather than a linear search through all data) helps speed software running time. Minimum data storage redundancy speeds modification procedures. The record list is effectively used for frequently changed information. Pointer defined lists allow for dynamic storage allocation meaning only as much storage space as is absolutely necessary is reserved by the system. Station database design must be flexible enough to accept all complexities of the scheduling process.

2.6 Alternate Approaches to Scheduling Automation

An objective of this system design is to introduce computer automation into the scheduling processes. There are four recognizable levels of scheduling automation. First, the computer could assist the scheduler in generating missions and schedules with a traditional database management system. This is currently used to speed the manipulation of pertinent alpha numeric data for the scheduling functions. Secondly, sophisticated graphic displays could be introduced, and interactive graphics methodologies could be used to manipulate scheduling data. Thirdly, some of the sub-processes in mission and schedule generation could be automated and called into interactive use to support and speed the decision making process. The highest level of automation for mission and schedule generation processes would use computer algorithms to seek complete optimal schedules.

The first level of scheduling automation is already in use and has been shown to speed the overall scheduling process. It allows the airlift scheduler to retrieve and enter scheduling information quickly. Systems have also been tested which assist the scheduler with simple schedule making decisions by adding automated support of flying time, ground time, time zone changes, flags for MOG's or curfews, etc.

Designing modern graphics displays of scheduling information with interactive symbolic manipulation is also within the current state of the art. The iterative search for scheduling information is made both simpler and easier, but this option requires the design of complicated graphis-to-scheduling database software. Introducing automated decision support sub-systems to assist the airlift scheduler as he seeks improved schedules is also feasible; but it is not in the current state of the art to automate completely even a moderately sized airlift scheduling process, especially if optimal answers are expected. Computer hardware size and speed are not the limiting factors but the poor performance and non-existence of scheduling algorithms for solving such complex scheduling problems at a large scale.

The adopted approach is to apply interactive graphics and interactive decision support automation to the airlift scheduling process. It has the advantage of easy transition from current practices and manual reversion should equipment or communications fail. This approach allows the application of human experience and judgement in creating schedules. The airlift scheduler may know the probability of successful execution of certain schedule segments is small and avoid them eventhough they are theoretically feasible. A scheduler may also have a backup schedule plan in mind in the event a schedule begins to fail. The completely automated scheduling approach would have no information on these expectations. It would generate good schedules feasible with the constraints and resource restrictions for which it is programmed to test against. The scheduler's experience will lead him to prefer schedules of slightly lower overall productivity but of higher probability of successful execution of future modification.

The interactive dynamic aircraft scheduling system will be a tool to speed and simplify the decision making processes of airlift scheduling, leaving the airlift scheduler with the responsibility for generating good schedules and totally in control of the schedule generation process. All final scheduling decisions are his alone. A computer equipped with modern graphics capabilities will become the scheduler's sole tool in both planning future schedules and updating existing schedules due to changing scenarios.

3. GRAPHICS DISPLAYS

3.1 Introduction

The object of this project is to speed and simplify aircraft scheduling and rescheduling without losing any necessary detail. Our goal is the introduction of new computer graphics offered today as a complete replacement for the scheduler's collection of pencils, rulers, data-charts, scratchpads, and wall charts. We aim to take the pencil away from the scheduler and replace it with a graphics display manipulation device ... the mouse. The result will hopefully be the disappearance of mountains of paper slips, scratchpads, and reference books. The storage and quick retrieval of all information should shift more of the scheduler's time to actually scheduling (adding, deleting, moving flights) and away from testing constraints, looking up data, and calculating times.

This section addresses the problem of making the schedule and scheduling process easily and totally understandable using only a modern computer graphics screen. The scheduling of airlift resources is a multidimensional problem in both time and space. Not only aircraft, but also crews and airport resources are considered in the overall airlift scheduling picture.

A clear and concise set of two-dimensional displays must be designed for clarity and ease of manipulation. These displays must have the capability of presenting as much pertinent information as needed by the scheduler at any point in the scheduling process in order to minimize the frequency a scheduler would have to consult another data source for scheduling decisions. Ultimately, all scheduling decisions will be made with only the use of the graphics displays and no other source. Therefore, the design of the displays is very important in that they must be complete in their presentation of information, flexible to handle all types of scheduling requests, and easily understandable to a wide range of possible users.

If the graphics are to be informationally complete, what exactly should be considered for display? Airlift scheduling follows, in time, the activities of all aircraft, crews, and ground support, while also considering the station, aircraft, crew, maintenance, and mission performance constraints. In the dynamic overload scenario, aircraft will be the most precious resource to be scheduled. Hence, graphics displays should concentrate on following the activities of the aircraft as a primary concern. The other constraints and activities are not neglected, however, they are simply displayed in a secondary or less obvious manner and only highlighted when a constraint has been violated.

The two basic sets of displays proposed are the "Station Activity Displays" and "Aircraft Mission Sequence Displays".

The first follows primarily aircraft activity at a given station while the second follows a given aircraft fleet or subfleet in time. The ability to emphasize certain aspects of the schedule through display design is a powerful tool in speeding up the decision making process.

The obvious major tradeoff in designing graphics displays occurs between the amount of information displayed on the screen versus the display's readability. The more information that we are able to call up on one display the less likely the scheduler will have to call another display for more information. However, as more information is placed on the screen, the screen becomes very cluttered and eventually nearly unreadable. A comfortable balance can mean an increase in user scheduling efficiency over the imbalanced cases. The scheduler must be able to control the amount of information, deleting information as it becomes unnecessary.

The human/computer communication interface is critical to the overall interactive scheduling system. Thus, the graphics displays must be as comfortable as possible to work with for the user. Along with the informantion readability versus clutter tradeoff, the system should be able to conform to the tastes of the individual user. The set of graphics displays should offer the user a wide range (ways to represent the same information) and varying degrees of information density (clutter). Specific display options will be discussed later in this chapter, but the availability of these options to the user and the ability of the system to handle such flexibility is an important requirement in the final system design.

3.2 Station Activity Display

The station activity display is the basic representation of a schedule. It follows the activities at one or more specified stations over time. The level of activity at a station can be easily viewed using this type of display. Not only does this display help the scheduler identify and correct ground constraint violations, it also is useful to the ground crew at the specific station in that it is a preview of their scheduled workload.

The station activity display is set up in the following way. The most obvious structure (which represents the most important scheduling dimension, time) is a "timeline" representing a given station for a given period of time. Station activities represented on the display are placed at the position down the "timeline" appropriate to their event time. The amount of time shown on one display screen can vary from a few days down to a few hours. The scheduler must request both a station name (identifier) and a band of time to identify a desired screen display.

The schedule database assembles specific events at a station on their event-time. Events can be aircraft departures, arrivals, begin/end of loading/unloading, K-loader availability, crew availability and much more. Attempting to represent all events on the station's timeline would cause an unreadable amount of display clutter. The most basic format displays only aircraft arrivals and departures and their appropriate mission number and aircraft type (Figure 3.2.1).

The basic station format must also be able to inform the scheduler of a violated station constraint or a constraint at its maximum capacity. Typical station related constraints are the"maximum on ground" problem (MOG), the maximum number of K-loaders in use problem and a fuel availability problem. The display must highlight any violation of capacitated situations on the display screen at the appropriate position along the stations timeline. For example on Figure 3.2.2 shaded areas were used to identify MOG and maximum K-loader situations. This is obvious to the scheduler who now knows to avoid scheduling any more aircraft in those time bands at the specific station, or who may wish to rearrange activities at the station to relieve or reduce the full capacity situation.

Any representation on the graphics display whose purpose is to alert the scheduler's attention to a situation where a constraint at or near violation is called a "flag". As the scheduling system is developed in more detail, more constraint test and hence more "flags" will be required.

FIGURE 3.2.1 **Basic Station Activity Display Format**

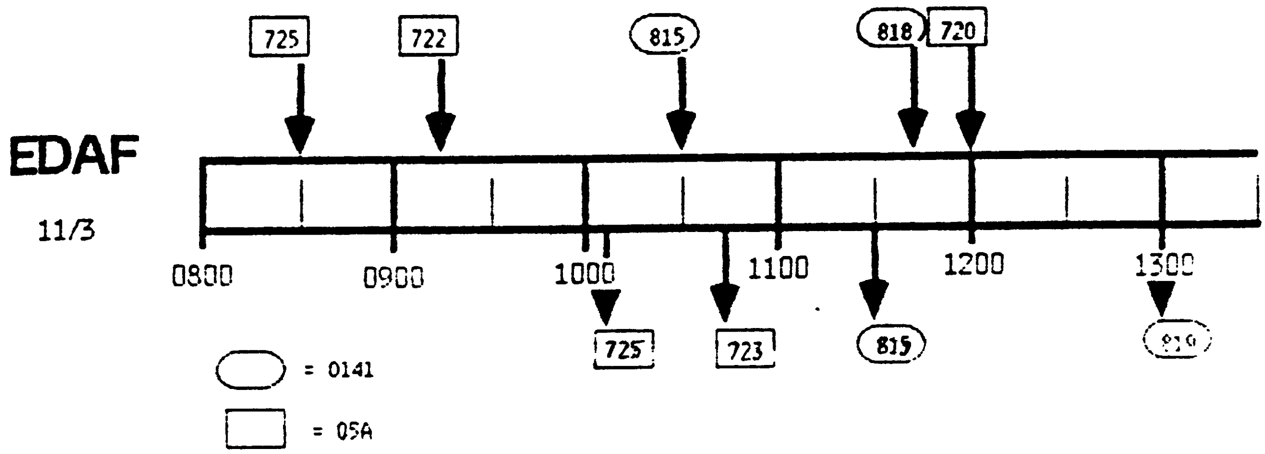
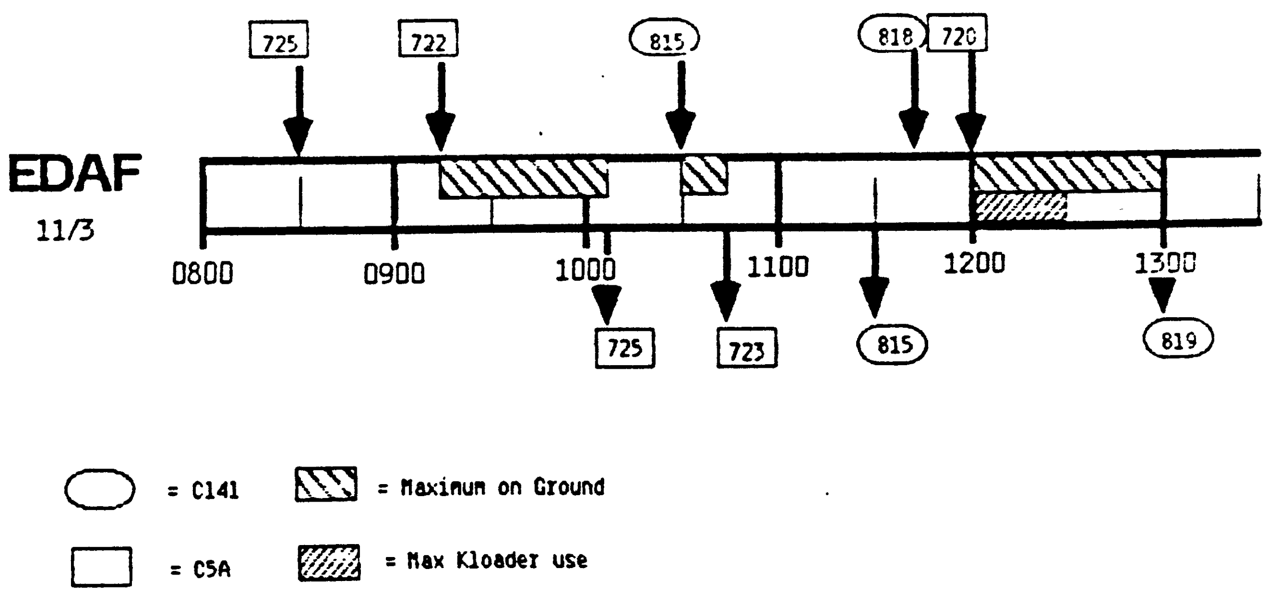


FIGURE 3.2.2 **Station Activity Display Showing Constraints**

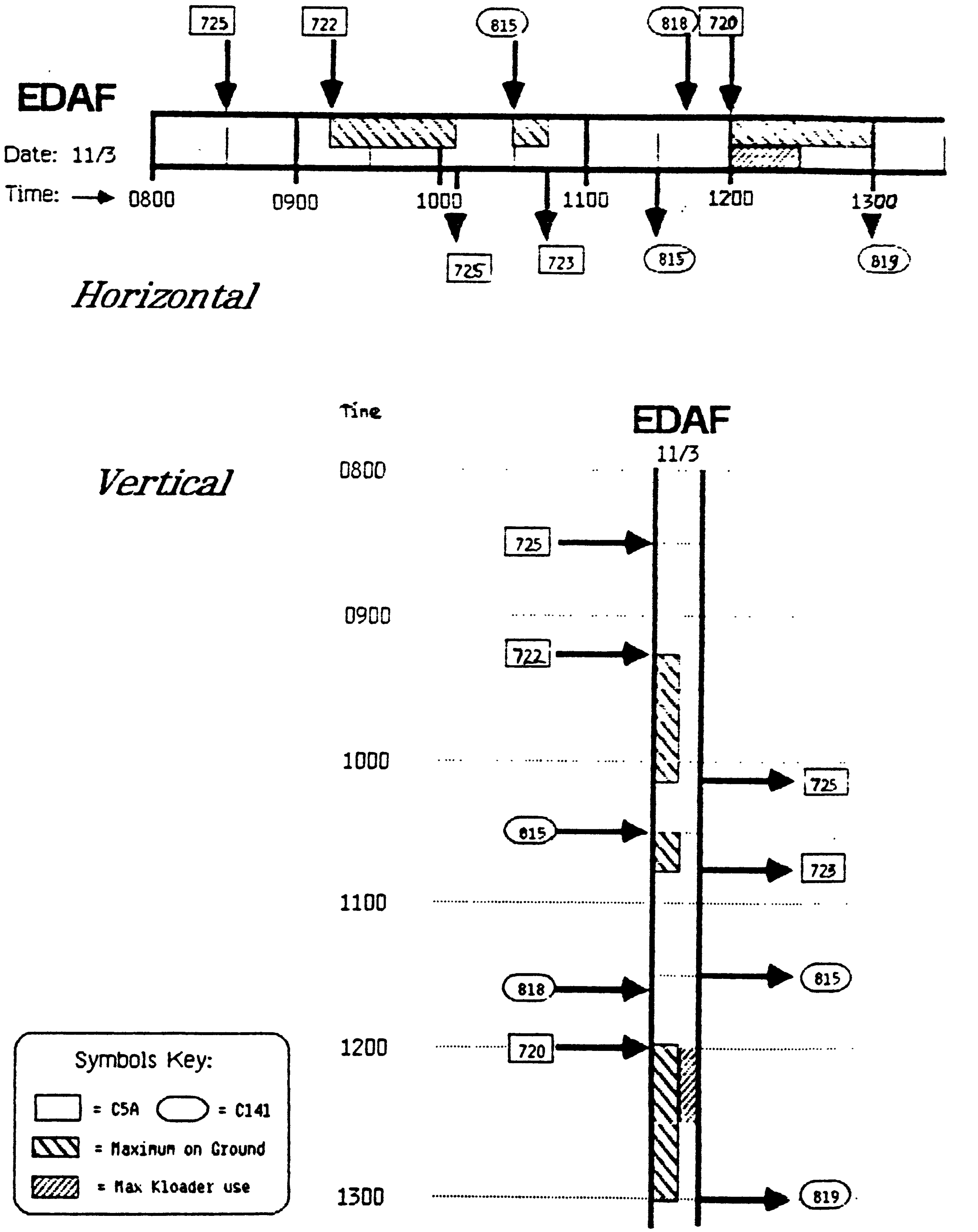


Flags break down into four groups: aircraft constraint flags, station constraint flags, mission/payload constraint flags and crew constraint flags. These flags should be immediately and automatically added to the displays when the scheduler violates a constraint. With the multiplicity of specific types of flags, each flag must be not only obvious to be recognized but also descriptive enough for the scheduler to have an immediate idea of the problem. A more detailed description of the four basic flag types will be discussed in chapter 4.

The basic elements for a station activity display have been outlined but this does not constrain the display into a specific layout. The requirement to design the scheduling system for a wide range of users requires a flexibility in display format. For example, the prominent station timeline is essential to the station activity display format, however, it may be aligned horizontally or vertically on the screen (Figure 3.2.3). Specific symbols for aircraft type, an arrival, a departure, etc., may also take on various formats. Flags may also be highlighted by a computer generated tone or blinking screen symbols as an option. The individual scheduler may be able to select the display format. This flexibility should assist in maximizing scheduling efficiency.

Different formats of station activity displays must be available to the scheduler to assist in the various steps

FIGURE 3.2.3 **Station Activity Display Alignment Formats**



of the scheduling process. A scheduler may begin with a request to add a new mission into a pair of stations. He will first want to get an overview on the day's activity at both stations. This requires a station activity display which shows a wide time period of over ten hours around the planned departure and arrival times (see Figure 3.2.4). From this display, the scheduler decides on a smaller band on which to concentrate and desires more specific information. A smaller time band, 2-4 hours in width, is then selected for display (Figure 3.2.5) and the scheduler inserts the new mission (891). For even more detail of station activity the scheduler may selectto view a collection of specific resources in use at the station (Figure 3.2.6). This multiple format set of station charts addresses the problem of the information density versus readability tradeoff. A wide time band display can show less schedule detail, while a short time band display can show more without looking cluttered. Using his options to add or delete additional schedule information from the basic format, a scheduler can control the level of clutter or readability of his specific screen display.

A significant format option for creating aircraft mission sequences is the ability to display multiple stations on a screen. A scheduler who wishes to schedule a flight between stations A and B or wishes to reroute existing flights and mission sequences would like to view simul-

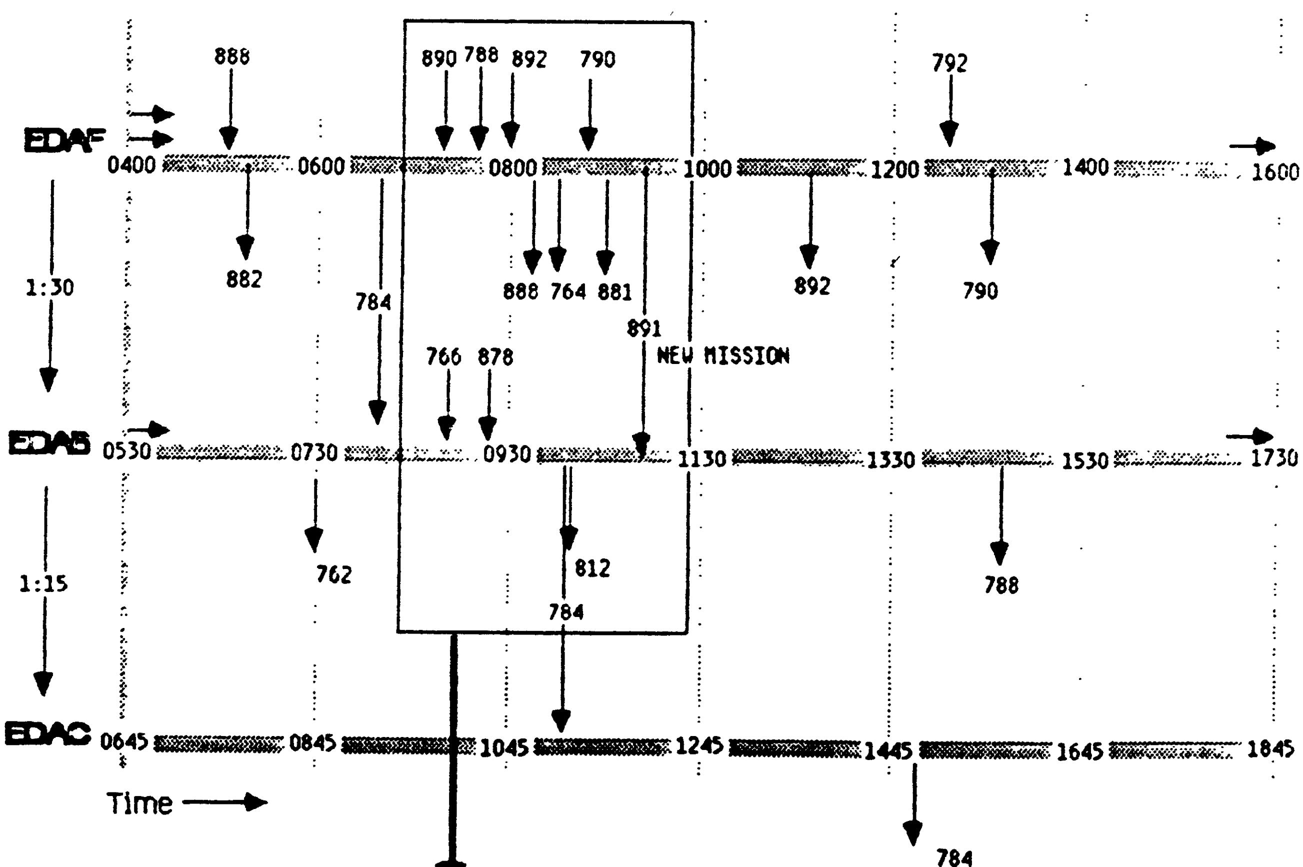
FIGURE 3.2.4

Wide Time Band Station Displays

DATE 11/3/83

Station/ Time Band Shown

EDAF (0400Z - 1600Z)
EDAB (0530Z - 1730Z)
EDAC (0645Z - 1845Z)



Select this portion
of Station Chart for
more detailed view

Figure 3.2.5

Station Time Lines
are offset by standard
flying times

FIGURE 3.2.5

Narrower Time Band Station Schedules

(Magnification of box selected in Figure 3.2.4)

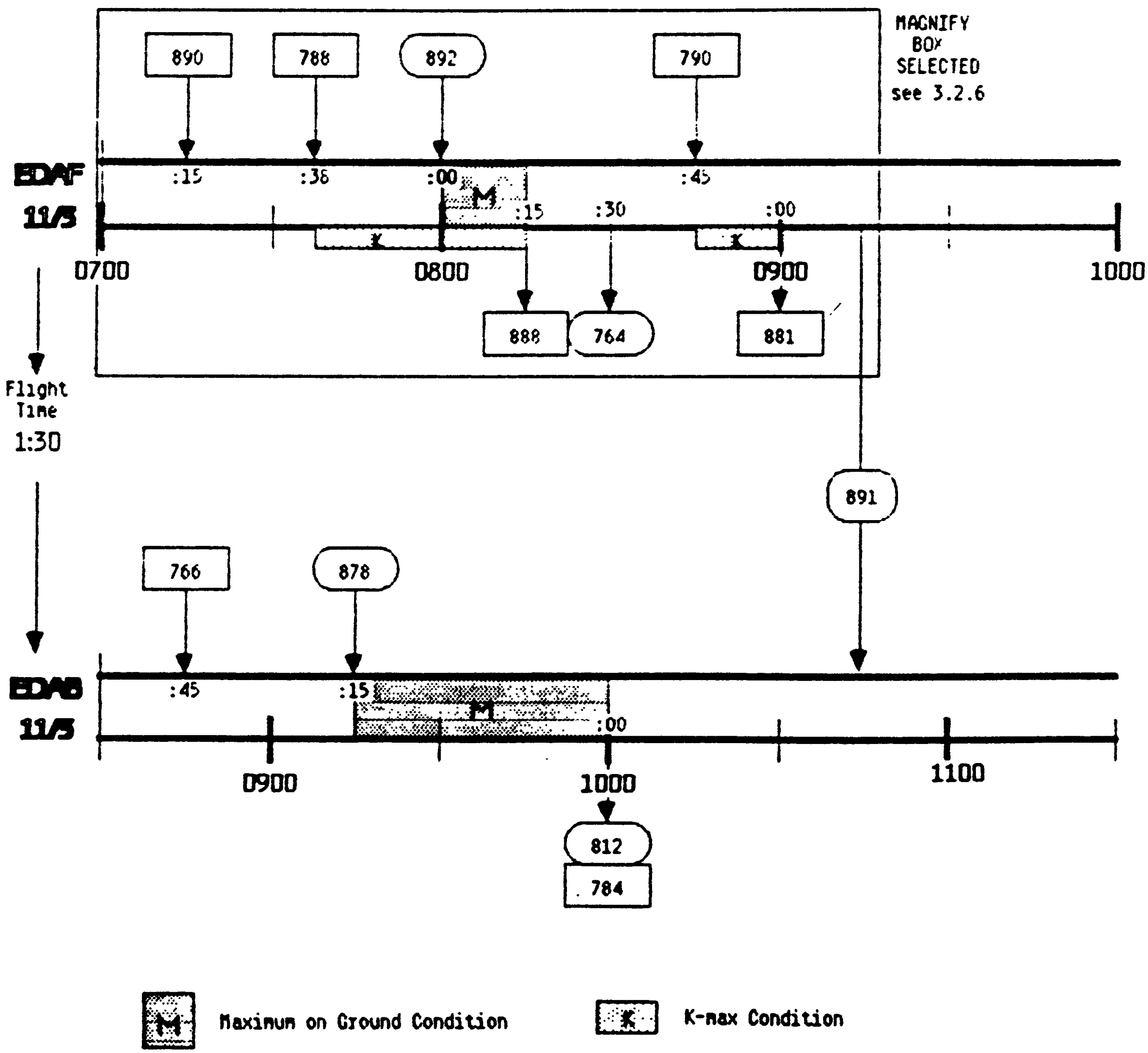
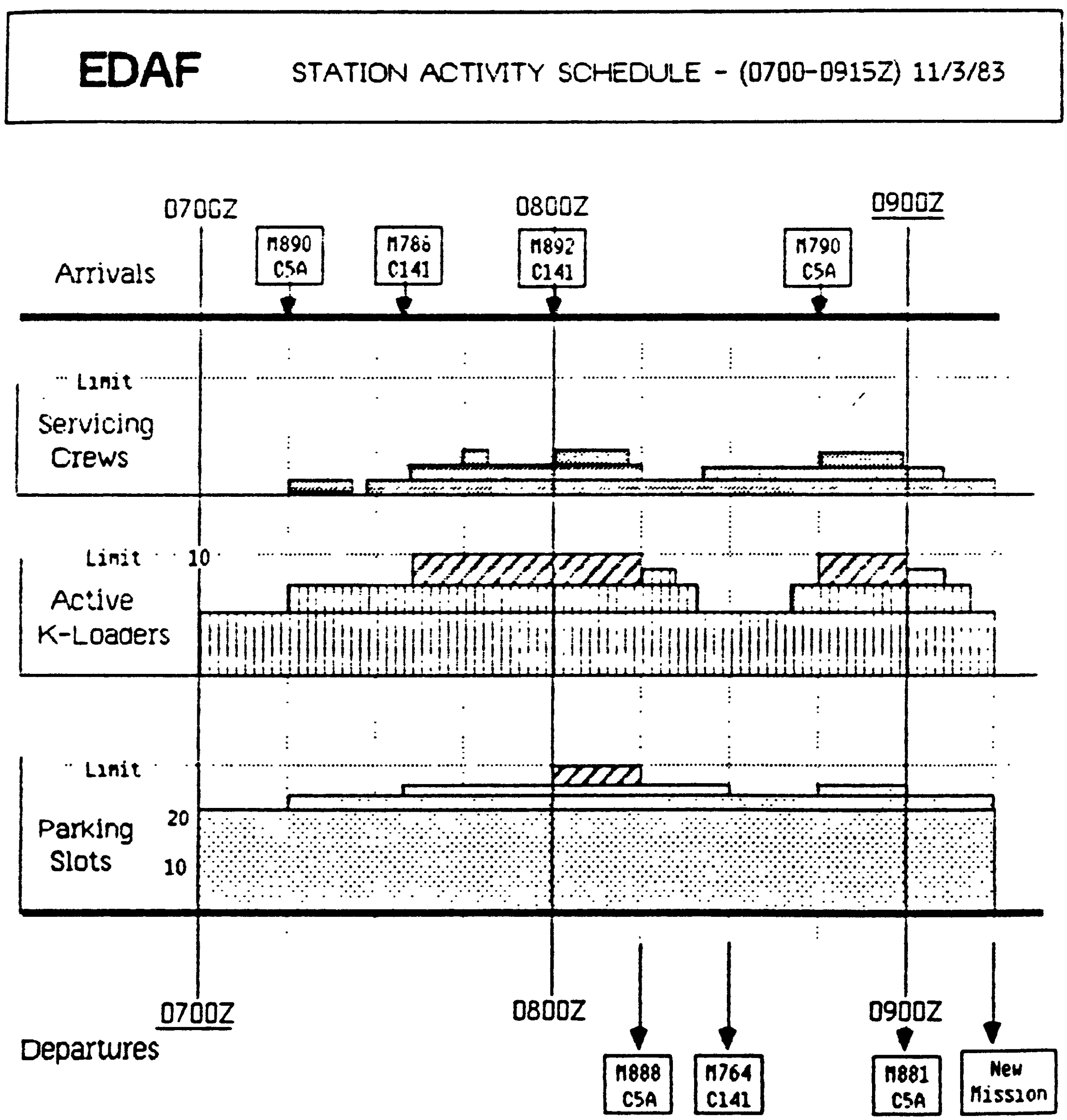


FIGURE 3.2.6

DETAILED SCHEDULE OF STATION OPERATIONS



taneously those origin and destination stations of the flight leg for review of their constraints with the time axis offset by the segment flying time (Figure 3.2.7). Thus, a horizontal straight line out of station A would be an arrival exactly on time at B. The possible departure and arrival times with a given window would become immediately obvious. Even without the optional flying time offset, the side-by-side time lines on the screen layout speeds aircraft routing and scheduling.

Two or three stations can fit on a standard screen... but what if more stations are desired by the scheduler? Present graphics allow a computer to prepare an electronic picture of a desired display over a hundred times larger than the size of the screen limits. Therefore, several stations can be recorded on a large electronic picture and viewed a section at a time. Figure 3.2.8 shows an illustration of this concept. Manipulation of the displayed portion of picture is very quick with present built-in computer software. A multistation electronic picture is useful to a scheduler when he wishes to consider a station constraint when scheduling a multi-segment mission, or in creating new aircraft mission sequences.

Of course, the alternative is to redraw the stations which the scheduler wishes to examine simultaneously i.e. instead of sliding the window to another station, he can select those few he wishes to view and change his selection back and forth.

FIGURE 3.2.7

Multistation Display with Time Windows Highlighted

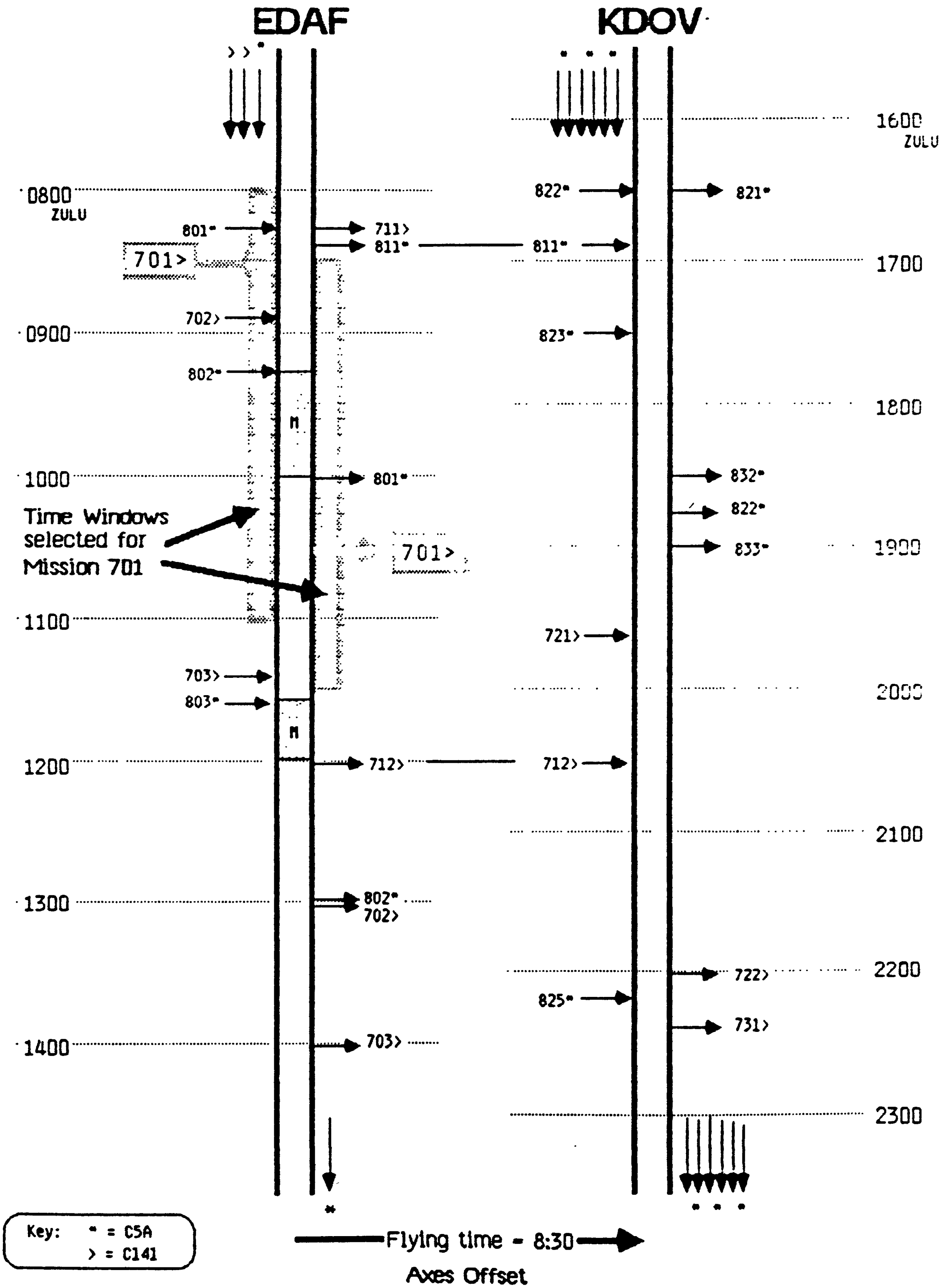
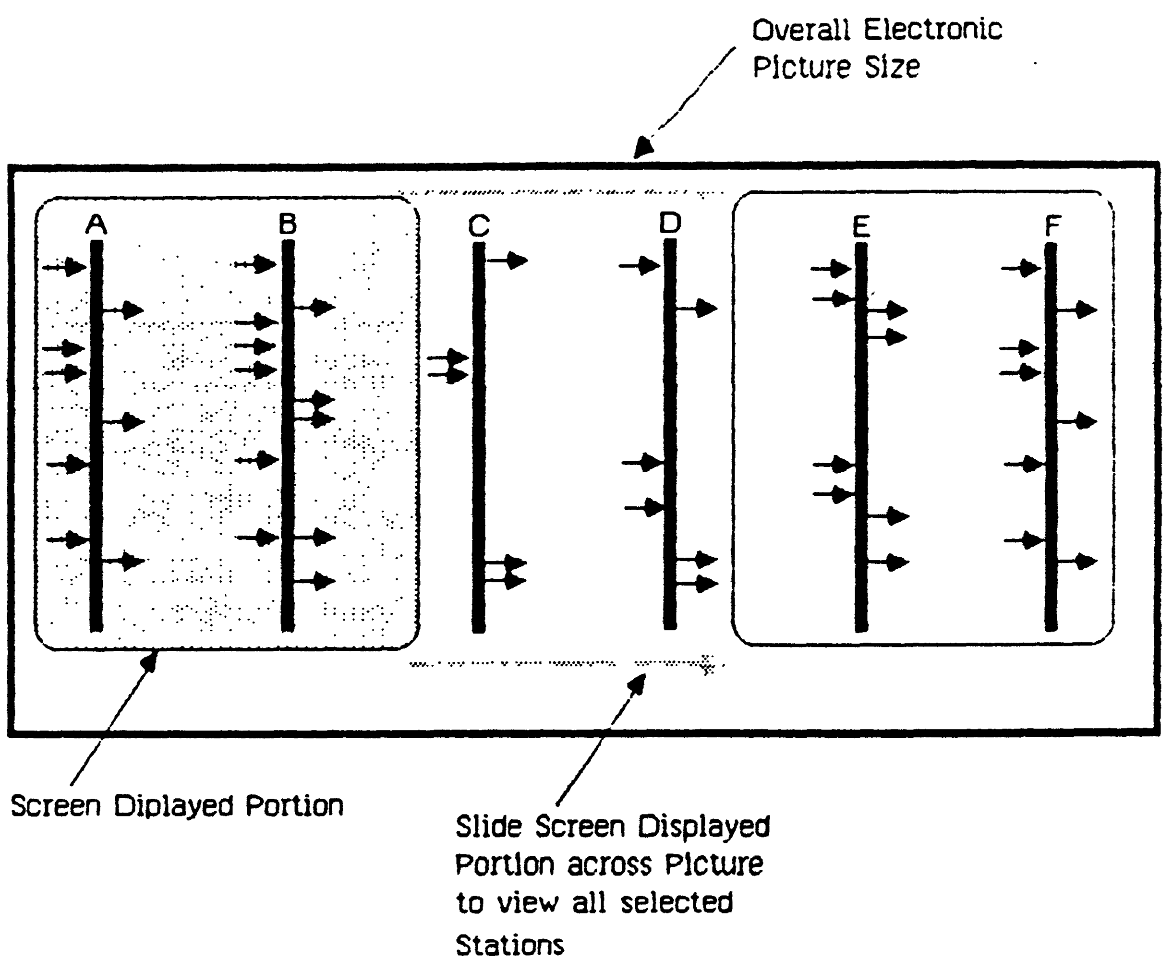


FIGURE 3.2.8 **Electronic Picture of a Multistation Station Display**



The station display format of representing a schedule displays the activities of a station and highlights any station constraints violated by the schedule present in the database. As in all graphics displays, there is a trade-off between the amount of information listed with the display and the readability of the display. Combining two or more station displays on the same display is very necessary for the routing and scheduling of aircraft and creating mission sequences. A set of station displays covering different levels of detail and time window widths can assist in the wide variety of steps in the mission scheduling process.

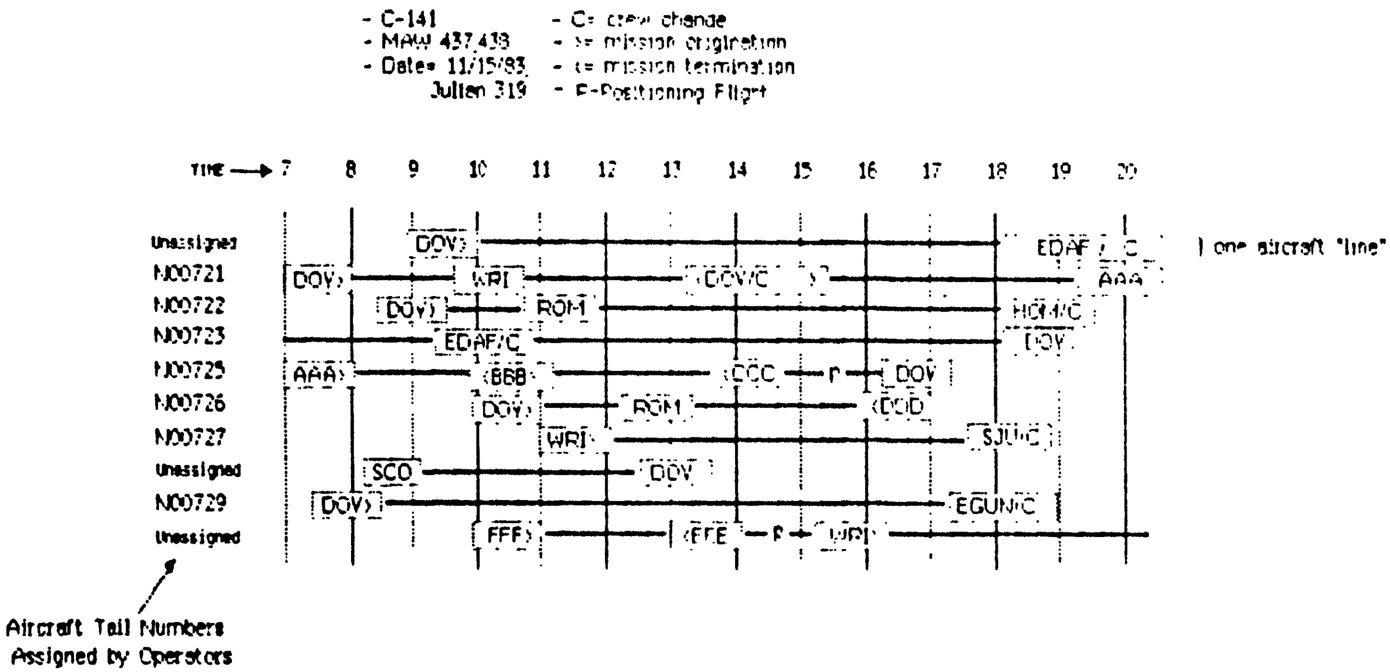
3.3 Aircraft Mission Sequence Display

This display is the second major type of schedule representation. The concept is similar to the station in that it follows activities in time but differs in that it follows the aircraft's activities rather than the station activities. Each aircraft, instead of each station, has a unique time axis "line" along which the aircraft activities are represented. This display is especially helpful for sequencing the activity which will eventually be assigned to a specific aircraft tail number, recognizing positioning legs, and visualizing aircraft and crew constraint violations.

The Aircraft Mission Sequence Display, as does the Station Activity Display, utilizes one dimension of the screen as the direction of time. In most cases the passage of time will be aligned horizontally. For each aircraft there is a single associated time line. The line initially corresponds to a generic aircraft (e.g. C141, C5A), and later will be assigned a tail number by the operators. In Figure 3.3.1 we see a basic view of the mission sequence display with a number of aircraft lines and with time increasing to the right. The solid lines, known as flying "bars", indicate the time when the aircraft is flying a segment. Positioning leg segments are denoted by a small "p" in the flying "bar". While bars indicate when the aircraft is inflight, the spaces denote times when the aircraft is on the ground. The station name being visited by the aircraft is found in the ground time space of the aircraft line (in between flying time bars). Also in this area are symbols helpful to the scheduler. To mark the beginning and end of a specific mission the symbols ">" and "<" are used. This helps the scheduler identify the sequencing of missions. A "/C" after the station identifier indicates a crew change. There should be a flexible format of a variety of similar notations available to the scheduler.

On the basic display aircraft lines are purposely designed to be thinner than the station display's time axis

FIGURE 3.3.1 **Aircraft Mission Sequence Display**



line in order to see more aircraft per screen. In the close pitched format, at least 15 aircraft lines can be displayed on the current LISA computer graphics with an appropriate heading. As discussed in the station display formats, an electronic picture can be prepared which can be much larger than the screen can display. For scheduling subfleets of more than 15 this concept must be used. Also the internal electronic picture of the Mission Sequence chart may be much longer in time than that portion displayed (see Figure 3.3.2). The time spacing on the aircraft lines can vary but for the horizontal mode cannot be less than 2hrs/inch and still maintain reasonable readability. The basic mission sequence chart also contains a complete set of flags similar to the station activity display which alert the scheduler of a constraint violation, its time, and appropriate aircraft affected.

In addition to the basic Aircraft Mission Sequence Display, the scheduler may want more specific information of a particular aircraft's activities. He could select a more detailed format called the "Detailed Mission Sequence Display", shown by Figure 3.3.3. It consists of an appropriate section of the aircraft mission sequence display showing a single aircraft mission sequence. The positioning leg or mission identifiers are printed above each flying bar. The exact same notation is used for the regular aircraft mission sequence display flying line itself. Specific

FIGURE 3.3.2 LARGE MISSION SEQUENCE DISPLAY

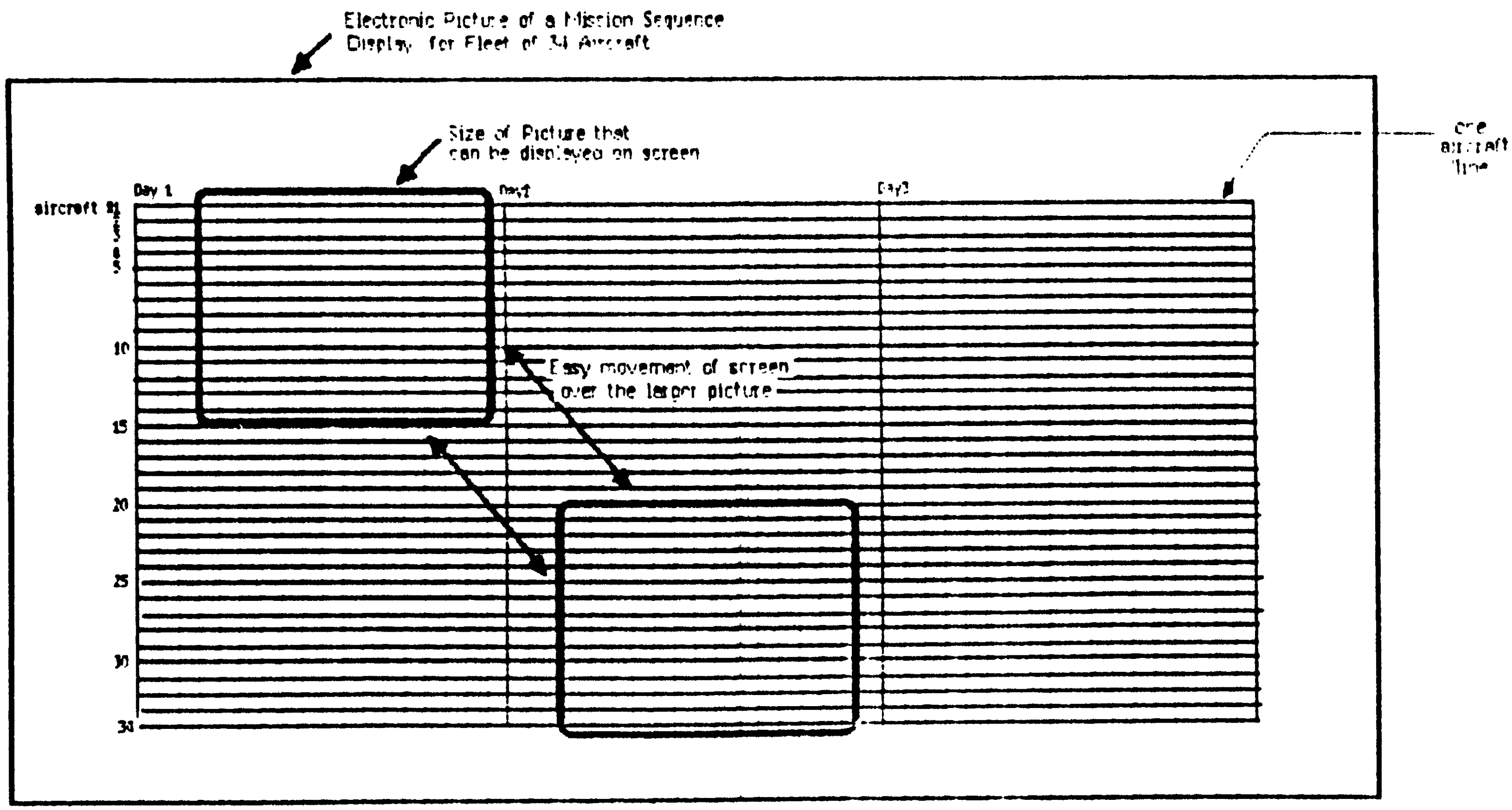
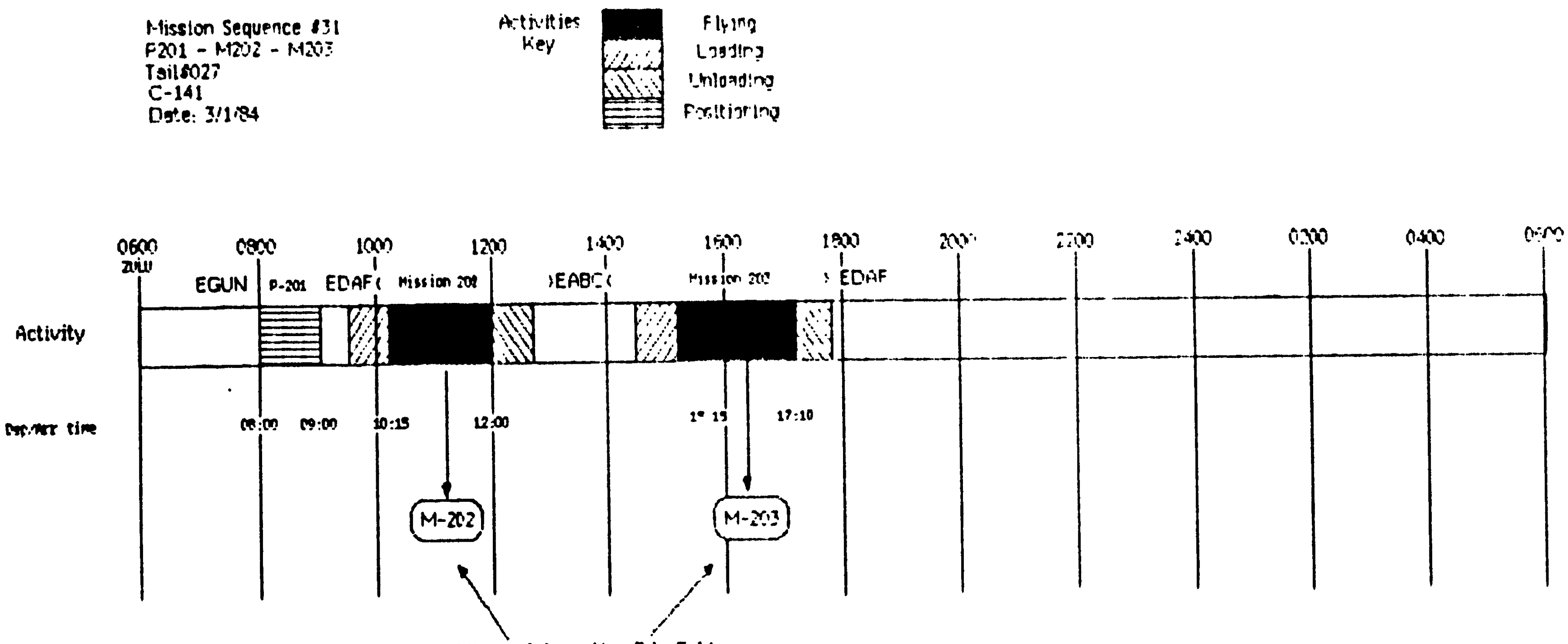


FIGURE 3.3.3 Detailed Mission Sequence Display



Mission Information File Folders:
if there are any special notes, memoranda added by the scheduler it can be displayed by selecting the folder. No folder if there are no special notes.

departure and arrival times are listed for each aircraft flight segment. Four basic aircraft activities are shaded at the appropriate scheduled times (flying, loading, unloading, positioning leg). Periods of time when the aircraft is not occupied by one of the four activities are left unshaded on the aircraft activities line. A descriptive heading completes all the basic information necessary for scheduling with this display. This format is thus far more detailed than the basic display. The tradeoff is that only one or two aircraft can be displayed on the screen at one time in this easily readable form.

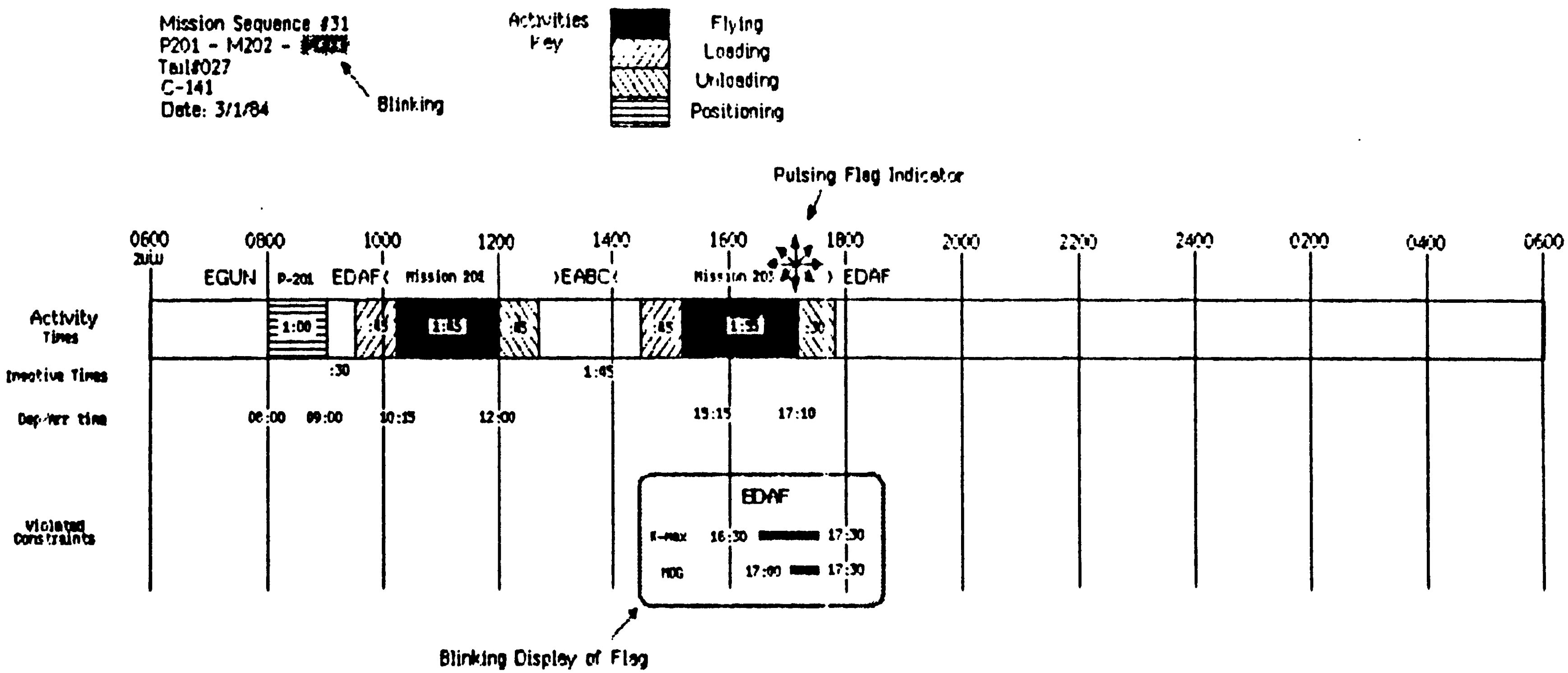
A wide range of optional information can be added to the Mission Sequence Display. Figure 3.3.4 shows the additions of the specific times for the begin and end of loading and unloading. This information may be essential to the scheduler if he is worried about K-loader scheduling at a station. Figure 3.3.5 adds the times of the active and inactive period of the aircraft. This display also shows an example of a flag. The star above the activities line is placed at the time of the computer recognized violated constraint. The flag description is found below the chart. This specific flag is alerting the scheduler that there is a MOG at EDAF between 1700 and 1730 and a K-max from 1630 to 1730. Thus the 1710 arrival of Mission 203 must be rescheduled. Missions may have mission windows in which they can be scheduled and a display of these windows is an appro-

Room 14-0551
77 Massachusetts Avenue
Cambridge, MA 02139
Ph: 617.253.5668 Fax: 617.253.1690
Email: docs@mit.edu
http://libraries.mit.edu/docs

Page 54 IS NOT missing. This is a page mis-numbering error by the author.

FIGURE 3.3.5

Activities Times and Flag Displayed

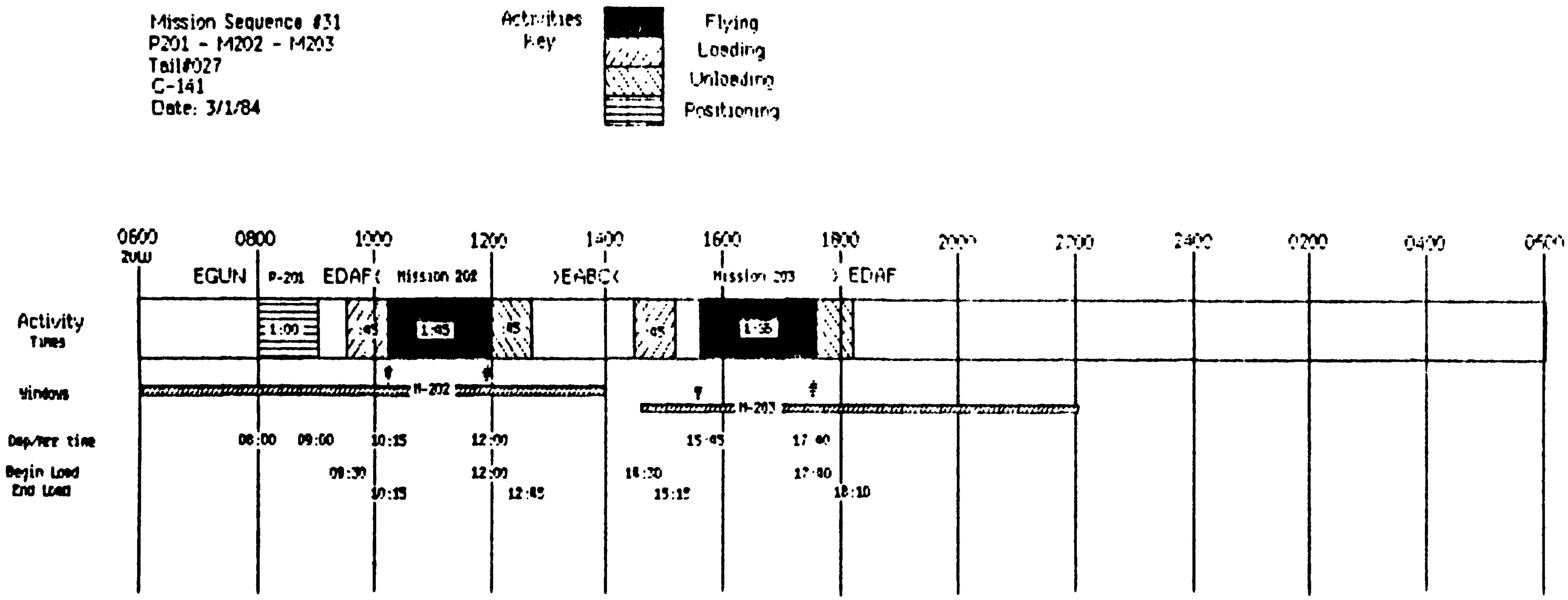


priate option for the mission sequence display (Figure 3.3.6). Mission windows and mission sequences are visualized more clearly in this format than on the Station Activity Display format.

The combination of the basic and the detailed mission sequence displays, is similar in purpose to the proposed set of station displays. It is to satisfy the scheduler's need to have both a reasonable overview of events, but at the same time have the ability to view the most minor detail accurately. The tradeoff between the amount of information and the readability of the screen requires the multiple format of the charts. The availability of optional additional information adding to the format by user request and the options in overall display alignment help each scheduler tailor the screen display to his own liking.

The Mission Sequence format of schedule representation follows the activities of the aircraft. Mission sequencing and multisegment missions are more easily represented in this format, although it is difficult to perform the routing and resequencing of missions which occur at stations. Scheduling crew Mission Sequences can be done using this display since crew generally follow aircraft. The set up of the display makes obvious the times when an aircraft is inactive. Large sections of the aircraft line without any flying time bars would indicate to the scheduler that that time band may be used to insert a new mission. The improper scheduling of

FIGURE 3.3.6 Detailed Mission Sequence Display with Time Windows



aircraft resources are obvious on the aircraft routing display. If one aircraft were scheduled to two missions simultaneously not only would the proper flags alert the scheduler but the aircraft mission sequence display would show segment overlap and clutter. If a specific scheduling cell had 10 aircraft, currently at their disposal and saw 11 aircraft lines on the aircraft mission sequence display, the scheduler would recognize the problem of inadequate available fleet. Just as the station display illustrates the activities at a station, mission sequence display represents a readable display of the activities of a fleet of aircraft.

3.4 Display Combinations and Additional Displays

Both the mission and station displays have been shown to be excellent displays in representing a schedule and are in common use by airline schedulers. Each alone could be used to schedule a fleet of aircraft but the use of them together would be even more helpful. First the scheduler would call up a station display to view activities at the origin and destination and would insert the new mission which may cause rerouting of aircraft. Continuing to a more detailed level, the scheduler could return to the mission sequence display for the final exact mission scheduling. The use of a combination of displays will speed the scheduling process but will also add a new requirement to the system that must be addressed.

An interactive scheduling system is only useful to the scheduler if the response time of the computer is relatively fast. Thus if a scheduler wishes to use a number of different displays to assist him in scheduling decision support, each must be able to be called up as quickly as possible. If it takes one minute for the computer to respond and display a requested station format the scheduler will become impatient with the system and may revert back to the old wall charts and grease pencils. The displays must be quickly generated or modified by the system. The Apple LISA computer features a desktop screen format. More than one electronic picture of a screen display can be held in local memory and can be moved on the screen like pages of paper over a desktop. One display can be placed on top of the other like pages, and any page can be brought instantly to the top of the desktop to be viewed. This paging technique will allow the system to handle frequent display changes at a speed necessary for a quality interactive system and avoids the need for a multiple display work station.

With the desktop paging technique available, the number of display designs does not have to be limited to solely station or mission displays. Additional minidisplays can be presented as assistant views to these primary displays giving even more detailed background information. specific aircraft, station and mission information would make up three possible new minidisplay structures.

While reviewing a mission or station display, a scheduler may wish to know the exact performance and additional information for a specific aircraft. The aircraft symbol could be selected by a "mousing" action (explained in Section 3.5) which automatically signals the system to bring up a minidisplay containing specific information such as: aircraft weight, range, payload, landing field, or any other data which is maintained by the database (see Figure 3.4.1).

By "mousing" any station, a station information minidisplay would offer specific information about the requested station; location, region, 5 nearest stations, stations within 100 miles, number of K-loaders, number of parking slots, runways, fuel storage etc. (Figure 3.4.2). One example of its use would be when a scheduler wishes to upgrade a mission to a larger aircraft type. He could call up the affected station's information displays and confirm if the runways were long enough or the parking facilities would accommodate the larger aircraft.

Similarly, by mousing a mission, a mission information minidisplay would inform the scheduler in great detail concerning each mission. This would include the mission number, identifier, routing, payload, time window, priority level and more. Figure 3.4.3 shows a typical layout for the mission information display, and includes mousing a "folder",

FIGURE 3A.1 **Aircraft Information Overlay**

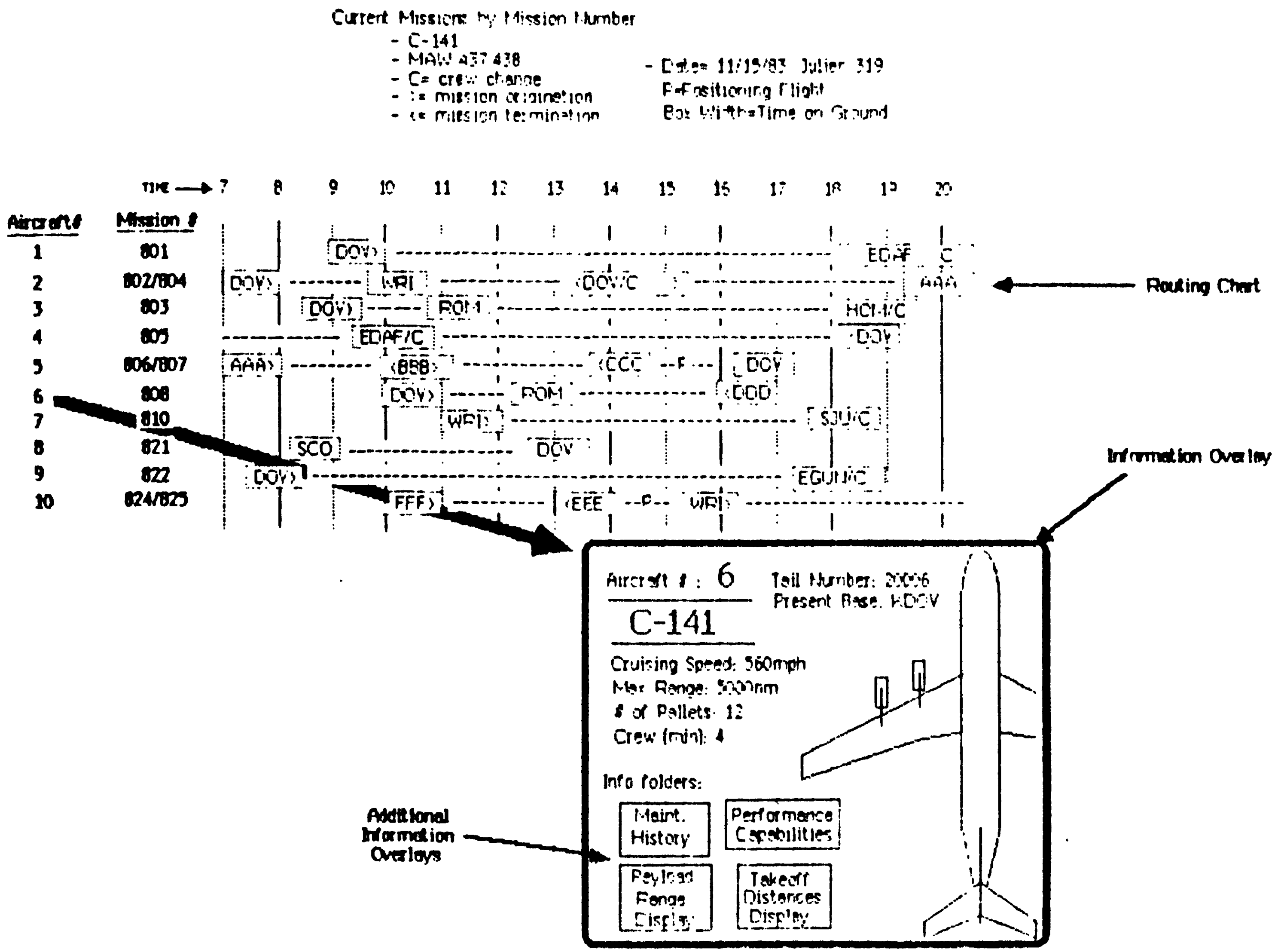


FIGURE 3.4.2

AIRCRAFT ROUTING CHART
WITH STATION INFORMATION OVERLAY

Current Missions by Mission Number
- C-141
- MAW 437,438
- C= crew change
- >= mission origination
- <= mission termination
- Date= 11/15/83, Julian 319

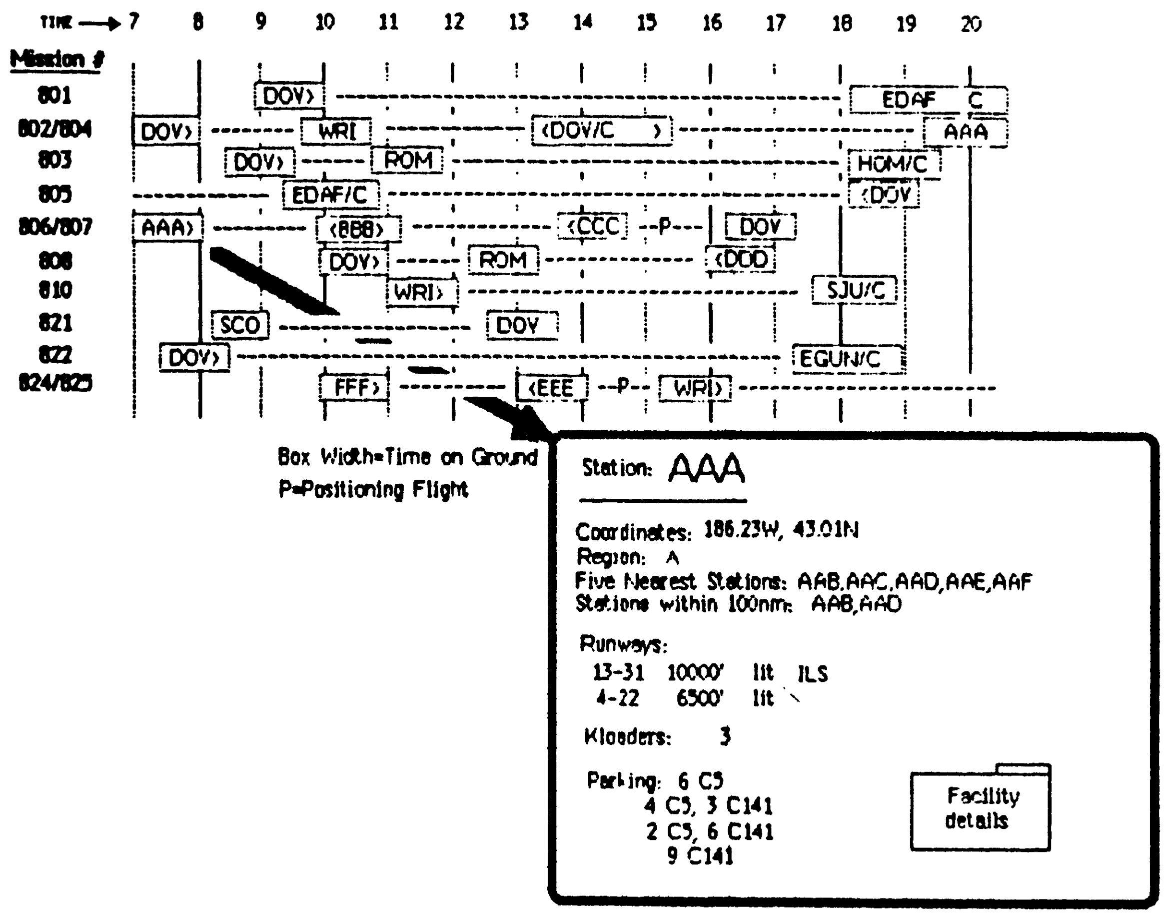
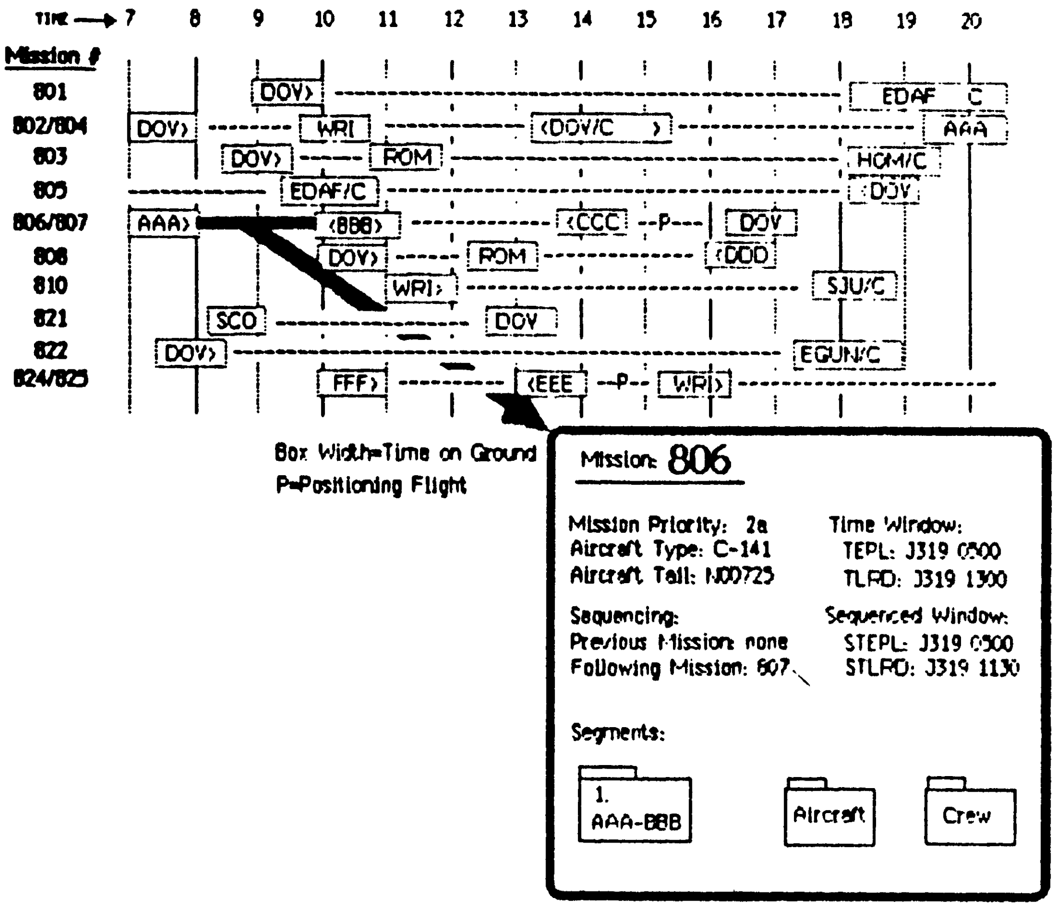


FIGURE 3.4.3

AIRCRAFT ROUTING CHART

WITH MISSION INFORMATION OVERLAY

Current Missions by Mission Number
- C-141
- MAW 437 438
- C= crew change
- >= mission origination
- <= mission termination
- Date= 11/15/83, Julian 319



Box Width=Time on Ground
P=Positioning Flight

much further detailed information on segment, aircraft or crew can be displayed. Figure 3.4.4 shows the Task and Load information in the segment folder.

The use of the aircraft, station and mission information displays as overlays to either the mission or station displays should give the scheduler all the data necessary for the most detailed scheduling decisions. The scheduling system must be not only informative in its wide range of display combinations but it must also be a comfortable tool for all users. One final display format is proposed. As the scheduler is getting used to the interactive system he may wish to revert to a scratchpad to jot down ideas and symbols. This too can be done on the graphics screen. It is simply another display overlay that will accept data just like a scratchpad which could be left as a "folder" at appropriate points nested within the above display structure. After using the system, the scheduler may find he does not need his electronic scratchpad any longer but he may just keep it in use even then. This idea of simulating all possible aspects of the old methods of scheduling electronically with advanced graphics should help new users feel comfortable with the interactive system.

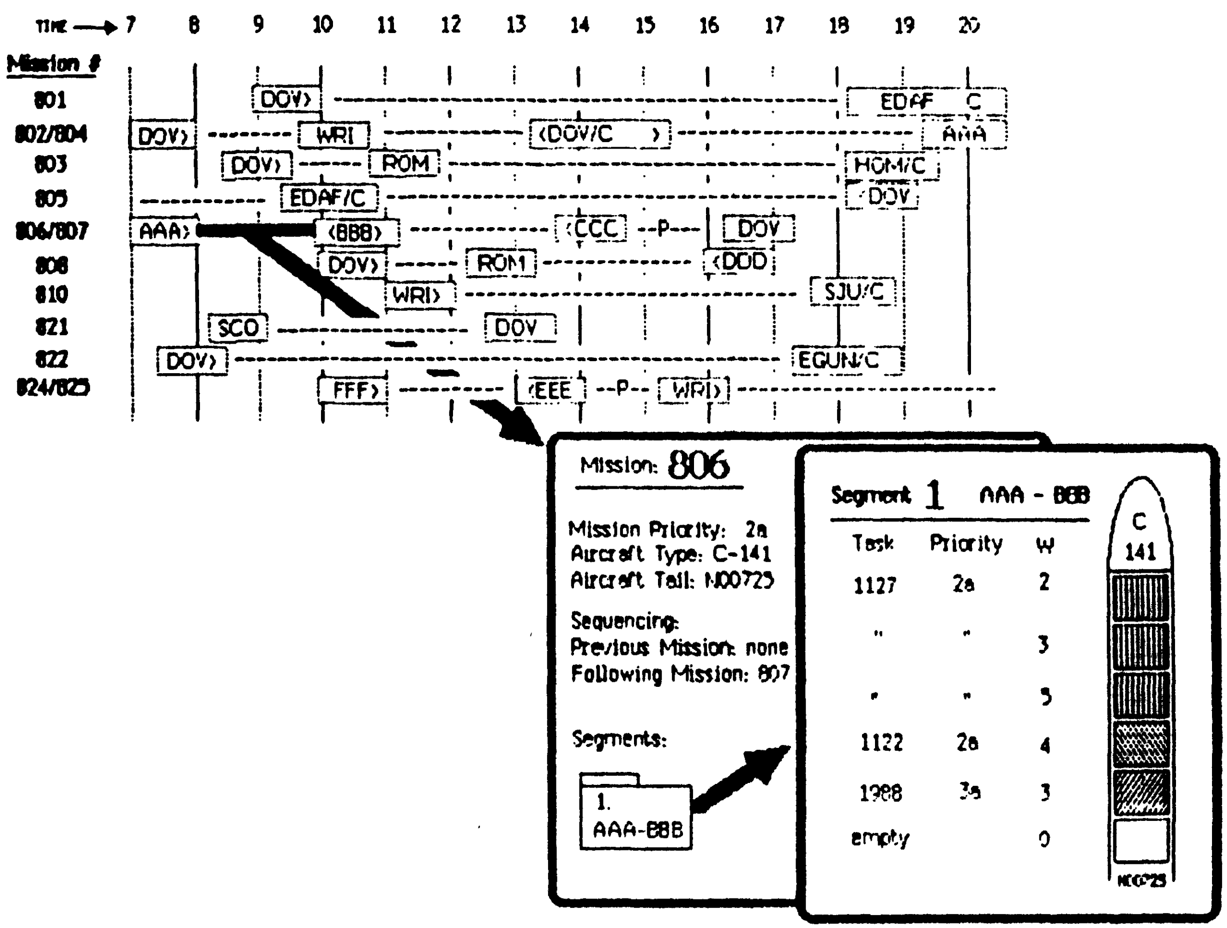
3.5 Graphics Manipulation Tools

The graphic displays on the computer screen are help-

FIGURE 3A4 **Nested Mission Information Overlays**

Current Missions by Mission Number
- C-141
- MAW 437,438
- C= crew change
-)= mission origination
- (= mission termination
- Date= 11/15/83, Julian 319

Box Width=Time on Ground
P=Positioning Flight

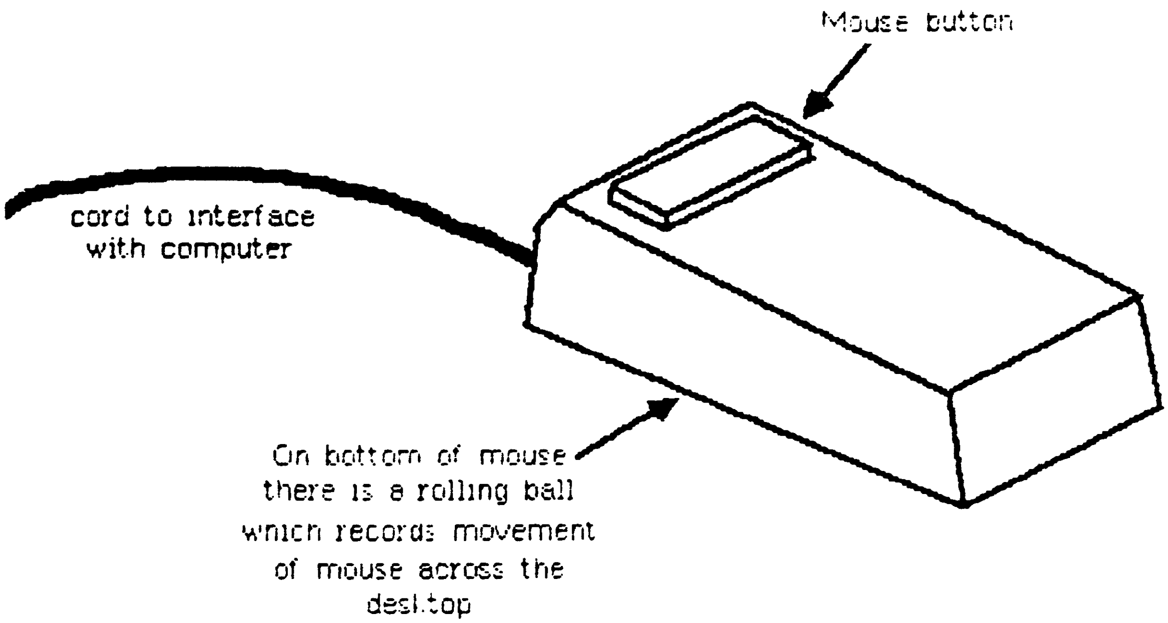


ful, readable representations of the schedule which is stored in the database. Manual scheduling techniques included drawing these displays and from them deciding how to modify the database (schedule) directly. The proposed approach is to have the schedule data indirectly modified. This entails the modification of the graphics displays which automatically update the detailed internal schedule database. Direct modification of the database is not necessary since the modification of the schedule-representative symbols on the screen is interpreted as a command to modify the database.

Since each part of the graphics display now is a representative of a part of the overall schedule, these symbols are entities that must be recognized and acted upon by the system. A way of selecting symbols on the screen is necessary. The "mouse" is the tool used to manipulate the symbols on the screen. The mouse (Figure 3.5.1) consists of a small hand maneuvered device which is rolled across a desktop. As the mouse is moved across the desktop, concurrently, an arrow-shaped (Figure 3.5.2) position indicator is moved across the screen. The mouse unit includes a button which, depending on how it is used, selects various locations and modes for the screen.

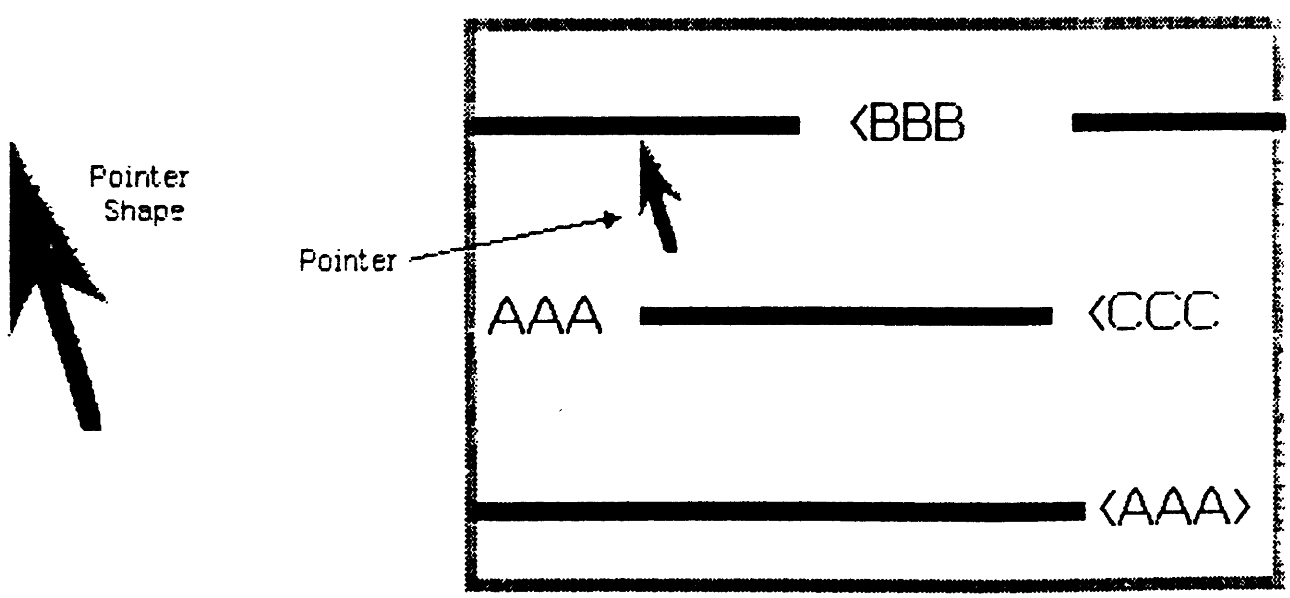
To select a position or acknowledge a symbol on the screen, the mouse is moved across the desktop until the positioning arrow is over the desired position on the

FIGURE 3.5.1 The Mouse



Arrow-shaped
Position Indicator

FIGURE 3.5.2



Routing Chart Example

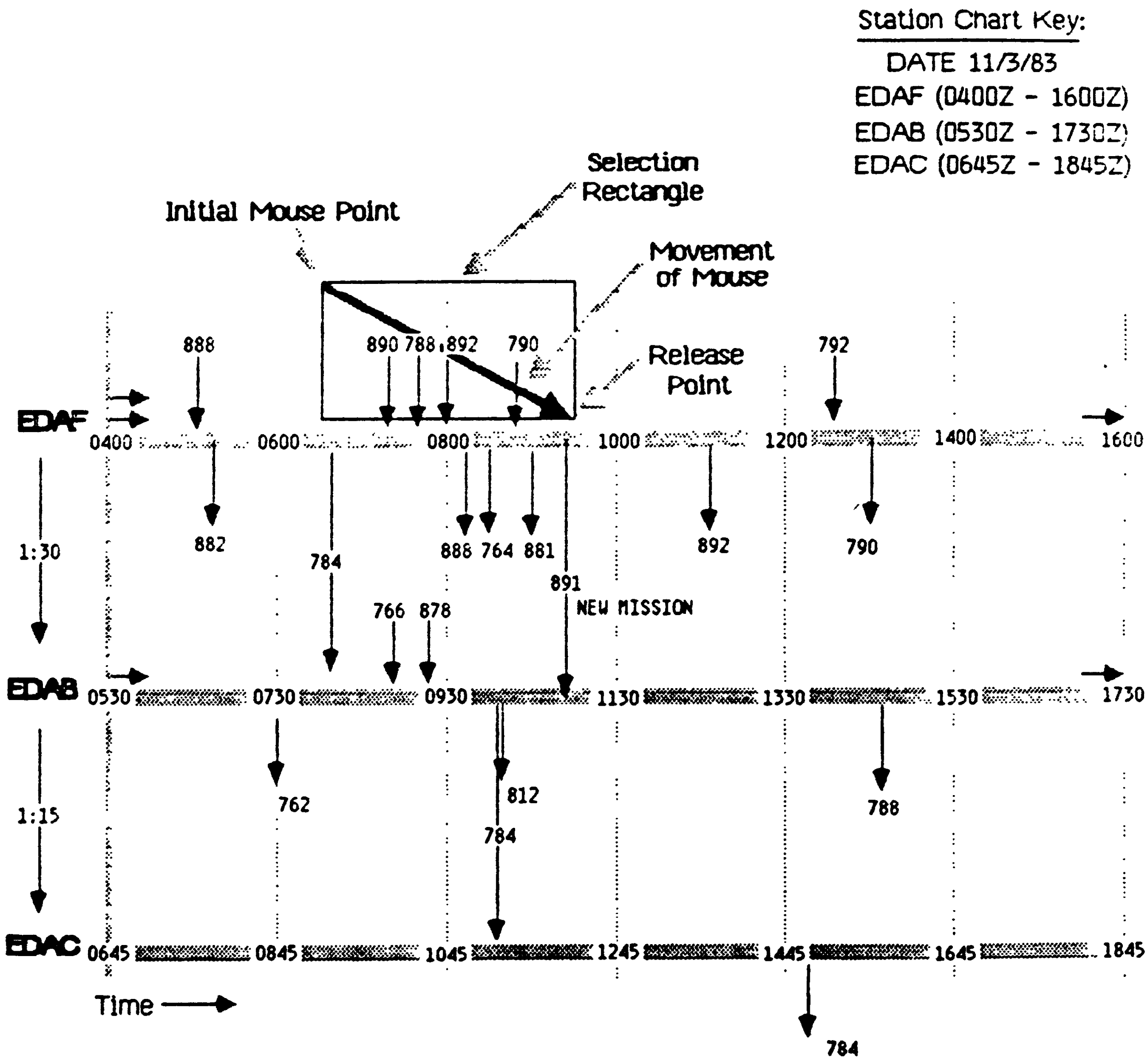
screen and then the button is pressed once and released. Depending on where the moused position lies and the set-up of the screen, many different things may happen. If the moused position has no symbol under it then the system will simply ignore the action as accidental. However, if the mouse falls upon a symbol or acknowledgable region that will become selected. The system will then either wait for a further command or execute the mode which is to be invoked when the specific symbol is selected.

As described in Section 3.4, part of a display may be selected and an information overlay will be added to the screen for that selected symbol. Any symbol that can also double as the command for the creationg of another display is called an "icon". The procedure to "open" an icon is different from acknowledging the symbol. To open an icon one must select the symbol twice in rapid succession. For each overlay display that is added to the screen, there will be a small icon on the overlay which when selected twice removes the overlay from the screen. Thus, the scheduler can add and delete extra scheduling decision support data with a simple set of mouse button commands.

The third basic mouse manipulation action is the "mouse and hold" procedure. In the example of sliding a scheduled segment in time, a scheduler will select a segment on the chart and hold the button down without releasing it. The scheduler has a "hold" on that specific segment symbol. While "holding" the segment, he moves the mouse around the desktop and the system redraws the new segment location continuously as it is moved across the screen with the mouse indicator. When the scheduler has moved the mouse to the position desired as a new schedule for the segment he then releases the mouse button and the selected segment is now moved in time both on the screen and automatically within the schedule database. The mouse and hold procedure can be used for many graphics manipulation operations.

When more than one symbol is to be selected the mouse and hold procedure is used to draw a rectangle around the desired symbols. By selecting a point on the screen where there is no symbol and holding the mouse button while moving diagonally, all symbols in the rectangle will be acknowledged. Figure 3.5.3 shows how the system draws the rectangular selection box on the screen when the mouse is appropriately operated. In addition to selecting an area of symbols, the "mouse and hold" procedure and selection box can be used for many other different operations depending on the system's selected mode. For example, suppose the scheduler wishes to change displays from a station display with a wide time band shown to one with a narrower time band. He must set the system into a display changing mode and then use the mouse and hold procedure to select the small time band which is to be magnified for the next dis-

FIGURE 3.5.3 **Selection Rectangle Generation**

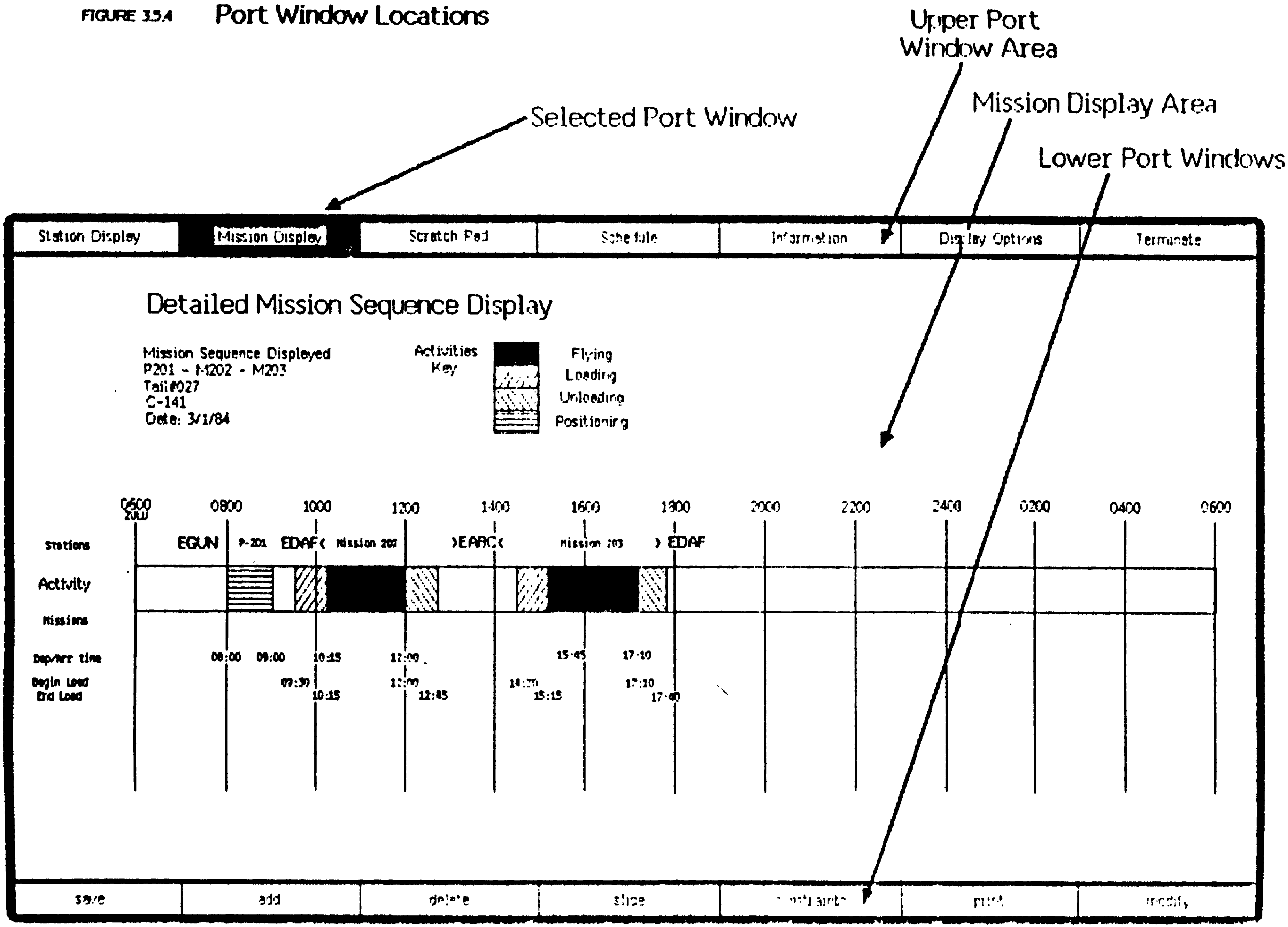


Result: Missions 890,788,892,790 have been selected play.

Selecting and mousing symbols on a schedule display can be accomplished with the mouse. The execution of specific instructions or modes and the selection of options can also be initiated with the mouse but with the added assistance of pulldown port windows. Port windows are a new addition to modern graphics systems which are designed to automatically execute specific instructions when moused. This means that the typing out of program instructions is replaced by a quick click of the mouse over the appropriate port location.

Port windows are found outside the main display usually at the very top or bottom of the screen. Figure 3.5.4 shows these typical locations. In a fully developed interactive dynamic scheduling system there will be a myriad of possible instructions and modes which the scheduler will want to execute. Chapter 4 will discuss these specific operations. However, the quick selection of graphics manipulation and system instructions by using the mouse and port windows is a graphics tool. As more instructions are available on the system more port windows are necessary on the screen. The ports will compete with the main screen display for the critically finite amount of screen area. To alleviate this confrontation, the idea of "windows" is introduced. The ports may carry selectable boxes which represent a set of instructions. When the port

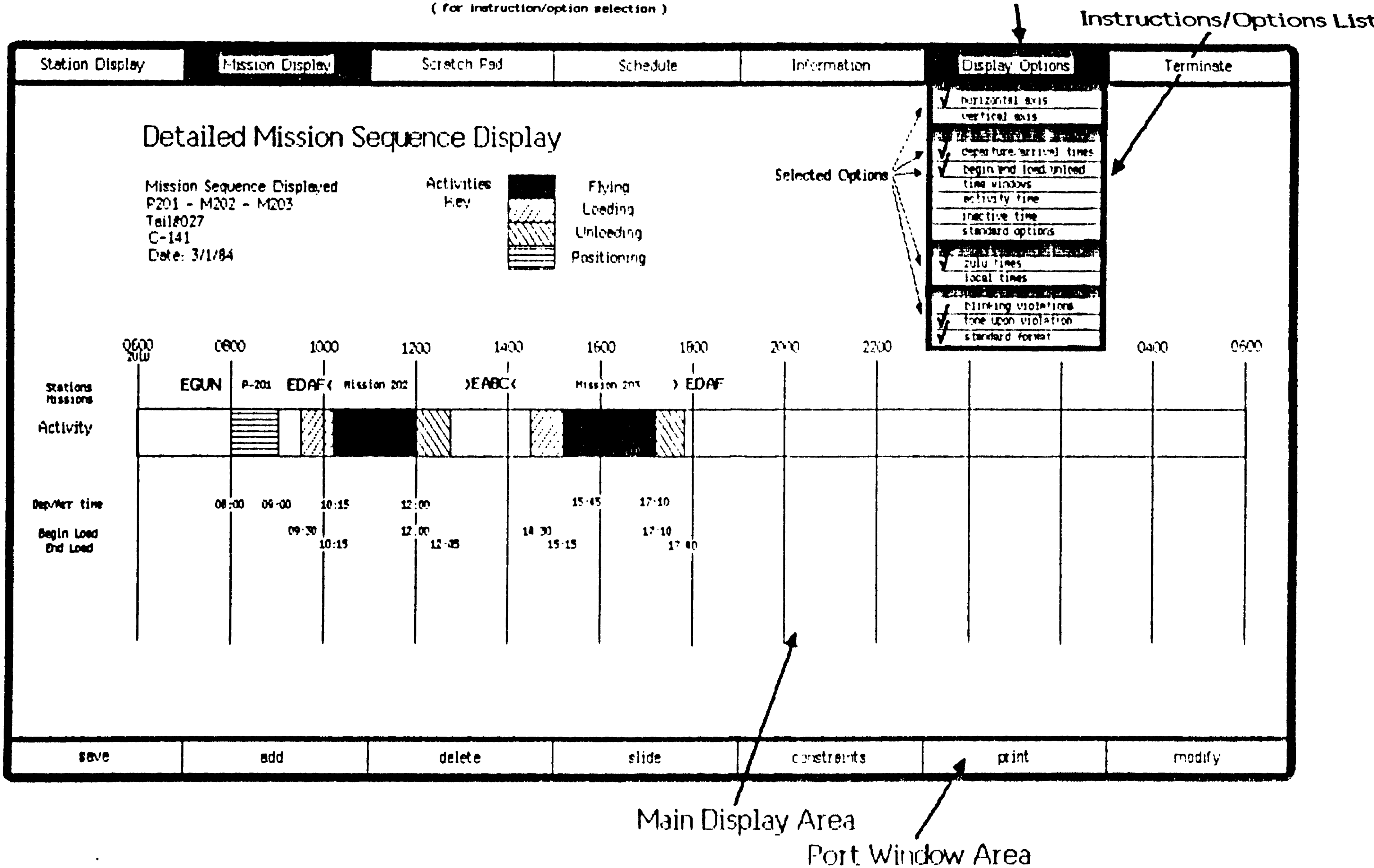
FIGURE 3.5.4 Port Window Locations



box is moused the port"window" is opened and displayed (Figure 3.5.5). The scheduler moves the mouse over the desired instruction in the list and then releases the mouse button. A multiplicity of instructions can be accessed by each port window which leaves more screen area for the main display but does not compromise detail in possible executable operations to be selected.

The port window system for executing instructions is preferred to the usual typing in of system instructions because mousing usually takes less time. Also, the port windows each have a short title for each instruction, thus, the specific instruction does not have to be memorized by the scheduler. He can search through the windows until he finds the instruction he wishes. Executing format for instructions will be set up to maximize the utilization of the mouse. The interactive instructions that will need actual typed input are those instructions that add new information to the database (i.e. new segment, station constraint). Designing interactive system instructions which are initiated by use of the mouse will speed the entire scheduling process; a major goal of this system design.

FIGURE 3.3.3 Optional Port Window
(for instruction/option selection)



4 SCHEDULING OPERATIONS

4.1 Introduction

The purpose of this chapter is to identify the basic elements of the scheduling process and to explore the issues and alternatives in improving aircraft scheduling. It describes the use of the interactive symbolic graphics displays, outlines the information processing needed for scheduling processes, and emphasizes the importance of the database structure design.

Before describing the scheduling process, the important objectives of the system should be reviewed. The most critical and difficult period for the scheduler is during the dynamic overload scenario: Here he finds all of his airlift resources fully allocated while still receiving continuously arriving and changing demands and resource availability information. The objectives for automating scheduling processes are: (1) to achieve higher productivity from a given set of airlift resources through creating missions and schedules for the aircraft fleets which minimize the rejection of higher priority requests and (2) to respond quickly and efficiently to high priority requests for immediate airlift and losses of resources such as aircraft stations, and personnel.

An outline of mission scheduling in the dynamic overload scenario would include these six major scheduling processes: 1) Mission Generation. This function converts tasks to missions by selecting aircraft type and routing. 2) Schedule Map Generation. Specific times are given to scheduled missions and aircraft mission sequences are created which connect missions and positioning flights for each type of aircraft. The graphic structure which displays the schedule for one given type of aircraft is called a "Schedule Map". A set of missions is found which attempts to minimize the number of higher priority tasks which cannot be flown by the available number of aircraft. 3) Crew Mission Sequence Generation. This function creates crew mission sequences and crew staging plans, modifying aircraft schedule maps if necessary. 4) Station Schedule Generation. Station schedules are generated from all schedule maps while reviewing all the station constraints. If there is a station constraint violation, the individual aircraft schedule maps and crew mission sequences must be modified. 5) Schedule Information Status Management. The task database must be kept updated and the planned schedule must be communicated to the operators, planners, and users. This function also saves alternate scratchpad schedules as contingency for subsequent changes in tasks/resources/operational deviations. 6) Schedule and Resource Status Monitoring. This function handles options that are peripheral to the scheduling of the missions but can be a great influence when rescheduling in the future.

These actions are: the monitoring of new or modified tasks as assigned and changes in resource capabilities in order to review their impact on schedules; to monitor operational deviations and review impact on schedules while coordinating requests for transit operations from other schedulers.

Functions 2, 3, and 4 form the core of the scheduling operations to be assisted by the new automated mission scheduling system. The system utilizes graphics display of schedule data with easy, quick manipulation of the scheduling graphics symbology, and decision support assistance in the form of computer algorithms to find good or optimal solutions to scheduling subprocesses.

4.2 Mission Generation

The scheduling of missions in both time and space is the heart of the new automated scheduling system. Hence, it is important that the makeup and generation of these missions is understood. Demand for airlift resources comes to the scheduler in the form of user requests. Those requests that have been scheduled into a planned mission are known as requirements. The word "task" becomes a general descripto for both requests and requirements.

During slack airlift operations, all requests can be successfully scheduled into airlift missions and thus become requirements. However, in the dynamic overload scenario, new requests of high priority may not be scheduled since they are not of sufficient priority for their location in time and space while others of lower priority but different locations are being accepted. After a rescheduling action, the scheduler may find that prior confirmed requests are being bumped. A task database is necessary for accepting requests from users and for the scheduler to draw from when building or modifying a schedule. Algorithmic and graphics manipulation capabilities are not needed for the task database but a simple, efficient database management system is a requirement.

The first major step of the mission scheduling process is mission generation. This function aggregates task loads in time and space to create candidate missions for each type of aircraft. The total airlift fleet consists of a given set of various aircraft types each defined by their payload capacity over stage lengths, block speed and ground times. The task database holds a large set of airlift tasks each specified by their priority, load origin and destination, time window, and load weight and volume criteria. Given this information, the scheduler must find a set of missions such that all tasks are done within their time windows and aircraft onboard loads for any mission segment do not exceed payload capacity.

Missions may not be generated from tasks on a one-to-one basis because of the wide range of task and aircraft sizes. Tasks requiring only a very small amount of space can be combined on one aircraft to make up a mission if their origin/destination and time windows allow. A task calling for movement of a great deal of payload may require a number of aircraft, each being a different mission by definition, to fly the total payload in portions.

Mission generation and schedule map generation (detailed in the next section) are interactive functions and constitute the critical portion of the scheduling system. When building a schedule, these functions are performed sequentially by aircraft type, starting with the largest capacity aircraft first in order to utilize its capacity effectively and to ensure a minimum of empty positioning flights in its schedule map. After the largest aircraft type is fully scheduled, the remaining tasks would be eligible for the next largest aircraft type which would then be scheduled. This process would be repeated for each smaller aircraft type. For two aircraft of similar size, the more efficient would be scheduled first. In the scenario of dynamic overload, there will still be many low priority tasks remaining after all of the aircraft have been scheduled. When there is surplus airlift this sequential process would tend to fully utilize the larger aircraft type and not use the smaller types. For this case, the scheduling system must let the scheduler control each aircraft type fleet size thereby forcing tasks which make up low load missions from the larger to the smaller capacity aircraft types.

The dynamic overload scenario introduces the added concerns that 1) Are the maximum number of highest priority missions scheduled? and 2) When a new high priority task is received, how do we select which task or mission presently scheduled is to be bumped? The first concern can be handled by restricting the given set of tasks for mission generation to those above any selected level, with the remaining requests then loaded on a priority basis into any available capacity in the resulting schedule.

The second concern creates the need for a "load reassignment" algorithm which would fit a given set of tasks of varying priorities, origins, destinations, etc. into an already complete schedule without exceeding any segment payload capacity. If a late request of higher priority arrives, this algorithm would first search the existing schedule in an attempt to find tasks to be bumped from the schedule without requiring a new mission or schedule to be generated.

4.3 Core Scheduling Functions

Schedule Map generation, crew mission sequencing and station schedule generation are the next three scheduling functions. They make up the core of the scheduling system since this is where the missions are given specific scheduled times, aircraft are assigned their routings, and aircraft, crew, and station constraints are tested. Beginning with a set of proposed missions, the core scheduling functions furnish a complete and operationally viable schedule.

More specifically, the schedule map generation function performs the routing of a given aircraft type by linking missions into mission sequences and adding positioning flights, where and when necessary. Given the aircraft type information (flying speed, loading times needed, etc.) and a set of potential missions, the scheduler attempts to find an operationally feasible schedule map which minimizes the number of aircraft required, or the number of high priority missions not flown by the limited airlift resources. Crew mission sequencing and station schedule generation are closely related to the schedule map generation process. Newly generated schedule maps with sequenced missions and positioning flights are tentative until a feasible set of crew mission sequences can also be produced for the schedule and all station constraints are satisfied. Although the core scheduling functions are three distinctly different processes, they must be considered simultaneously when mission scheduling.

4.4 Basic Mission Scheduling

In Chapter 2, the mission was defined as a movement of a specified aircraft carrying its assigned requests between their origins and destinations along a specified routing. The mission has a priority and a time window derived from the assigned requests. The mission is the key element which is moved among the various display charts in the scheduling process. A proper location in both time and space for each mission must be found while airlift resource efficiency is maximized and operational feasibility is maintained. The next few sections will discuss specific scheduling operation that must be handled by the proposed new system to create an modify a schedule in the dynamic overload scenario.

In building a schedule, the operation to add a mission is an obvious necessity. So, too, is a mission deletion operation in the case of a cancelled scheduled mission. Although these two operations could create and maintain a schedule, they are crude for efficient scheduling purposes if they must stand alone. Rescheduling is an extremely complex process and more detailed operations must be similar to how the scheduler wishes to manipulate his given set of missions.

Missions, unfortunately, are not neat packages to be moved rigidly about the schedule; they are somewhat amorphous. Not only can the mission move about in its time window but it can also be made up of many segments where each individual segment must meet schedule feasibility constraints. Although the segment sequence is fixed, the segments can each be moved independently within the mission time window complicating the scheduling process.

Multisegment missions come about in two basic ways. First: a mission is to carry a single task from an origin to a destination but the distance is beyond the assigned aircraft's payload range capabilities. The aircraft must stop at an intermediate point for refueling. Since it would be occupying apron space at this facility, the system must check against station constraints for the refueling aircraft just as any other aircraft using the station. The refueling aircraft would not be taking up any of the station loading/unloading resource. Second: a multisegment mission can be the result of the mission generation process where a number of tasks are fit together onto one mission. For example, given a mission of routing A-B-C-D : this mission may be taking a number of small payloads between A-B, B-D, B-C and so on. The mission window is the intersection of all the tasks windows. The mission number must remain the same until all payload is offloaded at a station by definition. Multisegment missions are thus both a reality in airlifting and an added complexity to the scheduler.

The final feasibly scheduled multisegment mission can take on an infinite number of departure time combinations. The shifting of flying segments within the mission in time is called "sliding" and is discussed in a later section concerned with mission and segment time scheduling.

Since demands in the airlift system are dynamic the scheduling system should accept modifications to existing missions. This could range from a change in load over a segment or a total rerouting of the mission. The scheduler may find he has the option of either making a fueling stop or aerial refueling. Segments may be added or deleted from a mission or time windows modified. To handle all of these cases, a scheduling operation could be designed, interactive with the scheduler to accept mission and segment information modifications.

Other scheduling operations could include the ability to split apart and fuse together missions. For example, a presently scheduled mission's itinerary is A to B to C since it is carrying loads from A to B and A to C. Perhaps the scheduler wishes to "split" this mission into two individual missions A to B and A to C. This would be an opportunity of a mission split operation. A mission "fuse" operation would accept two scheduled missions and create one multisegment mission by a larger capacity aircraft if each segment payload capacity is not exceeded by the combined mission loads.

Thus, the initial set of scheduling operations acting on the basic element, the mission, includes: mission adding, deleting, modifying, splitting and fusing. There must also be adding, deleting and modifying capabilities for the mission's segments. Although crude unto itself this set of scheduling operations is the basis for schedule building.

4.5 Positioning and Sequencing

When building an airlift schedule, the key resource utilized is the aircraft fleet. The routing of the specific aircraft is both the most difficult constraint and the function that can affect the overall airlift efficiency the most. Although it sounds trivial, each aircraft segment's origin station must be the previous segment's destination. This constraint introduces the idea of positioning legs and mission sequencing in order to build specific aircraft routings.

The positioning flight is a scheduling element similar to a single segment mission, whose prime purpose is to move the desired aircraft to a desired location in time and space from a previously inappropriate location. Two situations create the need for positioning flights. In the first case, airlift operations are operated from a fixed base or bases. Many airlift missions are handled in the following way: an available aircraft from the base is flown to the origin, A, in time to fly the mission to destination B, and then returned to its base. The base to A leg is known as a positioning flight and the B to base flight is known as a depositioning flight. Figure 4.5.1 describes this situation. The second case which introduces the use of the positioning flight is illustrated in Figure 4.5.2. Mission 1 is scheduled in its time window traveling between the base and A. A is conveniently close to B where mission 2 returns back to the base. To take advantage of this location in time and space matching, a positioning flight is added from A to B after mission 1 is unloaded and in time for B to be loaded. Positioning legs thus help route aircraft between missions of to/from its home base.

The scheduling technique shown in Figure 4.5.1 will result in very poor airlift efficiency. An example of this inefficient scheduling method is shown geographically in Figure 4.5.3. Three missions are scheduled for a given day from the base. The schedule shows that three aircraft are needed to cover the missions and a total of six positioning legs are flown. However, if we allow remote positioning legs (Figure 4.5.3 (b)) positioning legs can be decreased in number and total flying time. Also the number of aircraft can be reduced if the mission time windows allow.

FIGURE 4.5.1 **Positioning Flight Illustration**

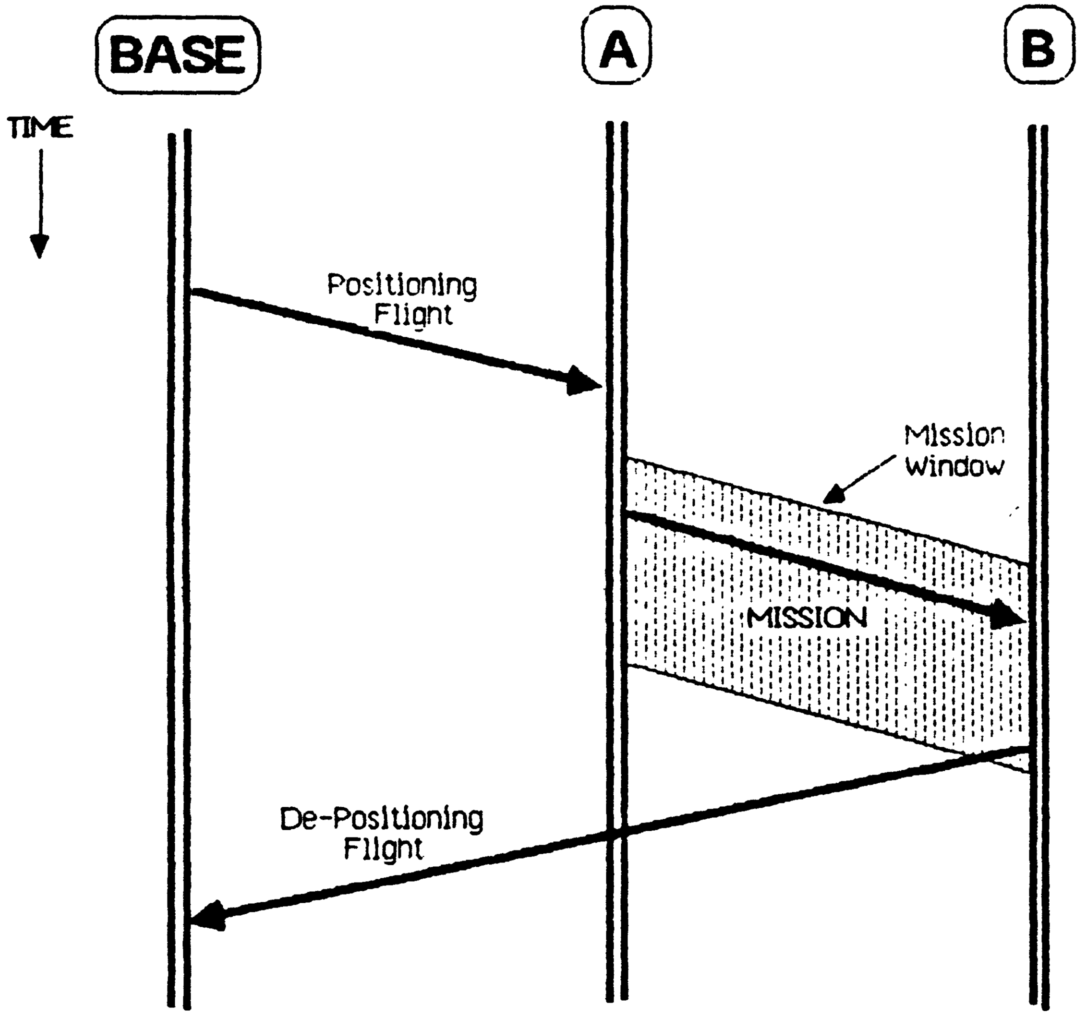
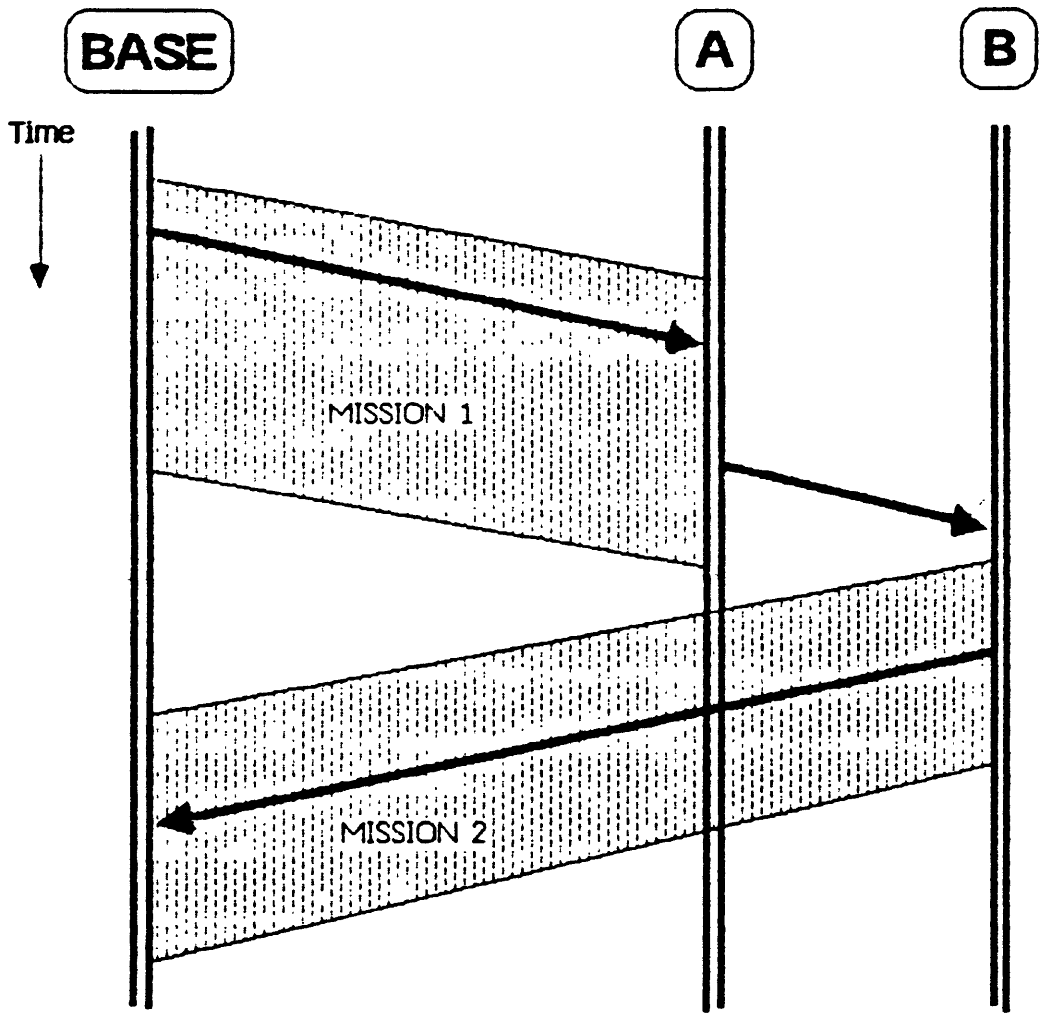


FIGURE 4.5.2

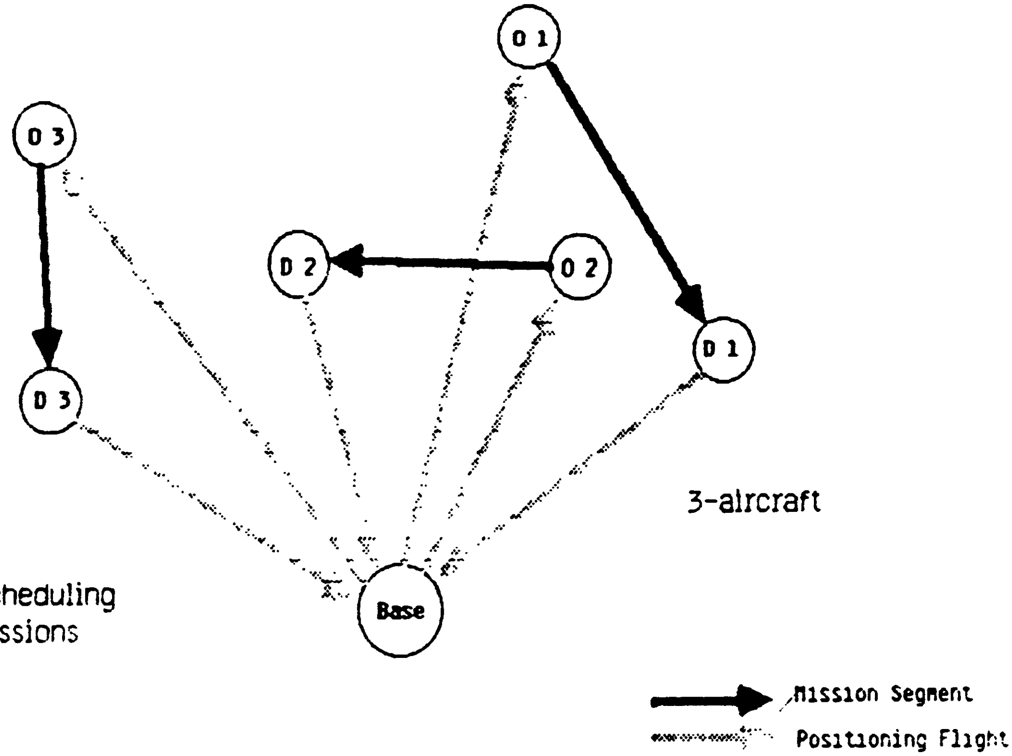
Positioning and Sequencing



Positioning Flight between Mission 1 and Mission 2

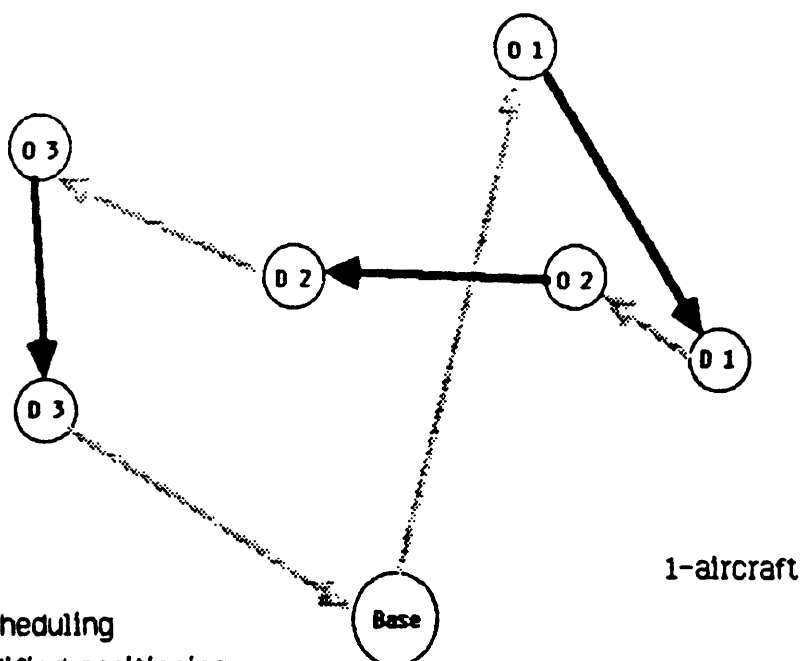
Mission Sequence: Mission 1 -> Mission 2

FIGURE 4.5.3 **Schedule Improvement through Sequencing and Positioning**



(a)

Inefficient Scheduling of Three Missions



(b)

Improved Scheduling through modified positioning flights and mission sequencing

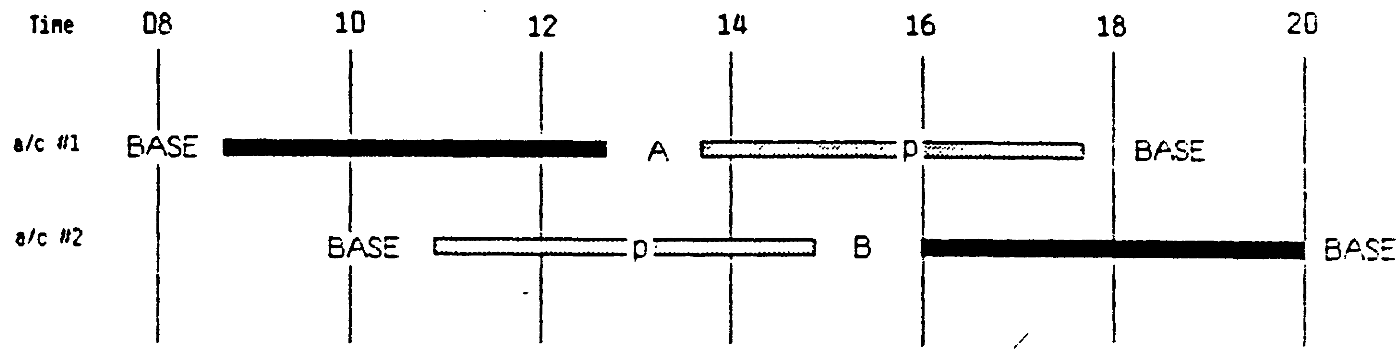
The modified routing to cover the three missions now only needs one aircraft. This idea of flying more than one mission before returning to base requires the introduction of the mission sequence.

Mission sequencing is the function of identifying mission tours that begin and end at the airlift base which increase airlift efficiency over the nonsequenced set of missions individually scheduled. Thus, on Figure 4.5.3, the mission sequence: mission 1 - mission 2 - mission 3 was created. This mission sequence information tells the scheduler that the aircraft which flies mission 1 will be the very same as the aircraft which flies mission 2. Since sequencing deals with aircraft routing, the scheduler will see these mission sequencing opportunities on the mission display easiest.

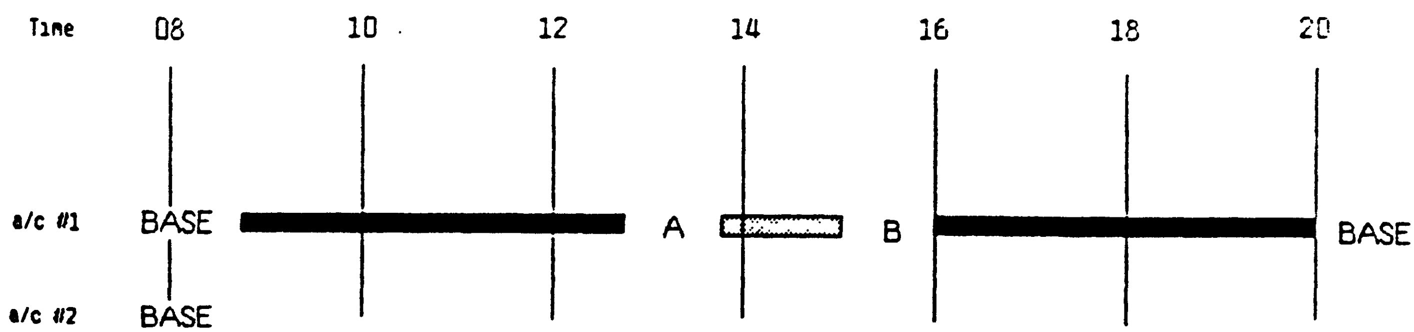
The mission display shows the length of flying time of each segment and also distinguishes between positioning legs and mission segments. Figure 4.5.4 (b) shows the routing of the aircraft described in Figure 4.5.2 (mission 1 - position - mission 2, base - A - B - base). Mission 1 and 2 are not sequenced in Figure 4.5.4 (a) requiring two aircraft (two mission display lines), while the sequenced mission set needs only one aircraft, a far more efficient use of airlift resources. A quick check of the two mission displays will usually show an improved schedule having shorter overall positioning flight times. Rerouting of aircraft,

FIGURE 4.5.4

Schedule Improvement through Sequencing



Example of Routings for two aircraft (expanded mission sequence format)



Improved Schedule utilizing one aircraft and fewer positioning flights
Two missions (black bars) sequenced.

resequencing missions, and deleting or shortening scheduled positioning flights can make significant improvements to the final schedule. An example of many different improvements simultaneously is shown in Figure 4.5.5. The initial display utilizes 5 aircraft, after resequencing and reassigning tail numbers all missions can still be flown at their desired times but with less inefficient positioning flying time.

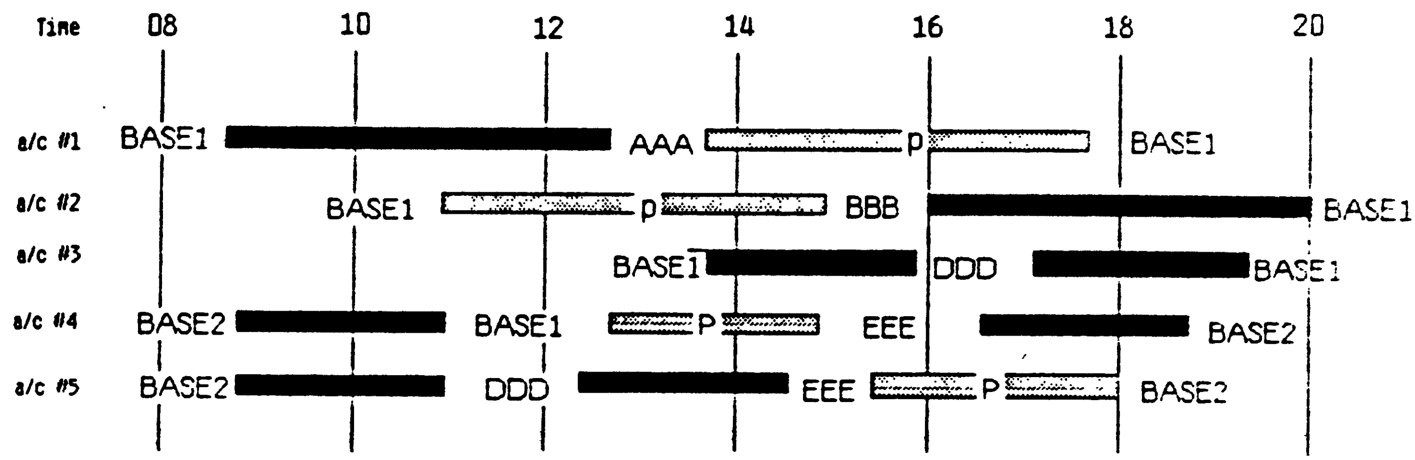
Note aircraft (1) and (2) were handled by (1) while (3) (4) (5) were modified to be flown by (2) and (3). These changes can be handled by "mousing" operations on the mission sequence display to ease the scheduler's search for efficient combinations.

The process of searching through the positioning flights of an aircraft type's schedule map could also be an automated subproblem for the system. An algorithm could be developed to eliminate empty positioning flights that pass each other while ferrying between the two same stations. Utilizing the geographical location of each station the algorithm could be designed to minimize the total distance flown by positioning flights for a given aircraft type and given time band. An algorithm that could minimize the flying of positioning given a set of missions could increase the productivity of the airlift resources immensely.

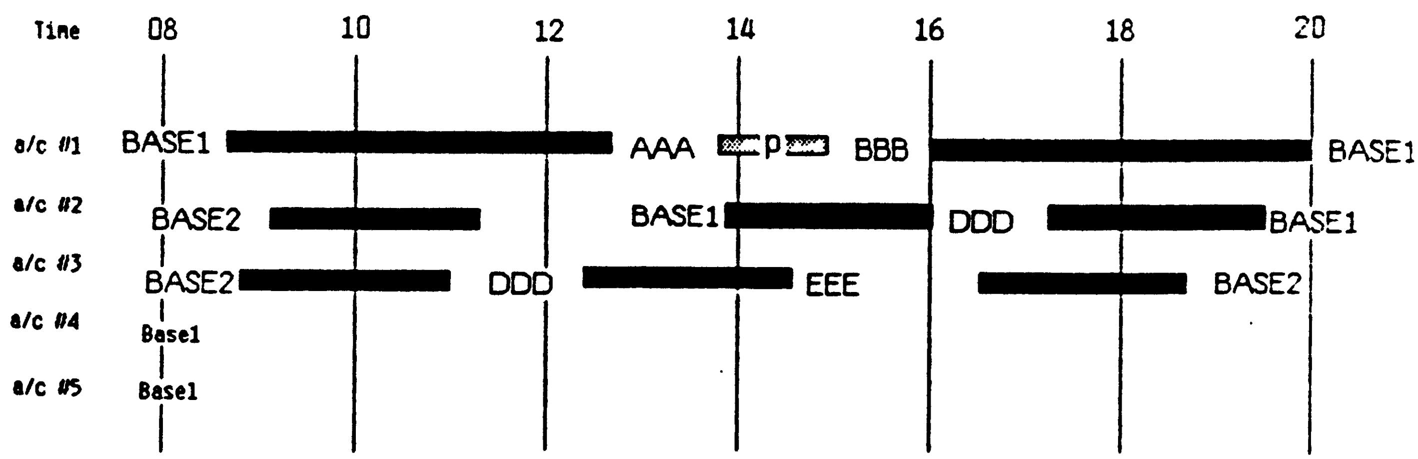
Mission sequencing links two or more missions together

FIGURE 4.5.5

Example of Resequencing Scheduled Missions



Original Schedule

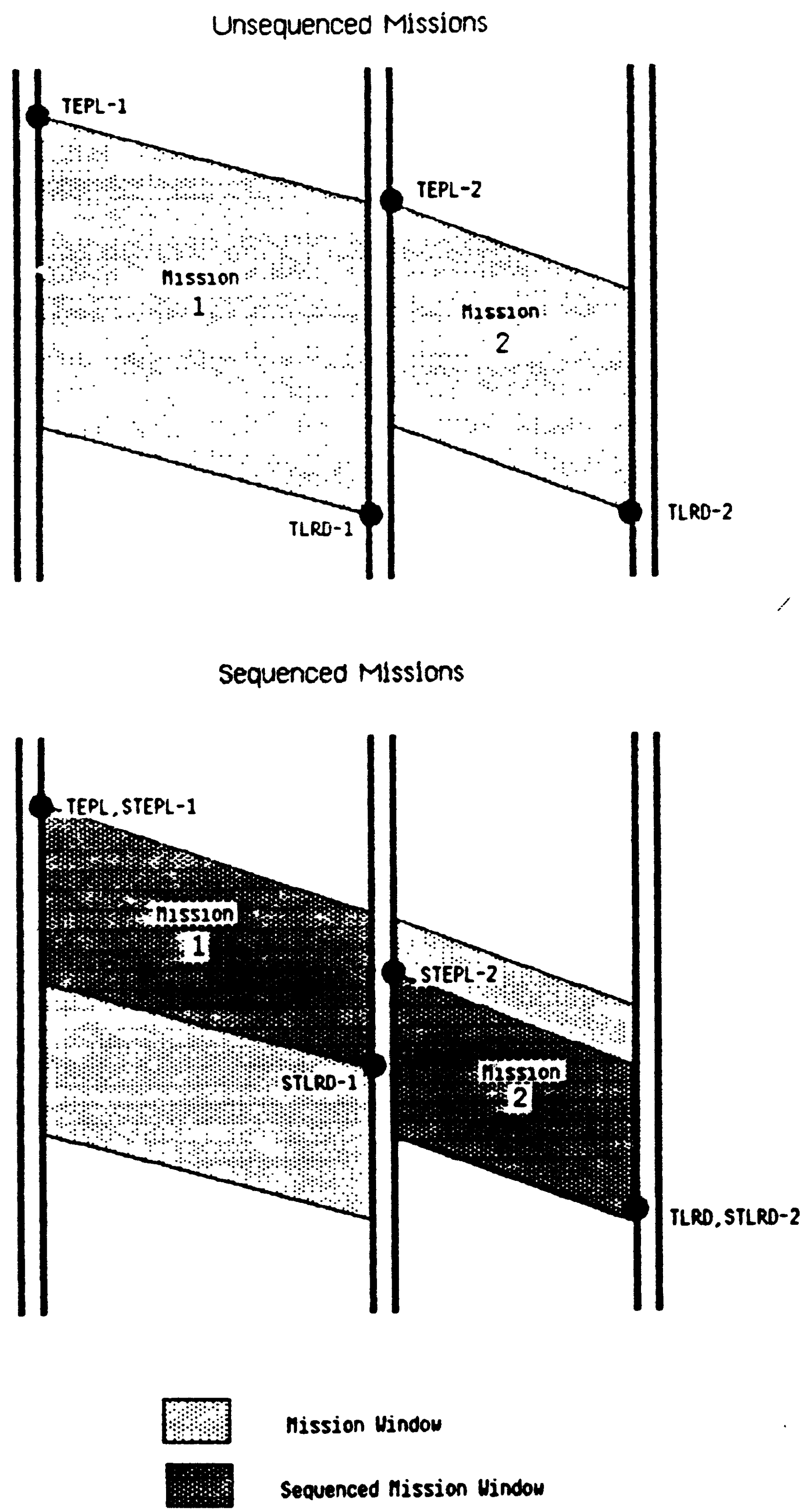


Improved Utilization of aircraft
All former missions flown, but now
two aircraft available for other missions

to the same aircraft for efficient routing purposes. Sequencing affects the time windows. Just as the individual request time windows of a multitask mission constrain the mission time window, so too do the mission time windows os sequenced missions affect each other. The mission sequence itself will have a time window, derived from the intersection of all of the mission time windows in the sequence. Each of the sequenced mission time windows will have the additional constraint of this sequenced mission time window. Figure 4.5.6 shows two missions and their window when unsequenced followed by how sequencing would constrain the individual windows. Each mission window is usually defined by its TEPL (time of earliest possible loading) and TLRD (time of latest required delivery). However, when the mission becomes a part of a definite sequence it must be described by its sequenced TEPL and TLRD since they are the true constraints to a sequenced mission. The STEPL and STLRD are equal to the mission TEPL and TLRD when the mission is unscheduled but takes on different values for each different sequence combination.

Sequencing of missions was introduced to route aircraft more efficiently by minimizing and shortening total positioning flying time. Positioning flights cannot, in the average schedule, be totally eliminated since in many cases they help the overall schedule. The positioning

GURE 4.5.6 Mission Sequence Window



flight must be handled just like missions in the database structure since they are normal flying segments. The positioning flight is assigned the priority of the mission it is flying to. This is done so that a high priority mission is not cancelled because its positioning flight (lower priority in this case) is bumped by a medium priority mission. Positioning flights are not total losses of airlift capability. Low priority tasks can be carried on these positioning flights if their windows are appropriate. These tasks would normally not be scheduled because of their very low priority and the overload of the airlift system.

Positioning and sequencing introduces the positioning flight and mission sequence which together help route the airlift fleet's aircraft efficiently. The positioning slights can be handled just as missions in the mission database. Sequencing introduces the sequenced time window constraint and the need to handle the mission sequence generation (and deletion) operation.

4.6 Scheduling Times

Up to now, general times and time bands have been discussed in the scheduling of missions. For a better understanding of the operation needed to schedule aircraft, the specific times and time bands must be introduced. There are specific times associated with missions, segments, crews, and stations each important to the scheduling process.

Missions and positioning flights are made up of flight segments. Aircraft activities during these segments can be broken down into "active" and "inactive" modes. Active modes include - aircraft flying time

- loading time
- unloading time

each of these times are variable. Flying time depends on the distance between stations, aircraft type, prevailing winds, aerial refueling option and more. Loading and unloading times for each segment will depend on the aircraft, payload, and station equipment to handle loading.

These activity times must be given specific when producing a final complete schedule. For aircraft activities, there would be six specific times possible per segment.

1) departure time, 2) arrival time, 3) begin of loading time, 4) end of loading time, 5) begin of unloading time and 6) end of unloading time. Sometimes two of these times will be the same. For example, the arrival time may be the same as the beginning of unloading; however, if the loading equipment were not available upon arrival the loading would have to be scheduled later.

Times in between aircraft activities is called aircraft "inactive" time. Aircraft are scheduled to have periods of inactivity for two reasons: schedule constraints and intended slack time. Constraints forcing aircraft inactivity are due to times of other resources. Station times such as curfew times, maximum on ground times, maximum K-loaders in use times, fueling times etc. can force aircraft inactivity. A night curfew will force a late arriving aircraft to wait on the ground for a morning departure. If an aircraft arrives at a station when all the loading/unloading resources are busy, it will have to wait until they become available to start unloading. An aircraft may be stopping at a station for fueling on the way to its final destination: although there is no loading or unloading at this station, enough ground time must be scheduled for the aircraft to be fueled. Aircraft may be held at one station waiting for a maximum-on-ground situation to clear at its next destination. The routing of crew and scheduled maintenance time may also force an aircraft into periods of inactivity.

"Slack" time is scheduled inactive time when there is no specific constraint related reason. Slack time is spread among segments and missions by the scheduler to give the real operation of the aircraft a margin of error in completing the schedule. With more slack time in the schedule an aircraft can absorb operational delays and still complete its scheduled routing. If there is no slack time or convenient inactive times, a small delay will require the system to reschedule all of the next several missions and test for feasibility constraints. Slack time in a schedule is very helpful in improving on-time performance of missions. Too much slack time however is a waste of airlift resources and may cause crew time problems. This tradeoff will be the responsibility of the individual scheduler.

The mission display which follows each aircraft is best suited to show various aircraft and crew times, while the station display is designed to show important station constraint times. The detailed view a single aircarft line on a mission display was described in Chapter 3 where the aircraft times can be easily recognized and manipulated. The level of detail of the mission times display is not always needed but the capability is necessary for the system to be complete. Figure 3.3.6 shows a sample mission times display which clearly enumerates the aircraft active and inactive times.

4.7 Sliding

Adding, deleting and sequencing missions and positioning flights can generate an airlift schedule with properly routed aircraft but is a less than optimal set of scheduling operations when rescheduling. In the critical dynamic overload scenario, the scheduler is continuously requested to fit new missions into an already fully capacitated schedule. The result is frequent modifications to the schedule requiring mission time to be moved and updated to accommodate the new "best" schedule. If the scheduler would have to delete mission data and then re-enter all of the information although the only difference is a departure time of 20 minutes, this process would soon become tedious and slow. A set of scheduling operations called "sliding" can avoid this tedium when made interactive with the scheduler. The sliding operations take segments or specific activities and move them in time only. All mission and segment information is unchanged - just the time of the activity is changing. With sliding, mission information must be only entered into the database once eventhough the times can be changed frequently as in the case of rescheduling.

The first sliding operation would be a "mission slide". This is where an entire mission (already scheduled) is "slid" forward or backward in time. For example, a user may have miscalculated the time of earliest possible delivery when filing his request and the scheduled departure happens to be earlier than the user is ready, this will cause a delay in the mission. The scheduler must not slide the mission back in time (within the mission window) until the mission can be completed sucessfully. Missions can be slid forward in time when the previous mission flown by the same aircraft is cancelled.

Missions can be made up of many segments. The sliding of single segments or a combination of segments must also be a scheduling operation. Mission time windows and mission sequence time windows are the primary constraints to the range of segment and mission sliding. Aircraft, station, and crew constraints in combination with time windows may be so restrictive that upon sliding a segment there is no possible way to schedule the mission. At this point the scheduler must attempt to relieve some of the constraints by sliding other segments.

Detailed activities slide is also important for a complete scheduling system. Consider a multisegment mission with slack time between each segment. The scheduler would like to have the following abilities: slide one segment in time not affecting the other segments: slide all segments after the n-th segment (due to a delay on segment n); reduce the total time of mission completion by shortening slack time between segments; increase probability of mission on time performance by inserting extra slack time. Of course, all segments that have been slid must meet all local time constraints before the slide can be deemed successful.

Sliding is not restricted to segments as a whole. Specific loading, unloading, and flying time can be manipulated using the mouse and hold procedure on the mission sequence display. The act of invoking a sliding operation will be interactive and very easy. The scheduler frist selects the segment(s) or activity time(s) to be slid, then holds down the mouse while moving the element symbol(s) across the screen to the new time. When the mouse is released the system recognizes the move and updates the database appropriately and runs checks of all constraints enumerating those which have been violated by the slide. This has been successfully achieved on the current LISA mission sequence display.

4.8 Scheduling Constraints

A detailed set of scheduling operations designed to create, delete, move, and modify airlift missions has been outlined. For each scheduling action there are a multipicity of constraints which are applicable and must be considered. The idea of mission sequencing and positioning flights were introduced as a result of aircraft routing constraints. However, there are also crew related, station and other general constraints to the scheduling process.

For a given schedule, a feasible set of crew mission sequences must be found which will supply each aircraft segment with a flight crew. Crews will have different routings that aircraft since they must follow certain duty time requirements, while aircraft do not. After a certain amount of duty flight time, the crew is required to have a specified amount of resting time before operating another flight segment. Also, crews are differentiated by the base which they are from. Aircraft, however, are not. A base may be in charge of ten aircraft but which specific ten aircraft tail numbers may be interchanged frequently with other bases. An aircraft may leave its home base and fly several missions returning weeks later to the base. The crew, however, would like to be scheduled home much sooner than that. Thus, to generate a feasible airlift schedule not only must the aircraft be routed but also the crews must be routed under different constraints.

Due to their different scheduling constraints, crews and aircraft will not follow each other on a one-to-one basis. Crew doubling and pre-positioning are used to make up for this descrepancy. On long haul missions two crews may be assigned to an aircraft. While one crew is on duty the other is resting. Using this method frequently begins to drain crew resources quickly. Positioning of crews can make better use of crew resources by may be complicated to schedule. For example a mission flies from the US to Europe and on to the Middle East. In Europe, the original crew must disembark due to duty-time constraints but the aircraft is scheduled to continue immediately. Here, a crew rested from the previous day's arrival can fly the Europe to Middle East leg. The crew that was left in Europe will within 8 to 12 hours be ready for duty again. The scheduler will try to find a convenient mission for them to return home or fly elsewhere. However, if the scheduler could not initially find a crew in Europe to fly the Middle East segment of the mission, he would have to delay the departure from Europe until the original crew had completed their rest period. The scheduling of crews may, thus, cause a rescheduling of the aircraft missions. In the above case the second segment of the original mission would have to be "slid".

The assignment of crews to a specific schedule is a set covering problem. However, crew scheduling will be still done manually since present set covering algorithms are not satisfactory for the scale and complexity of this airlift case. Instead, a less complicated subprogram would follow each of the crews and monitor duty and rest times, as scheduled. Whenever a crew constraint was violated the program would flag the violation. The number of available and resting crews at a given station would be added to the station database's event list. Whenever a crew arrival, departure, or change from rest to ready of a crew occurred a new event would update the station information. The scheduler would use this dynamic crew information when scheduling crew mission sequences.

In scheduling aircraft and crews, the finiteness of station resources must be recognized and respected. Station resources can change dynamically. For example, a station's runway may be temporarily closed for repair. The design of the station database (Chapter 2), uses an event base structure for easy collection and updating of dynamic station constraints. To build and maintain this dynamic constraint list, a set of appropriate interactive operations to accept and store incoming constraint information is necessary.

The following includes a partial list of station attributes or resources that can impose constraints on the aircraft or crew scheduling process. The number and length of runways is an attribute of the station. Runways may be periodically closed due to damage, disaster or scheduled repair. The length of the available runways (at a given point in time) will dictate which aircraft types can utilize the station and their payload/range capability.

Parking areas at each station are of finite size. Some stations will be able to accept 40 aircraft and some as few as one. It is important to schedule only as many aircraft at the station as there are parking slots. When the maximum number of aircraft are on the ground at a station this is called a "MOG" condition. Calculating a station

MOG is more complicated than simply counting up the total number of aircraft. A station may be able to accept five large aircraft or 10 small aircraft or a combination of the two. Thus, the MOG condition must be specified as a number of representative combinations. For borderline cases it will be the responsibility of the scheduler to judge the feasibility of a station's parking condition. In some cases, the order of departures and arrivals may modify the parking problem. If a MOG is exceeded the scheduler must attempt to reschedule the participating missions within their time windows in order to relieve this condition.

Another finite station resource is the loading and unloading capability. Much of the cargo flown on the airlift system is loaded and unloaded using K-loaders. Each station has a given number of these K-loaders. This number can be dynamic as K-loaders can be moved from station to station. The number of K-loaders can constrain scheduled activity in that a K-loader may not be available to an arriving aircraft immediately, forcing the aircraft's turn time to be longer than the minimum turn time of the station. The scheduler will have to "slide" the original mission times scheduled. Aircraft ground servicing crews may also affect the schedule in the same way.

Fuel availability can be critical at some stations due to a conflict in delivery, storage and scheduled activity. If the scheduler finds that the fuel available will not be enough for the present schedule he must 1) for the short term, reschedule flights to avoid that station or "tanker" in fuel, and 2) if the condition persists, make suggestions on a modified delivery or storage arrangement. Fuel may be used by aircraft activities at each station outside of those scheduled. A prediction of such fuel usage and delivery is required.

Stations which are located in noise sensitive areas may have arrival and departure curfews imposed. All missions scheduled into and out of the station must avoid operations during the curfew hours.

The above mentioned station attributes and constraints are specific to each station. In the manual scheduling process the scheduler would have to keep track of all aircraft activities at a station and then check them against a set of constraints for the station. This time-consuming task can be easily taken over by the computer. The computer would have two time based lists. One list would update the status of activities actually happening or scheduled at a station; a new element to the list would be generated at each event time. The second list would hold the dynamic station resources and attributes; each list element would update the station constraints. Whenever an activity was modified in the schedule due to mission generation or sliding, the computer would search through the station database lists and compare the new event station status to the appropriate station resource list element and check for any violated constraints. Upon recognizing a violation, the system would alert the scheduler with a "flag" (Chapter 3) on the screen.

In addition to monitoring crew and station constraints, the system could also monitor more general aircraft and scheduling constraints. The system could test whether more aircraft are being scheduled that are in the fleet (more lines on the mission display that aircraft). A scheduler may unknowingly plan a mission requiring a greater payload or range than the selected aircraft type is capable of. An aircraft may be scheduled into a station whose runways are insufficient in length for operation of the type. These violations can be automatically recognized and displayed by the system.

Automatic constraint testing can occur during the data entry process of mission, station, and other schedule data (e.g. the scheduler attemps to create a new mission or station that already exists). Mission flight segments must be scheduled within their time window. The same aircraft cannot be scheduled for two activities at the same time, and missions can only be sequenced with appropriate positioning flights and time window compatibility. These and many other scheduling constraints could be tested automatically by the system.

The instant updating of the database and automatic constraint resting upon rescheduling is a powerful advance over the manual scheduling process. The scheduler is free of these time consuming and tedious tasks. He can devote his efforts to more frequent change to the schedule, moving it closer to the "best" solution in a given amount of time. He can have complete confidence in scheduling, knowing that if a constraint were violated or an infeasible schedule proposed the system would immediately alert him.

4.9 Scheduling Examples

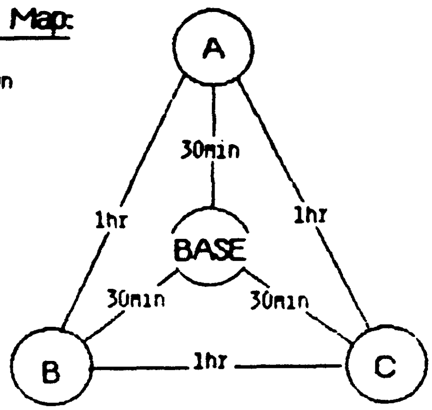
The basic scheduling operations and schedule display designs have been detailed in this and the previous chapters. A few examples of the typical use of the proposed systems and methods will help in understanding the system design and improvements over the normal scheduling process.

The first example of scheduling will include both schedule building and rescheduling. A very simple case of a small scheduling universe will be introduced. The scheduling universe is described in detail on Figure 4.9.1. There are four stations; one base and three other stations A, B, C. There is one half hour flying time to stations A, B, and C, while flight times among A, B, and C are each one hour. To simplify the constraints, all loading and

FIGURE 4.9.1 Sample Scheduling MiniUniverse

Universe Route Map:

Flying Times Shown



Aircraft data:

- 3 aircraft available
- Each aircraft can fly all possible segments
- All aircraft overnight at the base.

Station Constraints:

Station	Loading Time	Unloading Time	Parking Slots	Number of K-loaders	Curfew Ends
Base	30min	30min	15	5	0600
A	30min	30min	3	1	0600
B	30min	30min	3	1	0600
C	30min	30min	3	1	0600

unloading times are each one half hour at all stations. Stations all have night curfew until 0600 on all flight operations. A, B, and C have each only three parking positions and one K-loader (all loading and unloading is assumed to be done with the use of a K-loader). The airlift universe has three aircraft available, all initially at the base.

The example is limited to the scheduling of aircraft on one morning only, not worrying where the three aircraft are at the noon hour as long as they follow a feasible routing and satisfy all constraints. Imagine it is the night before and the scheduler is accepting (dynamically) missions to be flown the next morning. He now begins to build the schedule.

Suppose the first set of missions to be scheduled are as follows:

Mission	Routing	Time Window
#1	A-B	before noon
#2	A-C	before noon
#3	B-C	before noon
#4	B-A	before noon

The scheduler begins with the first three missions. He chooses to schedule positioning flights to the cities of A, B, and A respectively to prepare for these missions. Figure 4.9.2 shows the first three missions scheduled each with a positioning leg one half hour for loading and then the hour flying time. The system however is flashing a

MiniUniverse Displays

Mission Displays

Station Displays

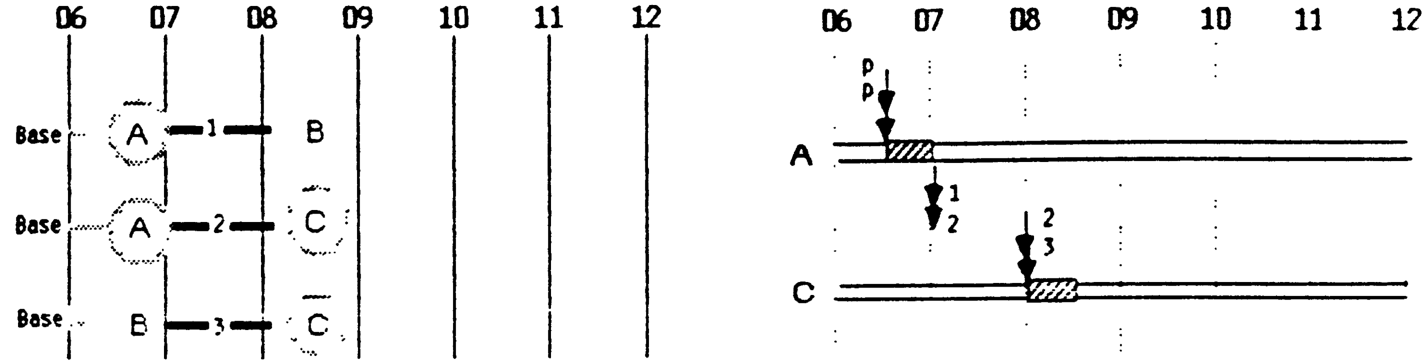


FIGURE 4.9.2 STEP 1

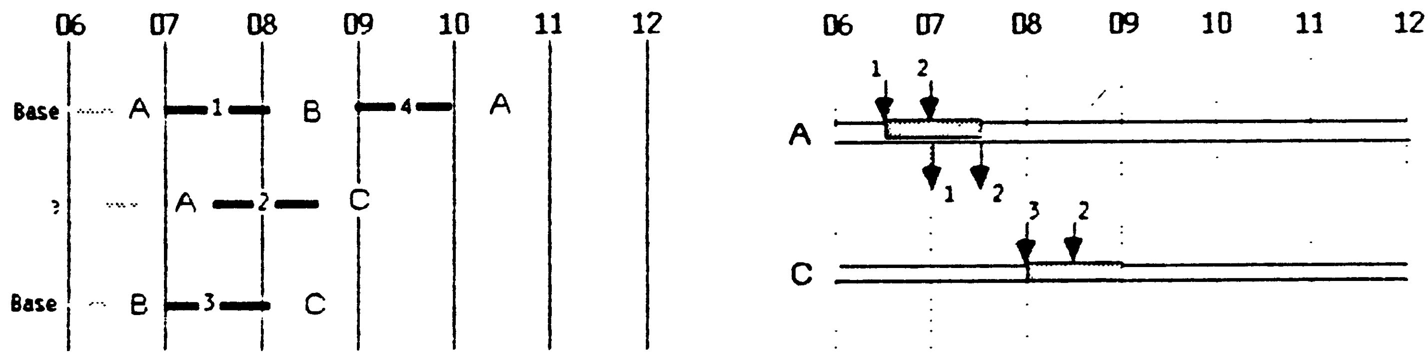


FIGURE 4.9.3 STEP 2

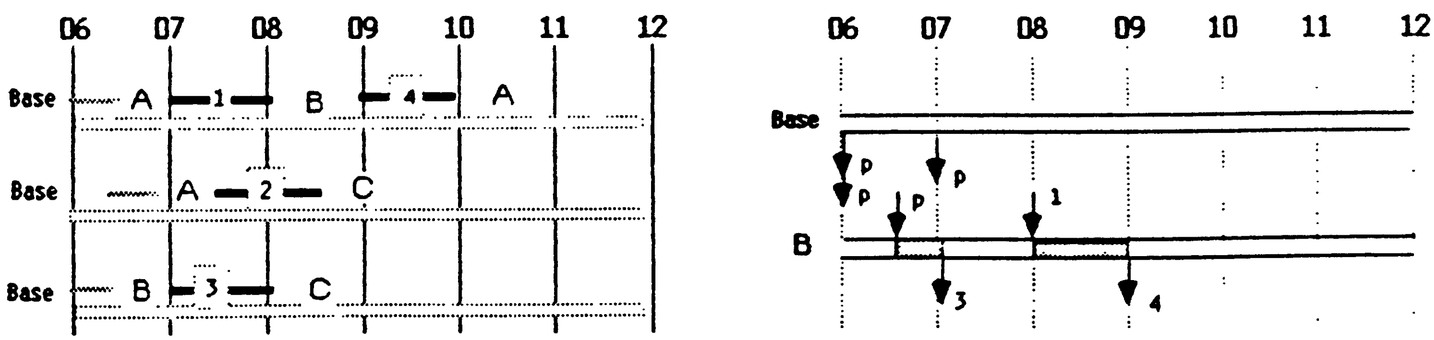


FIGURE 4.9.4 STEP 3

flag for the 4.9.2 schedule because of loading and unloading constraint violations. Aircraft #1 and #2 are both loading between 0730 to 0800 at A and aircraft #1 and #3 are both scheduled to unload between 0900-0930 at station C. The scheduler must act to avoid these violations. He chooses to slide mission #2 (A-C) later by one half hour to satisfy the loading constraints at both A and C making the schedule once again feasible. He now adds mission #4 to the schedule by adding it onto the end of mission #1 (Figure 4.9.3).

The scheduler finds himself with a little time to review his schedule. He thinks he can reduce the number of positioning flights or aircraft needed by resequencing the existing schedule. To help him decide he selects to view the time windows of missions #2, #3, #4 on the routing chart (Figure 4.9.4). He sees that four segments can be flown by only two aircraft and only two positioning flights. He "slides" the schedule missions appropriately using the mouse. The improved feasible schedule is shown on Figure 4.9.5.

Additional candidate missions have just arrived at the scheduler's workstation and he is asked to fit them into the next day's schedule if possible. The new requests are:

Mission	Routing	Time Window
#5	C-A	before noon
#6	Base-C	before noon

MiniUniverse Displays

Mission Displays

Station Displays

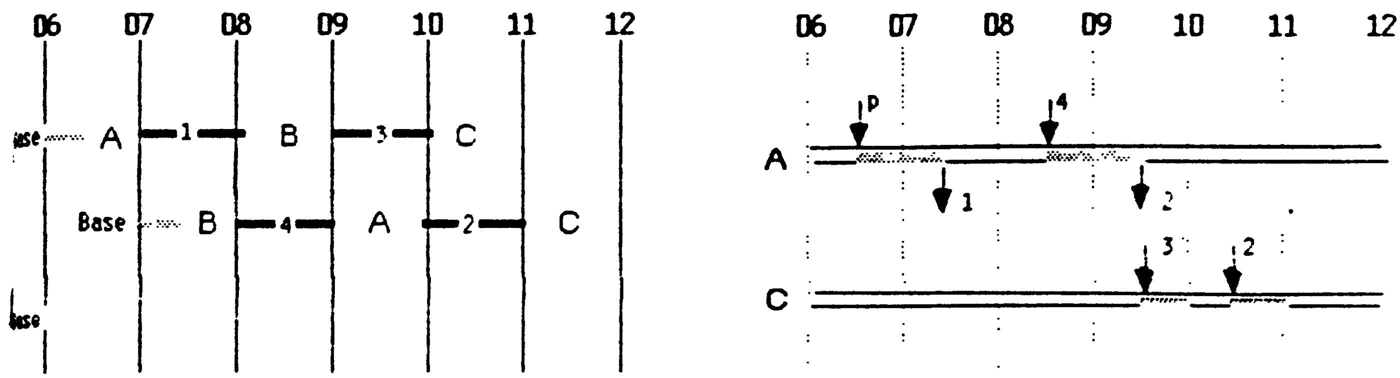


FIGURE 4.9.5 STEP 4

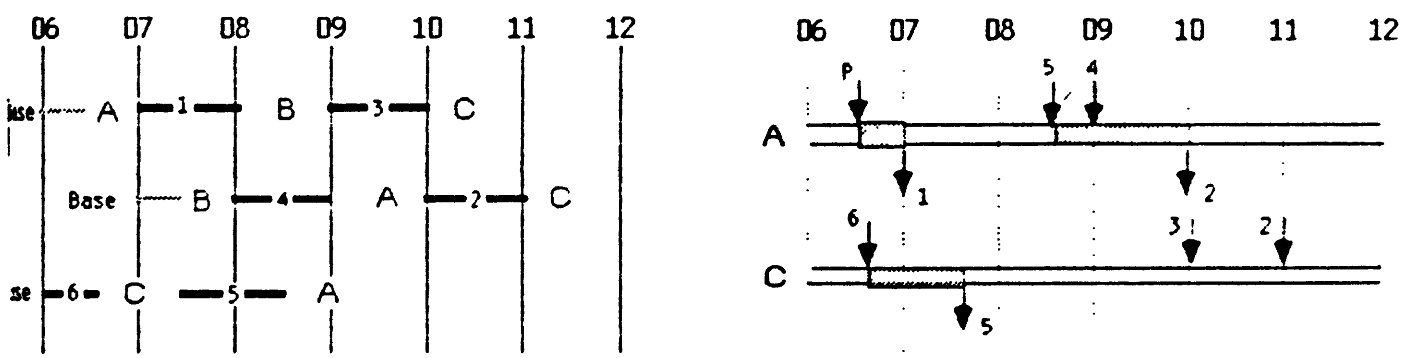


FIGURE 4.9.6 STEP 5

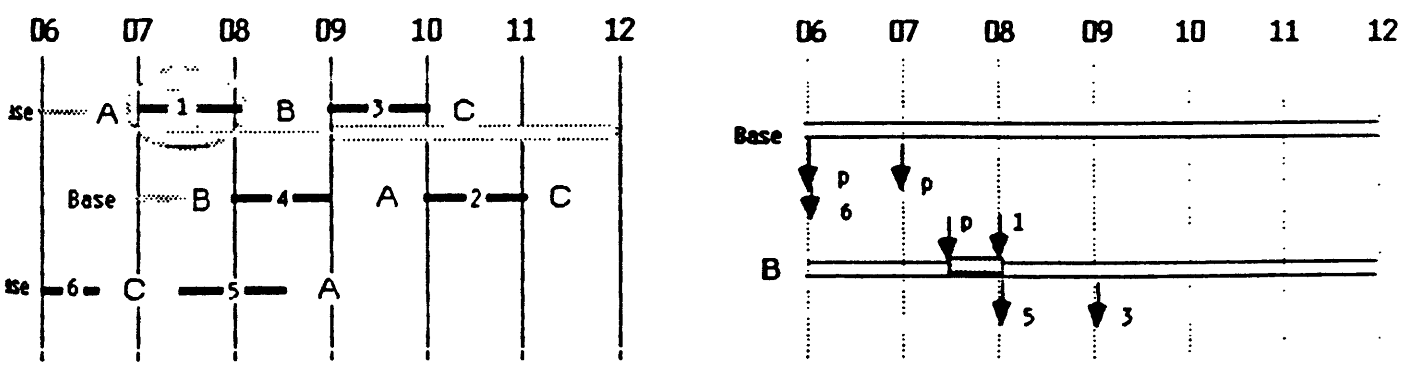


FIGURE 4.9.7 STEP 6

By creating a new sequence of mission #6 followed by #5 and assigning this to aircraft #3, the scheduler can accommodate these new missions without changing the other scheduled mission times (see Figure 4.9.6).

One of the users has just informed the scheduler of a modification to a mission. Mission #1 will not be able to depart before 0900. The mission is presently scheduled to depart A for B at 0700 (Figure 4.9.7). By sliding mission #1 from its present position to aircraft #3 re-sequenced behind mission #6 and #5 the mission modification can be handled. Aircraft #1 must have a new positioning flight to station B as the first segment (Figure 4.9.8). The resulting schedule is flawed due to a violation of loading constraints at A between 0900 and 0930. The entire sequence for aircraft #2 is slid by 30 minutes to create a operationally feasible schedule (Figure 4.9.9).

The schedule must accept one last high priority mission. Mission #7 must be flown between the Base and A before noon. The scheduler is hoping to fit this new mission into the schedule without bumping any of the other existing missions. Through a complicated combination of mission sliding and sequencing, a feasible schedule (Figure 4.9.10) is found to accommodate all of the seven missions within their respective time windows. For a small universe such as this the scheduler could have invoked an interactive scheduling algorithm to give him this final

MiniUniverse Displays

Mission Displays

Station Displays

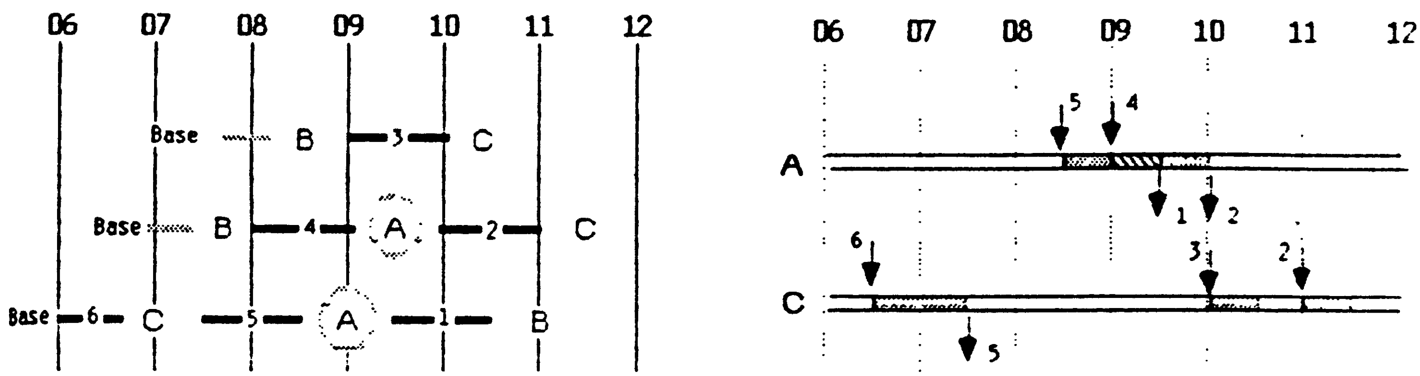


FIGURE 4.9.8 STEP 7

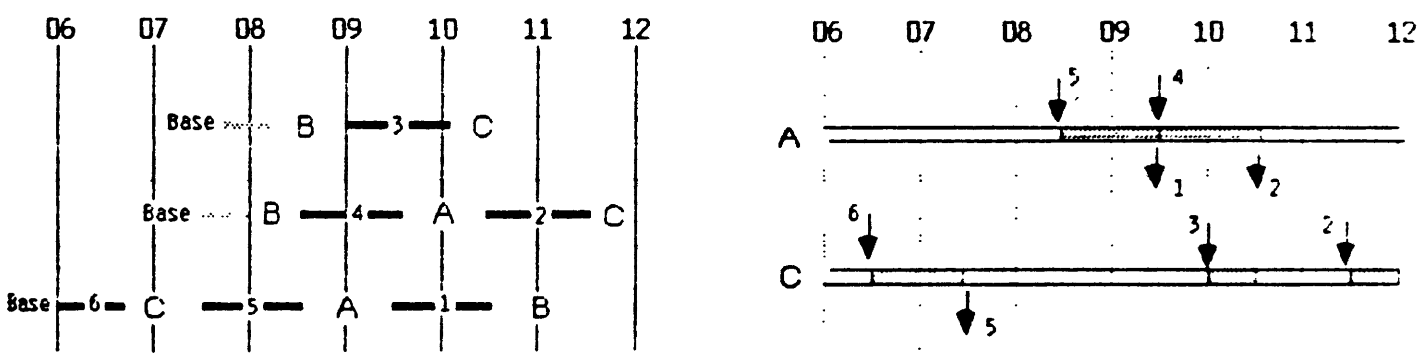


FIGURE 4.9.9 STEP 8

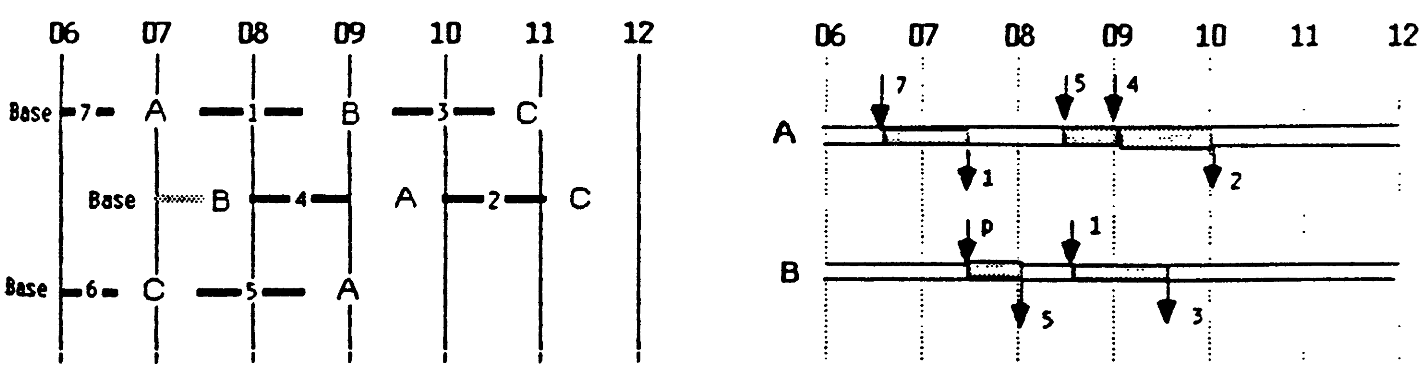


FIGURE 4.9.10 STEP 9

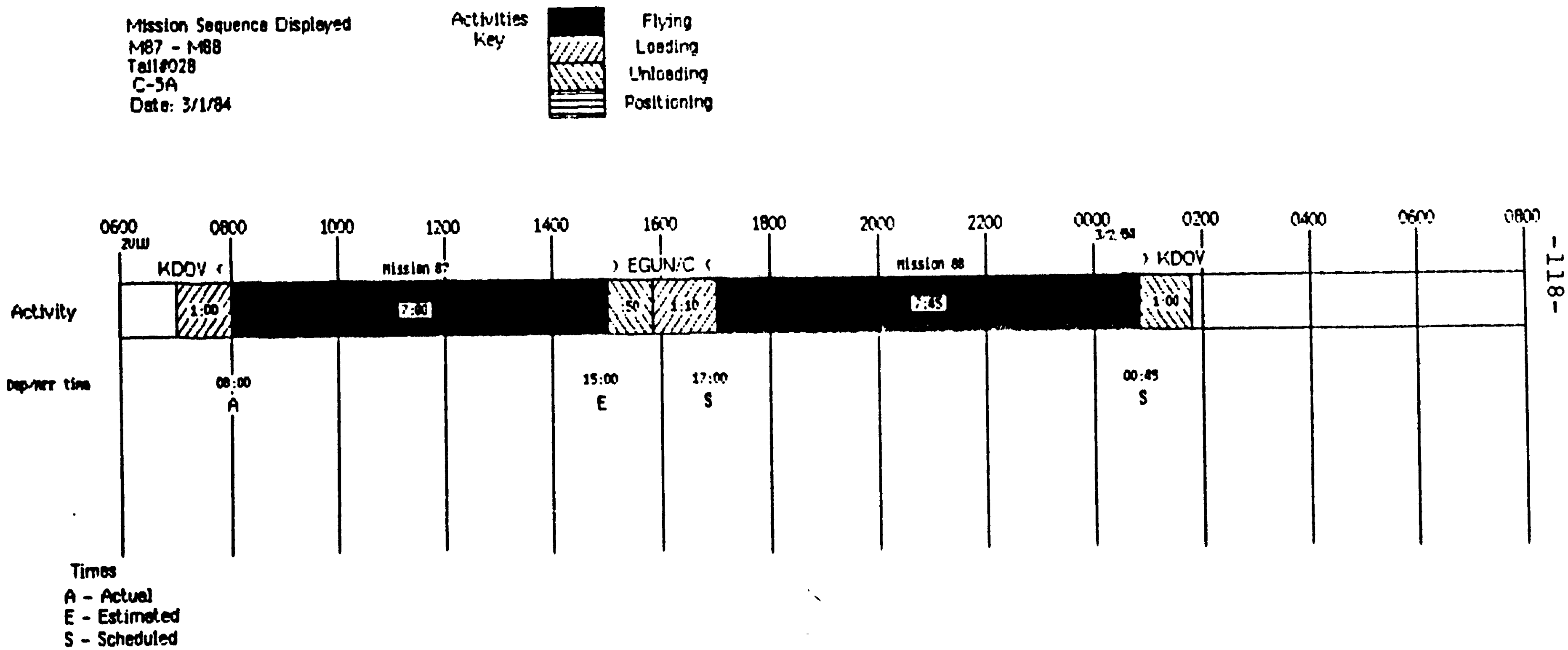
answer. The MiniUniverse example shows the need for convenient and simple procedures to slide and sequence missions and positioning legs.

A second case of sample airlift scheduling includes both crew and aircraft scheduling considerations. Initially the scheduler has planned a trans Atlantic roundtrip for aircraft we are interested in following. Figure 4.9.11 shows the aircraft loading at Dover Delaware, flying seven hours to Mildenhall (England) where mission 87 is unloaded and there is a crew change.

Immediately, thereafter mission 88 is loaded and the aircraft returns to Dover. These scheduled activities satisfy all present schedule constraints since there are no violation flags showing. The letters A, E, and S are introduced next to the departure/arrival times. These symbols denote whether the time shown is Actual, Estimated or Scheduled. Since the mission 87 departure time is actual and the arrival time is estimate, the scheduler knows that mission 87 is already in progress.

Dynamic changes in resources are typical of the airlift system. Suppose, the scheduler was just informed that there would no longer be a crew available in England to effect the scheduled crew change. The scheduler removes the crew change at England from the schedule where upon the computer returns with a crew constraint violation alert. The computer specifies that at least 8 hours of rest are

FIGURE 4.9.11 Example of Aircraft Activity



needed for the crew between mission 87 and 88.

The scheduler adds the mission 88 time window to the display to see the flexibility he will have to work with (Figure 4.9.12). The time window shows the band between the TEPL and TLPD for mission 88. Fortunately, the time window is large enough for the mission to be slid far enough to allow for the necessary crew rest time. Also, the aircraft is not yet scheduled for other activity during this needed time band. Figure 4.9.13 shows the "slid" mission 88 and the new feasible schedule. Both aircraft and constraints are satisfied.

In the previous scheduling examples, a feasible solution was always found for various demands and constraints in the schedule. In some cases, however, the scheduler will not be as fortunate. Figure 4.9.14 shows an example of a new mission X to be added. The combination of the mission time window, aircraft flying times, and station curfews allow no possible method to schedule the A-B-C routing of mission X. In this case, the scheduler must report to the user that his request, regardless of priority, cannot be scheduled.

4.10 Scheduling Assistance through Automation

The proposed new scheduling system introduces the modern computer with graphics as a new tool used for ef-

FIGURE 4.9.12 Crew Constraint Violation

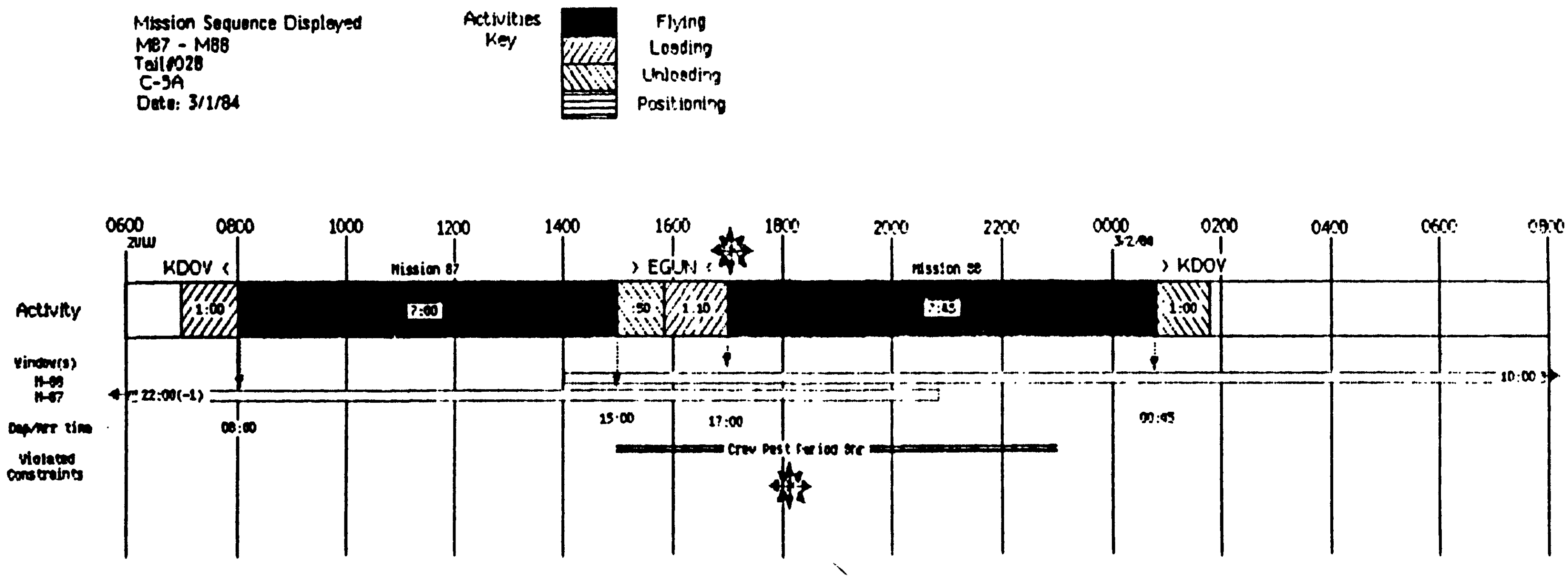


FIGURE 4.9.13 Rescheduled Feasible Solution

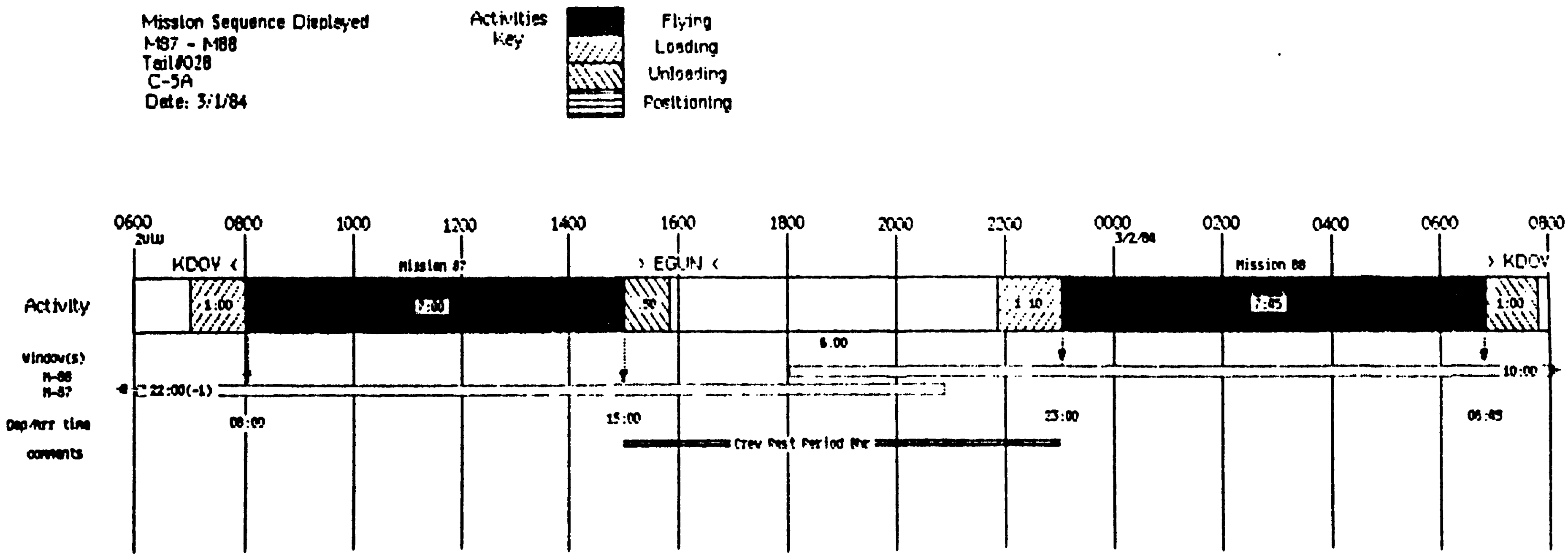


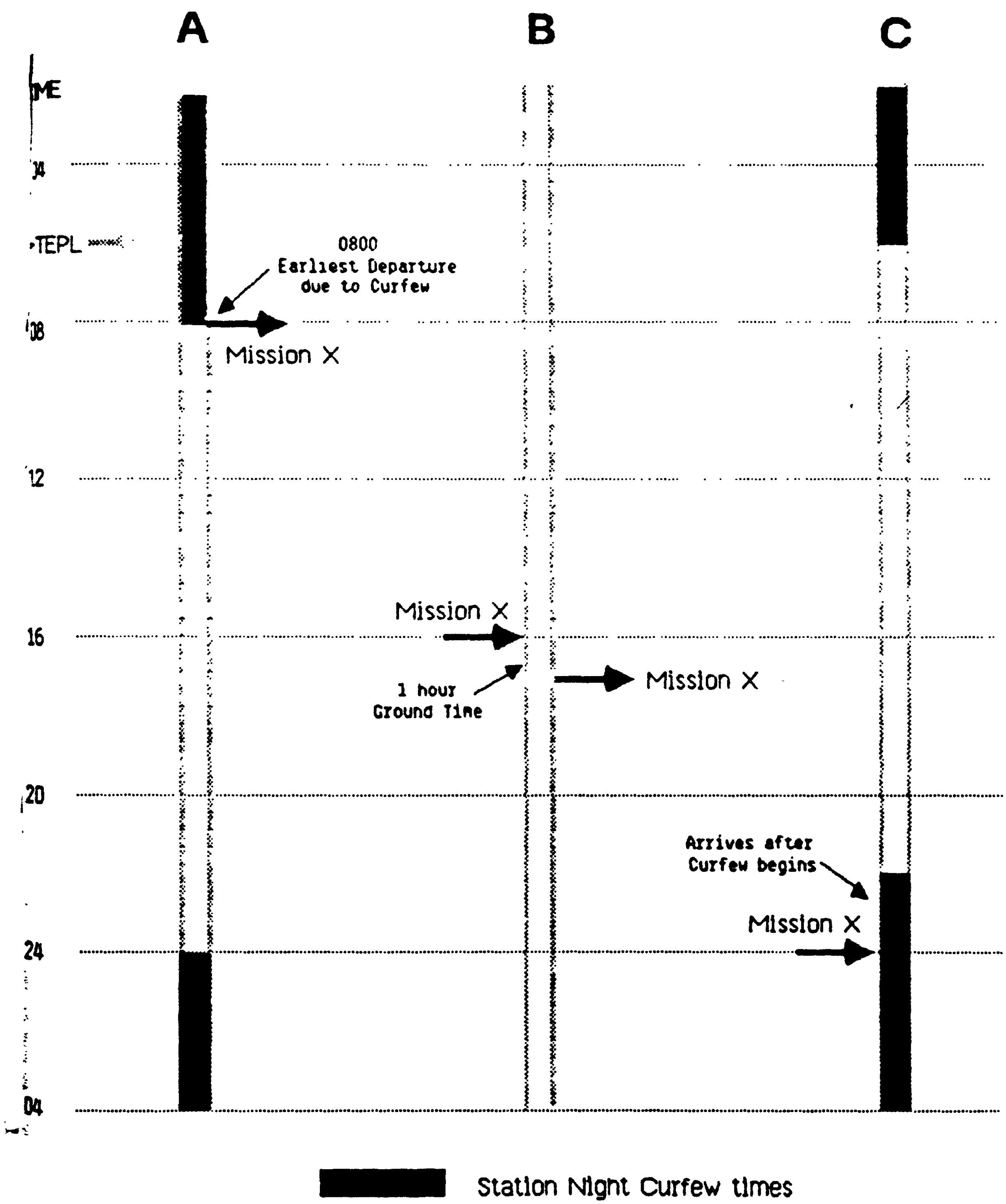
FIG 4.9.14

Infeasible New Mission

New Mission X
Routing A-B-C

TEPL = 0600 @ A
TLRD = 0600+1 @ C

A-B = 8hrs
B-C = 7Hrs



fective dynamic aircraft scheduling. Designing the system around this idea was not done to simply add high tech graphics to scheduling but to convincingly improve the capabilities of a scheduler and the scheduling process The introduction of the computer can be shown to both speed and increase efficiency in scheduling over the manual method.

The scheduling operations have been specifically designed to minimize typing-in of data. The mouse, which is operated by moving across a desktop and clicking the button, is used to manipulate symbols on the screen and initiate scheduling commands. Complicated rescheduling actions can be effected by a few appropriate movements and clicks of the mouse.

Automatic redrawing of the schedule upon modification is a simple task for the computer. However, in the manual process, that action may be so very time consuming that it is done as infrequently as possible. The scheduler works with cluttered drawings for long perios before redrawing the scheduling charts. This may compromise the scheduling process. Immediate screen display updating guarantees readable uncluttered displays for the scheduler to review and manipulate.

Since the computer has the ability to store both schedule database and display pictures, a number of complete schedules can be stored. This can be very helpful to schedulers who wish to create a number of candidate schedules known as "scratchpad" schedules. There can be three status levels of schedule data. The first, "scratchpad" level of schedule information is where the scheduler works to create one or more schedules which are operationally feasible, but he does not want anyone else to see this data. The second level is "planned" schedules where an operationally feasible schedule is issued for preview and comment by planners and operations. This planned schedule may still be changed due to late arriving requests and unexpected deviations but is intended to be the final committed schedule. The third level is "committed" schedules where mission times have been specified and forwarded to the users and operators.

The idea of manipulating symbols on schedule displays with automatic update of the schedule database is the most significant proposal in this interactive dynamic scheduling system design. This concept requires the marriage of a detailed schedule database and modern computer graphics through database management and graphics software.

In Chapter 2 a simple proposal for the schedule database structure, Chapter 3 presented the designs, options, and flexibilities of proposed graphics schedule displays, and this chapter has enumerated the necessary schedule manipulation operations needed for effective airlift scheduling. Each of these elements were designed with the others in mind, so that when these elements are all combined by the final master software programs, the scheduling systems will execute smoothly and efficiently. The graphics/operations/database interface software will be highly specific to the computer chosen for being the most to this new scheduling system. Prerequisites for any system control program . are that the system must recognize the graphics display symbols as representations of the schedule database and that specific operations of the screen manipulation device (the mouse) initiate scheduling operations using database modifying subroutines.

Computer automation of the constraint testing functions is probably the most significant innovation of the proposed system to speed the scheduling process. The detailed and difficult task of testing the proposed schedules feasibility can be left to the computer on the appropriate constraints have been entered into the database. Aircraft, crew and station constraints can all be tested by automatic subprograms each time a schedule modification has been effected.

Testing aircraft constraints can be made easier by proposing a new aircraft following algorithms. Here the computer automatically (from schedule data) creates a small file on each aircraft tail number. It keeps track of the flight times scheduled for the specific aircraft assigned making sure no two segments or activities overlap.

It tests whether the specific aircraft can fly the scheduled payload/range requirement of each segment. The algorithm can also keep track of total flight time between maintenance overhauls and alert the scheduler when next scheduled maintenance work for the specific aircraft nears. With an aircraft tail following algorithm in place, the scheduler can modify the schedule with confidence knowing that all important aircraft related constraints are being constantly tested.

Similar in purpose and structure to the aircraft tail following algorithm is the crew following algorithm. Crews must be scheduled to cover all scheduled flight segments. Through staging and doubling of crews a feasible crew schedule can be found appropriate for a given flight schedule. The crew following algorithm would follow each individual crew through the schedule, calculating dutytime and rest time for each crew tour. The duty/rest constraints would be automatically rested by the algorithm while crew duty/rest aggregate times would be available for the scheduler to review. Both the aircraft following and crew following function are necessary in airlift scheduling. The automation of the processes can take advantage of the computer's ability to quickly run through the schedule database pulling out and aggregating information desired.

A station activity following function is also necessary in scheduling. The status of the station is updated at the time of each event. The station database holds all of this information at each event time. This is the primary purpose for the event based structure of the schedule database. When an event is added, deleted or modified in the station event list, the station status at the time and all subsequent event times are updated accordingly. When they are updated, the system tests the status against the appropriate station constraint list. All violations are immediately reported to the scheduler through flags.

These resource following algorithms are invoked automatically, returning any violations immediately to the scheduler. A scheduler will know instantly if a schedule modification idea were feasible. This interactive set of algorithms can speed scheduling time significantly.

As just described, the automation of scheduling subfunctions can speed scheduling. The automation of schedule constraint testing can improve schedule feasibility reliability, since in the manual scheduling the scheduler may have overlooked a constraint. The computer will not forget to test a constraint unless the scheduler did not advise the computer initially of the constraint. Computer automation can be used to speed the most trivial, yet necessary, subfunctions of the scheduling process yielding overall a significant benefit.

An example of a scheduling subfunction candidate for computer automation would be flight segment flying time calculation. The computer can assist the scheduler in two ways. First, the computer could store a large matrix of city pairs and their standard flying times. When invoked, the flying time subprogram would prompt the scheduler to enter the origin and destination stations interactively and then search the data matrix for the appropriate standard flying time. If the origin destination pair were not found the computer would attempt a second method of flying time calculation. From the station database, the subprogram would obtain the coordinates fo the stations and then calculate the great circle distance between them. Using the segment distance, aircraft type's cruising speed, an estimated wind factor for that area and season, and a standard time for both air maneuvering, the flying time could be calculated. Using either method of flying time calculation the subprogram would allow the scheduler to modify the flying time due to specific segment circumstances (e.g. aerial refueling or strong winds allowance).

In addition scheduling subfunctions, automation can be introduced for peripheral to the direct scheduling process. For example, the preparation and printing of daily, weekly, or monthly schedule reports could be handled by the computer. Summaries of past performance as well as forecasts for planned activities are arduous tasks when compiled by hand. Using schedule database interrogating subprograms, reports can be automatically generated for personnel requirements, fuel consumption, flight crew schedules, and a multiplicity of other airlift activities.

The automation of both direct and indirect scheduling function can assist and significantly speed the scheduling process. Schedule database interrogation operations such as constraint testing and report generation can be done accurately and quickly using appropriate database management programs.

5. ADVANCED SCHEDULING TOPICS

5.1 Introduction

The interactive dynamic aircraft scheduling system introduces a number of new and advanced topics to the scheduling process. A set of modern graphics displays are used to view schedule information at various levels of detail. Interactive symbol manipulation of those displays allow for quick and easy entering of schedule information or changes and features automated support on subfunctions such as constraint testing, flying time calculation, etc. Feasible automated decision support sub-systems are also introduced to assist the airlift scheduler as he seeks improved schedule, but it is simply not in the current state of the art to automate the entire mission generation and scheduling process, especially if optimal answers are expected. The challenge to this system design is to automate certain of the subprocesses in airlift scheduling and make them interactive with the airlift schedule.

5.2 Proposed Decision Support Functions

A number of decision support functions were introduced during the discussion of scheduling operations. This summary section extracts these support systems to briefly describe their focus and their requirements of working interactive software.

Mission generation is a scheduling process that could benefit from the assistance of an interactive algorithm. The algorithm would search through the task database and lump together requests similar in time and space creating candidate missions. No automated decision support algorithm to perform this function specifically exist but the development of one would not be difficult.

The scheduling process described created the need for a load reassignment algorithm which would load a given set of tasks of varying priorities, origin/destinations, etc. into an already complete schedule such as not to violate any payload constraints. Upon the arrival of a late request of higher priority, this algorithm would search the schedule to find the existing lower priority task(s), if any, which must be "bumped" to accommodate the new request without regenerating a new mission or schedule. This algorithm could be developed by modifying existing network flown algorithms and creating appropriate software to make it interactive with the airlift scheduler.

There exists a heuristic computer algorithm at MIT called "REDUCTA" which if made interactive with scheduler can assist scheduling. Given a set of missions within their time windows, the algorithm will minimize the number of aircraft required. It will not, however, solve the inverse problem of flying the most high priority missions, given a fixed number of aircraft. Support software could be designed to present changes in times, mission segments, etc. from the current solution when the algorithm is invoked for global minimization of the number of aircraft. The final acceptance or rejection of the optimized schedule solution would be the scheduler's responsibility since the algorithm's solution can guarantee feasibility but cannot be suspicious of operational on time performance probabilities.

Another computer algorithm which when made interactive with the system could provide some automated decision support is the MIT Fleet Routing (FR-4) algorithm. It can solve the problem of finding the missions of total highest priority value flown by a fixed number of aircraft, given a set of missions with their commited times. This algorithm can be used on a large fleet of aircraft, but cannot handle time windows, thus limiting its true optimization capabilities.

5.3 Advanced Algorithm

Although the knowledge of an experienced scheduler can never be duplicated by a computer, decision support algorithms can significantly improve the dynamic scheduling process especially when the scheduler does not have the time to explore many rescheduling possibilities. The proposed modified decision support algorithms based on REDUCTA (minimizing aircraft required by sliding missions within their windows) and FR-4 (maximizing carriage of high priority tasks, given fleet size) are helpful individually but if both optimization procedures were to be combined and solved simultaneously the resulting algorithm would be an extremely powerful tool for the airlift scheduler. This would bring the decision support role in scheduling one giant step closer to full scheduling capability.

The generation of crew mission sequences given a set of aircraft mission sequences is yet another scheduling problem for which there is no known optimality algorithm. Present crew scheduling procedures used by some airlines are very complex, requiring excessive amounts of computing time and power making them poor candidates for use as an interactive tool. A future development of a crew scheduling algorithm complete with crew staging planning would constitute a further improvement to the airlift scheduling system.

The scope of algorithms proposed for the new interactive dynamic scheduling system are by no means limited to those described in these chapters. The system can incorporate new algorithms in the future as they are developed and improved. Algorithms could be developed for more peripheral scheduling functions. For example, an interactive quickest delivery algorithm could be used to find routings for mechanics and spare parts given a schedule.

5.4 Advanced Topics

The scheduling of aircraft missions and crews has been the primary purpose of the new system design. The storage of dynamic aircraft crew and station information by the computer can be used for automated resource status and forecast reports generation. Other airlift functions can be made part of the overall system as well. For example, the system could eventually be used to schedule the station resources, aircraft maintenance and more.

The flexibility and ability to accept new advancements in scheduling is key to the long term effectiveness of the proposed system. An important airlift capability, not discussed directly in the previous chapter, is the use of connecting traffic. Here a payload may depart its origin on one mission, be offloaded at a transshipment point (a connecting station) and then reloaded on another mission which takes it to its desired destination. Thus, two or more missions must be properly timed when scheduling connecting traffic. New and more complicated constraints must be stored for each connecting payload. Whenever a mission is rescheduled (e.g. slid in time) the system must test that all desired connections to and from that mission are still feasible. The database must be designed to accept those connections and their constraints. Graphics displays may need symbols added representing connecting traffic information. Decision support algorithms for connecting traffic could be developed for the system. This is but one of many examples of additional function which could be added to the interactive dynamic scheduling system. Thus, flexibility in its original design is a necessity for further improvements.

6 CONCLUSION

A feasible design for an interactive dynamic scheduling system for aircraft has been presented. Incorporating current computer technology in screen graphics and database manipulation, the proposed system is a very attractive tool for the (human) aircraft scheduler.

The system design features a detailed set of modern graphics displays which can concisely present to the scheduler all event, resource and constraint information necessary for the complex decision making processes in aircraft scheduling. The manipulation of the graphics display symbols results in appropriate schedule database changes. Interactive schedule modification and information searching commands can both speed and simplify the scheduler's work. Several scheduling subprocesses and decision support functions are eligible for automation and the introduction of flags and alerts in interactive scheduling are valuable in speeding the search for feasible schedules. A quick graphics response can be shown.

The design of this interactive dynamic scheduling system outlines both computer hardware and software needs. High resolution computer graphics capabilities are an essential system requirement. A relational schedule database with pointers must be used to handle the dynamics and complexities of schedule information storage efficiently. The interface between the graphics screen displays and the detailed schedule database is a crucial software challenge. It requires the coexistence of modern computer graphics software, interactive command recognition software, database manipulation software and decision support algorithm software and decision support algorithm software. System flexibility is essential for handling aircraft scheduling complexities and future system upgrade.

Further system development would include the specific design of the schedule database, final graphics displays, and system software to achieve a fully operational demonstration scheduling system. Users' acceptance and display preferences program response time, and overall system scheduling efficiency and speed could be tested from field development projects.

REFERENCES

Lubow, Bruce C Aircraft Scheduling: An Interactive Graphics Approach. M.S. Thesis, Massachusetts Institute of Technology Jan 1981.

Mathaisel, Dennis F X, Simpson, Robert W New Systems for Scheduling Airlift in the MAC C^2 Upgrade Program. working paper, MIT Flight Transportation Laboratory, May 1984.