Discussed are design principles, file structures, and programming techniques of a scheduling system that approximates the overall magnitude and complexity of an airline scheduling system.

Used worldwide by the National Aeronautics and Space Administration to schedule its manned and unmanned space flight missions up to one year in advance, the system operates in either batch or interactive modes to produce, modify, and observe actual and simulated schedules.

Interactive scheduling system

by A. C. Brewer

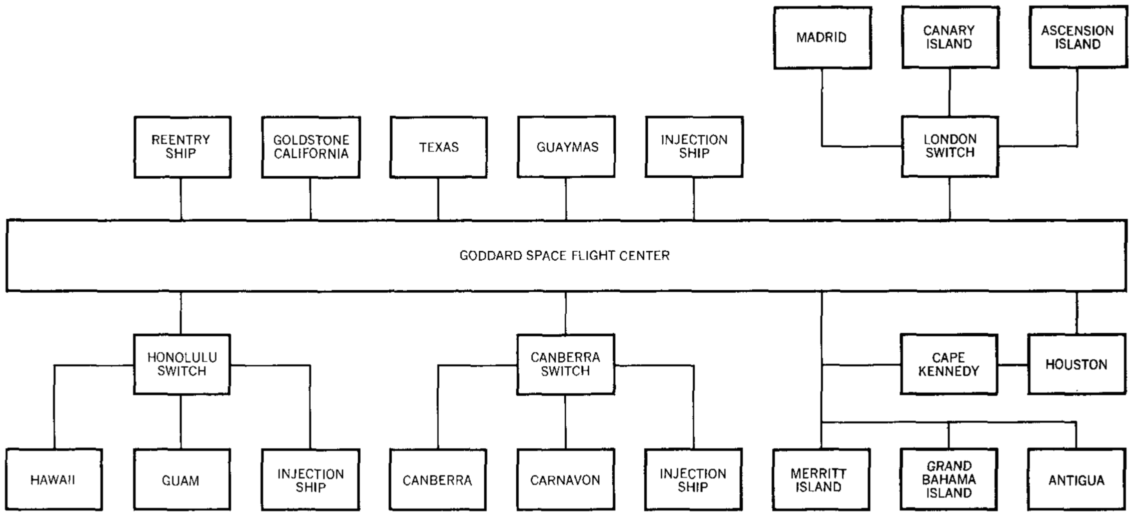
Scheduling the worldwide facilities of the NASA Manned Space Flight Network (MSFN) is a task approximating the overall magnitude and complexity of airline reservation systems,[1] but with functional specifications and user requirements that demand a different approach. The author discusses in this paper system design principles, file structures, and programming techniques developed under contract with the National Aeronautics and Space Administration (NASA) for scheduling the MSFN. The system supports both manned and unmanned missions up to one year in advance of the current date without conflicts. Termed Computer Assisted Network Scheduling System (CANS), operation is in either batch or interactive modes for producing, modifying, and observing the effects of actual or simulated changes. Simulation, or the ability to answer the question "What if?" and to take simulated action analogous to game playing, is a major contribution of the CANS system.

Presented first are the overall system design principles, which parallel the existing structure and inventory of the Manned Space Flight Network. Discussed next are the four system files: schedules, inventory, events or actions, and inputs together with their control and data flow. Since many user applications differ from the example discussed in this paper, system programming principles and techniques are emphasized.

scheduling and planning

Resource scheduling by the system discussed in this paper is the allocation of organizational resources (manpower, equipment, time,

Figure 1 Manned Space Flight Network communications



money, etc.) to achieve planned objectives. Thus scheduling is a part of the overall *planning function* whereby detailed action plans are produced.[2,3] Scheduling as discussed in this paper involves no technical dependence of one event upon another and therefore contrasts with critical-path methods.[4] In the critical-path method, which is a form of scheduling, time and technical dependence determine the sequencing of events. In the methods discussed in this paper, priorities and required resources determine the sequence of events. Our scheduling procedures specify current or proposed actions based on priorities, whereas critical-path techniques may be used for sequencing an overall planning function for a longer period of time based on technological precedence.

network and requirements

Examples of resources available for scheduling can be vehicles, aircraft, trains, machinery, people, dollars (budget, cash, and cash flow), workload or whatever is basic to the mission or goal of the organization. In the case of the Manned Space Flight Network, as shown in Figure 1, the manned and unmanned space flight operations require world-wide ground support facilities to track, transmit, and receive data, and to provide voice, telemetry, and command communications for spacecraft. There are between twenty-seven and thirty facilities in the network. Up to one hundred pieces of equipment are availiable for scheduling at each station in the net. Figure 2, taken from the equipment inventory file, shows the items available for scheduling at Ascension Island (ACN) and Bermuda (BDA).

The demands against network resources are of five major types: launches, orbiting spacecraft support, tests (such as communications and performance), maintenance, and modifications. Some

Figure 2 Equipment inventory file

NAME	EQUIP	QUAN	EQUIP	QUAN	EQUIP	QUAN	EQUIP	QUAN	EQUIP	QUAN
ACN	USB ANT	1	USB REC	2	TDP	1	APP	1	RANGE	1
	TIME	1	R/E	2	P/A	1	VER REC	1	SERVO	1
	TV	1	DECOMM	3	VHF TLM	12	PCM SIM	1	FM	1
	642B-2	1	642B-1	1	1218	1	MTU	2	EMU	1
	DTU	101	HSP	1	CAM	1	A/C	101	UHF CMD	1
	M-22	2	ACQ AID	1	TTY	2	VOICE	3	HSD	3
	MODEMS	3	FEC-500	101	I/COMM	101	A/G	1	SATCOM	1
	M-25	1	S/C RCDR	8	FR-100	1	ASE	101	V/D	1
	USB SYS	1								
		WORK	START=0700		WORK STOP=1700					
BDA	USB ANT	1	USB REC	2	TDP	1	APP	1	RANGE	1
	TIME	1	R/E	2	P/A	1	VER REC	2	SERVO	1
	TV	1	DECOMM	4	VHF TLM	15	PCM SIM	1	FM	1
	642B-2	1	642B-1	1	1218	1	MTU	2	EMU	1
	DTU	101	HSP	1	CAM	1	A/C	101	UHF CMD	2
	M-22	2	ACQ AID	3	TTY	2	VOICE	4	HSD	4
	MODEMS	4	FEC-600	101	I/COMM	101	A/G	1	FPS-16	1
	FPQ-6	1	VR-3600	1	FR-100	1	S/C RCDR	18	ASE	101
	DQM-1	1	USB SYS	1	V/D	1				
		WORK	START=1200		WORK STOP=2000					

events require all equipment at a station at a given time. Bermuda immediately after an Apollo launch is an example of such a dedicated station. The traverse of a station by an unmanned satellite may not require all the equipment at that station. Thus the remaining equipment is available for other work. The control center for the network is the Goddard Space Flight Center at Greenbelt, Maryland.

scheduling algorithms

Fundamental to all scheduling problems is the concept of the *scheduling period*, which is the time during which the planned action is to take place. Additionally, one must specify either the time the action must begin or terminate. Also critical to deriving a conflict-free schedule is the *time scale* of the planned action. An example of a short but critical action is command and data transmission while a spacecraft is within range of a given station. Such a short period must be minutely examined for possible conflicts. The scheduled period of transmission being typically measured in the scale of minutes, conflict-free scheduling involves a second-by-second scan of the activity of the required equipment at each station involved. On the other hand, the period of an Apollo mission could exceed thirty days, thereby requiring five-minute scans for every major action, for every station, for every orbit. Thus in all time scales, the compounded factors are approximately constant, i.e., scans per station per equipment per unit time might approximate 900,000 for each action. If a conflict is detected, analysis and additional scans must be made to resolve the conflict.

A scheduling algorithm might shed some light on ways of reconfiguring a system for greater efficiency. However, objective efficiency may not be a feasible goal of a system. Astronaut safety and meeting mission goals are objectives of the MSFN that override a theoretical network goal of optimum utilization of equipment. Schedulers often observe that the actions to be scheduled have evolved to meet human needs whether these actions be space flight, school bus routing, or delivering mail. In designing a scheduling system, one may thus be required to model his system after the existing system rather than strive for maximum efficiency and optimization.

For optimization, linear programming techniques may contribute to efficient scheduling. In such a system, the user specifies the work to be accomplished, and the algorithm determines the optimum schedule. For example, linear programming has been effectively employed in designing job schedulers for operating systems.[5]

In the problem we are considering, however, the user specifies the desired schedule, and the algorithm determines its feasibility. Beside time factors in schedule feasibility is the priority of the desired action, which is ranked from a high of 1 to a low of 9 in our algorithm. The effect of priorities is that although a proposed action may be feasible from a time optimization point of view, the mission cost may be prohibitive.

In the scheduling system discussed in this paper, optimization is effected after feasibility and priority considerations. The system also accommodates substantially non-priority actions whereby the user may specify a number of acceptable start times, including "don't care." Equipment malfunctions are treated as new actions and are assigned an emergency priority, which is higher than that of any other action.

System design

An interactive, computer assisted network scheduling system (CANS) was developed that applies the preceding modeling, optimizational, and priority requirements to the problems of the Manned Space Flight Network. The resultant system is a System/360 application program that operates under and uses the facilities of the System/360 Operating System (OS/360) having the option of multiprogramming with a variable number of tasks (MVT). With some modifications, the CANS program can operate in a multiprogramming environment with a fixed number of tasks (MFT). The system may be operated either interactively in a time-sharing mode using remote terminals or as a job in the batch mode under OS/360. Source languages for CANS are FORTRAN (75%) and Basic Assembly Language (25%). The CANS program generates a conflict-free schedule of actions (or events) for a specified period for up to one hundred pieces of equipment at each of one hundred stations. In general, the system accepts scheduling requests, organizes these requests into time-ordered groups by priority, and then creates the best schedule. The program is discussed further in the last section where we describe the functions and modes that the system is designed to perform primarily in terms of the charts and reports that CANS produces.

Figure 3 Teletype schedule printout

```
A.   NCG              TITLE
B.   START            END                   T-0                        WINDOW
C.   STATIONS                          START          END
D.   CONDUCTOR              TELEPHONE
E.   EQUIPMENT
F.   COMMENTS

A.   NCG-335          METRIC DATA FLOW TEST
B.   16/1305Z         16/1435Z
C.   BDA                               16/1336Z       16/1356Z
     VAN                               16/1335Z       15/1355Z
     GSC
     NOC
D.   DNC
E.   HSD          VOICE        FPS-16       FPQ-6
F.   REF NCG-093 NUP.
```

The system discussed in this paper is executed by System/360 Model 95. However, CANS can be used in any System/360 with at least 300,000 bytes of main storage and sufficient direct access storage to accommodate data files of 7.2 million bytes, load modules of 0.39 million bytes, and one hundred control programs and system subroutines.

The CANS system does its own information management and retrieval in performing its four primary functions:

- Creation of operational schedules
- Modification of operational schedules
- Production of simulation schedules
- Modification of simulations

The creation of operational schedules by station and equipment for the worldwide network for selected scheduling periods is the primary function of the system. The program accommodates long-range scheduling up to several years into the future. The programming principles would be the same for other applications, such as airline operations, which might use other scheduling periods. In creating operational schedules, the CANS program performs a conflict analysis of the requested action against the schedule file. The program then stores the new schedule in the file and produces reports on all requested conflict-free, feasible actions. Requested actions are also related to manpower, local times, and working hours, and the requester is notified of extra-shift requirements necessary to take the requested action. Five types of reports are created.

Teletype formatted schedules are produced in both hard copy (a sample of which is shown in Figure 3) and punched Teletype tape for direct transmission to all stations. Alphabetic designations in the left column indicate the action to be taken, and they are keyed to the legend heading the figure. Affected stations are listed in the first column to the right. All stations receive all schedules, and the local manager has only to scan the listing to determine when and what equipment at his station is affected. Times are given in Greenwich Mean Time, but local time has been taken into account by the program.

Resource utilization charts such as is shown in Figure 4 for the Bermuda station are generated on request by date and station. A line of numbers (4's in this case) which are understood as points on a bar graph, indicate the hours and quantity of the equipment required. In the example, one of the 15 available VHF-TLM (very-high-frequency telemetering equipment) is being used from 1200 Z to 2000 Z. Note also that for test NCG-381, four decommutators (DECOMM) are used from 2030 Z to 2130 Z.

Schedule bar charts, one of which is shown in Figure 5, can also be produced on request. These charts display actions (by name and number) that occur during specific time periods and the stations involved in the action.

Activity reports, a part of which is shown in Figure 6, list actions for a specific time period and the stations involved.

Conflict analysis reports are produced during the schedule generation to indicate a previously scheduled test or action that conflicts with one the user is requesting. Figure 7 shows that on January 3, 1971 at three o'clock a conflict occurs between an antenna scheduled for NCG-100 and requested test NCG-210. The system had tried other times between twelve and five o'clock P.M. and would have printed alternative times within that period. No conflict-free start times having been found, the requester may assign a higher priority to his action, a new time (hour), or a new date. Of course he can

Figure 4 Resource utilization chart

```
SITE = BDA
                                           *-----------------------------------------------------------*
                                           *    QUANTITY UTILIZED       ---      GREENWICH MEAN TIME    *
                                           *----+----+----+----+----+----+----+----+----+----+----+----*
EQUIPMENT    QUANTITY    TEST USING        * 0  * 2  * 4  * 6  * 8  *10  *12  *14  *16  *18  *20  *22  *
  NAME       AT SITE     EQUIPMENT         *   1*   3*   5*   7*   9*  11*  13*  15*  17*  19*  21*  23*
                                           *----+----+----+----+----+----+----+----+----+----+----+----*
USB ANT         1                          *    *    *    *    *    *    *    *    *    *    *    *    *
USB REC         2                          *    *    *    *    *    *    *    *    *    *    *    *    *
TDP             1                          *    *    *    *    *    *    *    *    *    *    *    *    *
APP             1                          *    *    *    *    *    *    *    *    *    *    *    *    *
RANGE           1                          *    *    *    *    *    *    *    *    *    *    *    *    *
TIME            1                          *    *    *    *    *    *    *    *    *    *    *    *    *
R/E             2                          *    *    *    *    *    *    *    *    *    *    *    *    *
P/A             1                          *    *    *    *    *    *    *    *    *    *    *    *    *
VER REC         2                          *    *    *    *    *    *    *    *    *    *    *    *    *
SERVO           1                          *    *    *    *    *    *    *    *    *    *    *    *    *
TV              1                          *    *    *    *    *    *    *    *    *    *    *    *    *
DECOMM          4        NCG-381           *    *    *    *    *    *    *    *    *    *    *  44*4    *
VHF TLM        15        NCG-273           *    *    *    *    *    *    *1111*1111*1111*1111*1   *    *
                         NCG-331           *    *    *    *    *    *    *    *    *    *    *  11*1    *
PCM SIM         1                          *    *    *    *    *    *    *    *    *    *    *    *    *
FM              1                          *    *    *    *    *    *    *    *    *    *    *    *    *
                                           *-----------------------------------------------------------*
```

also cancel, but the previous schedule stands until a revised request is submitted and resolved in a conflict-free manner. As an aid in resolving such a conflict, a *conflict analysis summary report* is produced, giving alternative conflict-free start times. Especially useful in resolving more complex conflicts, a *detailed analysis report* can be produced that lists all stations, equipment subsystems, and other events involved in a conflict and the items in conflict.

Figure 5 Schedule bar chart

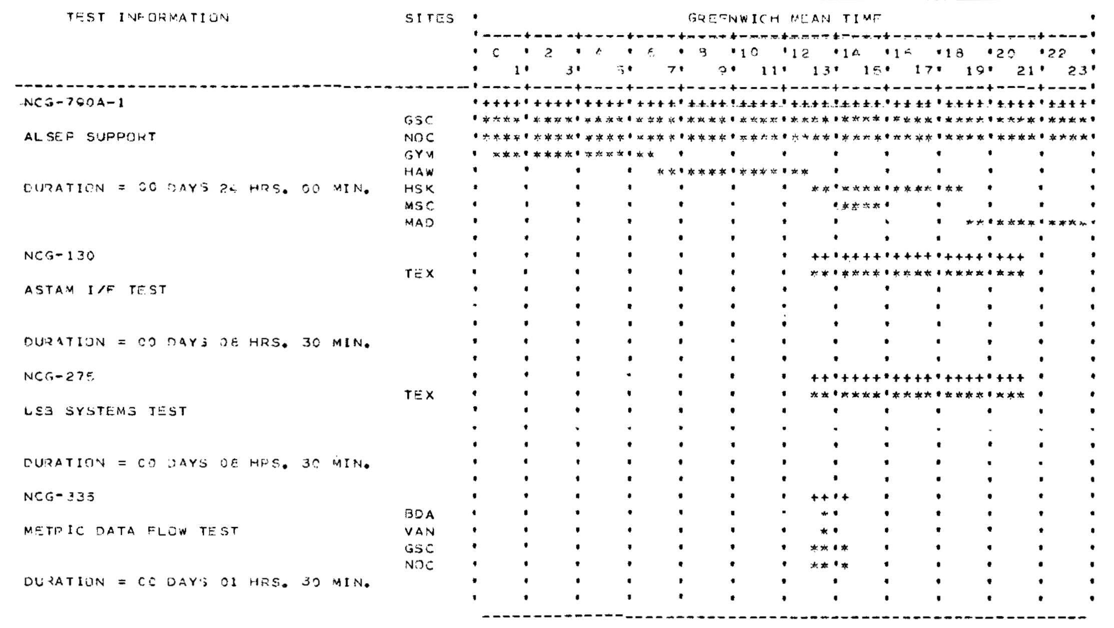


Figure 6 Activity report

UNMANNED ORBITAL TRACK TESTS SCHEDULED

DATE	TEST NO.	TEST TITLE	SITES USED		START TIME
JUL 15, 1970	NCG-790A-1	ALSEP SUPPORT	GSC HAW MSC	NOC HSK MAD	0000Z
JUL 15, 1970	NCG-381B	(ETR-1093) OV1-18 ORBITAL SUPPORT	GSC NOC	ETR CRO	0920Z
JUL 15, 1970	NCG-335	METRIC DATA FLOW TEST	BDA GSC	VAN NOC	1435Z
JUL 15, 1970	NCG-381B	(ETR-1093) OV1-18 ORBITAL SUPPORT	GSC NOC	ETR BDA	2035Z
JUL 16, 1970	NCG-790A-1	ALSEP SUPPORT	GSC GYM HSK MAD	NOC HAW MSC	0000Z

Figure 7 Conflict analysis report

CANS406 - DETAILED CONFLICT ANALYSIS

TEST ANALYZED	START TIME	SITE	UNIT OF GEAR	CONFLICTING TEST(S)	TIME OF CONFLICT
NCG-210	7101031500	ACN	USB ANT	NCG-100	7101031500

The CANS program performs modifications of operational schedules by modifying the schedule file, which might typically be required by changes of operating plans. As is done in producing operational schedules, analysis is performed, and upon request, the program produces all scheduling reports. Interactive schedule modification can be performed rapidly because only the segment of the schedule file being modified is operated upon. In most cases, the effect can be immediately observed.

simulation

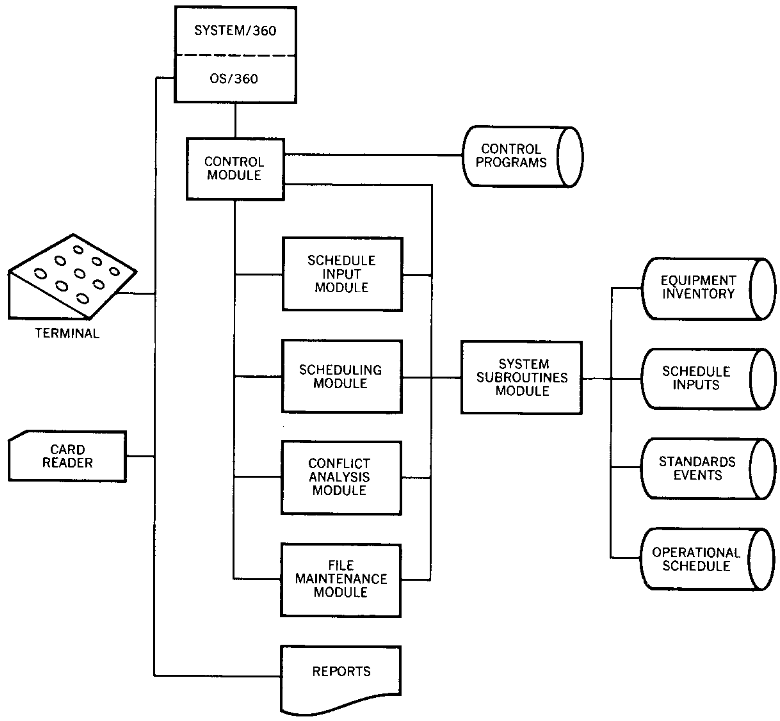
Identical from the user's viewpoint to operational scheduling is simulated scheduling, which permits the user to find out "what would happen-if" without disturbing the permanent files. In this mode, schedules can be proposed and their effects observed, thereby permitting management game playing. All functions of operational scheduling previously discussed are performed. Internally, however, the simulated schedule is not placed in the schedule file. Since scheduling is simulated interactively, the effect of these simulations goes beyond mere scheduling. For example, system capacity or the number and type of actions that an airline, say, could take using existing resources could be simulated. In general, management could learn much about the efficient and effective utilization of available resources by simulated scheduling. By extension, management could simulate the effect of additional resources in meeting current or future demands. The impact of changes in resource allocation, configuration, increase, and reduction can be observed without interfering with current operations.

Besides operational and simulated scheduling and their modifications, the CANS system performs information retrieval and file maintenance functions. File operations performed by CANS are: creating new files, adding file entries, changing data in existing entries, and information retrieval. The files themselves are discussed in the following section.

System files

The data base for the computer assisted network scheduling system consists of the four direct-access storage files shown in Figure 8: equipment inventory, schedule inputs, standard events, and operational schedules. Equipment inventory files are maintained for up to one hundred stations and up to one hundred units of equipment available for scheduling at each station. Equipment information

Figure 8 The CANS system



includes description, quantity available, and current status. Equipment and stations not available for scheduling are indicated as "down" and are given emergency priority during a specified period and are not considered for scheduling until the down status is removed.

In principle, each piece of equipment at each station can be scheduled independently at any time. In practice, of course, certain missions imply the combined use of certain pieces of equipment. Therefore, both the schedule inputs and standard events files specify both the necessary combination and the time requested. The input file is a temporary data set that contains all the necessary scheduling information used by the schedule program module to be discussed later in this paper. Formatted like the schedule file except that start and stop times and dates are left blank, the standard events file saves data preparation and processing time for recurring events. The user only inserts the name (or number) and the request period to have a schedule prepared.

Conflict-free schedules are produced by matching schedule inputs or standard events against the inventory file. The resultant schedule is stored in the direct-access schedule file. The input and comparison functions are performed by the schedule input and schedule program modules. For efficiency, conflicting actions are passed to the conflict analysis program module, and the schedule module continues until the scheduling function is completed.

The user interacts with the CANS system through a simple, conversational input and control language, which is not discussed in this paper. Basically, the language gives the system (1) the control information that directs the information flow through the system and (2) the data information that gives the requested event or file maintenance data.

Programming techniques

The CANS program is segmented into six functional program modules:

- Control
- File maintenance
- Schedule input
- Scheduling
- Conflict analysis
- System subroutines

control

The control module directs the execution of the CANS system using parameters of the control data entered either from input cards or manually at terminal keyboards. Control information is read and interpreted, and the system begins performing the requested function. After receiving control from the operating system, the control module initializes variables, accesses system file record numbers, and then reads a control entry from the input stream. If there is no input, or if an end of data is indicated, the file record numbers are rewritten and execution terminates. Otherwise, the entry is read and interpreted.

Error checking is also accomplished by the control module. Whenever an error condition is detected in the input stream, the control module causes the system to enter and remain in a "scan" mode of operation. In this mode, all succeeding control and data entries in the input stream are checked for errors, an error report is printed, and the run terminates. The scan mode allows for faster job turnaround since all apparent errors in a given job are detected by one run. The alternative would be to terminate after the first error, which would miss any succeeding errors.

If all inputs are correct, the control module passes job control and sequencing to the appropriate module(s) until the requested function is completed. At this time, another input control entry is read, and the process is repeated. There are various points in the sequence of execution of other modules, at which the control module performs housekeeping activities on the system files and data sets.

file maintenance

The file maintenance module performs four basic functions on the permanent CANS data files (schedules, standard events, and equipment inventory):

1. Adds new entries
2. Modifies data in existing entries
3. Deletes entries
4. Retrieves and prints information

When the file maintenance module receives control from the control module, indicators are set in system common locations that specify the function to perform. This indicator information is taken from a control entry that is read and interpreted by the control module. In addition to these indicators, there may or may not be data entries for the file maintenance module to process. If the module is to perform an ADD, MODIFY, or DELETE function, there are always data cards to process. If the module is to perform a PRINT function, there may or may not be data to process. When there are no data, the indicators in system common locations are sufficient to inform the module which entries should be included in the printout.

The ADD function is performed by the file maintenance module for the major system functions of creation, simulation, and modification of schedules. A schedule entry is used whenever entries are to be added to the network schedule file; a file-add entry is used when entries are to be added to one of the other files in the system.

Data entries contain information for one or more entries to be added to a particular file. The module first reads and interprets the information for one entry, and at the same time it makes some preliminary error checks on format and field lengths. Information from the data entries is set up in two internal tables, the seek table which holds control information, and the data table which holds data to be entered into the specified file. These two tables are inputs to a group of programs responsible for setting up the entry in the file. When the entry is set up, it is checked to determine whether the information can be used for scheduling, and three cases can result: (1) there are no errors in the entry; (2) there are minor errors, but the entry can still be used for scheduling; and (3) there are major errors, and the information cannot be used for scheduling. If there are no errors, the information is printed in a fixed format to show the user what he is adding to the file; also, a system subroutine writes the entry into the file. If there is a minor error, such as one of the items of equipment does not appear in the inventory file, a warning message is printed. The entry is printed out and written in the file. If there is a major error, for example no event number is given, the error message is printed, and this entry is not added to the file. After the processing for one entry has been completed, a check is made to determine whether more data entries are to be processed.

If so, the procedure is repeated; if not, a return is made to the control module.

When the MODIFY function is specified, two types of information must appear in the data: information to find the entries in the file to be modified, called "find data," and information indicating what modifications are to be made, called "set data." In the find data, certain fields are specified. Data for these fields from the input entries are matched against data for fields from the file entry. If they do not match for all fields given, the file entry is not to be changed. If they do match, the entry will be changed according to the specifications in the set data. The processing for the MODIFY function is performed in a manner similar to the processing for the ADD function. The data are interpreted, and internal tables are set up. These tables are used to locate and modify the entries in the file. Verification and printouts for the modified entries are the same as those for the ADD function.

If a field in a network schedule file entry is modified and that modification creates a conflict in the schedule, an indicator is set to inform the control module that the conflict analysis module must be called. This case differs from other modifications in that only one entry may be modified and checked for conflicts.

The DELETE function is performed by the file maintenance module after a file-delete control entry has been processed by the control module. When this function is specified, only "find" data appears in the data cards. The processing of the DELETE function is similar to that for the MODIFY case, except that when an entry is found that satisfies the find-data condition, it is eliminated from the file.

The PRINT function is performed by the file maintenance module when file-print and schedule-print conditions occur in the control card. This function is performed for schedule creation, simulation and modification operations.

In the file-print and schedule-print cases, there may or may not be data entries, and in scheduling and modification cases there are never data entries. The information on the data entries can only be the find-data type, which specifies those entries that are to be included in the output. If data entries are not included after the file-print entries, it means all the entries from the specified file are to be printed. If data entries are not included after a schedule-print command, or in scheduling and modification, it means that only entries from the network schedule file that have start times between the two dates given in the schedule entry are to be included in the printouts.

Processing for the print function starts by checking whether there are data entries. If there are, these are interpreted and internal tables are set up. When this is completed, or if there are no data entries, a branch to the output section is made. The requested outputs are then determined, and programs are operated in sequence to produce the various outputs. As these programs read in the entries from the file, a check is made to determine whether the entry satisfies the conditions given on the data entries. If it does not, the entry is not used in the output. If the entry does satisfy the specified conditions, it is formatted and printed.

schedule input

During either an actual or a simulated schedule preparation, the control module calls the *schedule input module*, which prepares action scheduling data for transfer to the schedule module (to be discussed in the next section). Control returns to the control module, which calls the schedule module to perform the scheduling functions.

Inputs to the schedule input module come from four sources:

- New requests for a scheduling action from the job stream
- Previously scheduled actions from the schedule file
- Required resources extracted from the equipment inventory file
- Standard data from the standard events file

The output of the schedule input module is a temporary data set consisting of scheduling data, which is a time-ordered series of requested actions, together with other necessary information such as start times, duration, priorities, and the equipment required for each action.

To build this data set, the input module arranges data for the schedule module in fixed order. The first (highest) priority category is that of equipment scheduled as "down" for the duration of the requested scheduling period. The second through eleventh categories are previously scheduled actions with priorities one through nine that were scheduled in the last scheduling run, and that are contained within the present scheduling period. The twelfth through the twentieth, and last categories are actions with priorities one through nine that have not previously been scheduled.

Within each category, actions are ordered according to the numbers of start times specified by users, with those actions having fewer numbers of starts being placed ahead of those having greater numbers of starts. Because of schedule module main storage limitations, actions within a category are put into a temporary data set on direct-access storage (usually disk storage). All of the requested actions in a category may fit into one group, or the requested actions may have to be separated into several groups. The size of the scheduling group is determined by the amount of main storage allotted to the schedule module. The input module places the scheduling groups onto the data set in order of highest priority first, so that the schedule module can schedule the requested events by groups as they are read into main storage from disk storage.

Requested actions are ordered in the manner just described so that generated schedules satisfy the following requirements:

- Priority is specified by the user from 1 (highest) to 9 (lowest).
- Possible start times are requested by the user, up to a maximum of ten, with the first preferred.
- "Down" equipment is not scheduled.
- Previously scheduled actions have priority over new actions.
- High priority actions are not excluded because of low priority resource requirements.
- Maximum number of conflict-free actions is scheduled.

The input module has completed its task when all actions have been grouped according to priority and their resource requirements have been chain linked and written in the schedule input file. Control then returns to the control module.

scheduling

The objective of the schedule module is to seek the best combination of actions and start times, so as to designate as "scheduled" all the events and start times that are free of conflict. Input to this module is data formatted by the schedule input module, which includes the number of scheduling periods, the number of stations in the network, an array that represents all the network equipment available, and a directory to this array. Another input is a set of logical blocks, each of which contains a group of actions to be scheduled. Start times, stop times (duration), and resource requirements are given for each proposed action in a scheduling group. These scheduling groups are entered according to their priority (the first group having highest priority and last group having lowest priority). The output of the schedule module is a list of action numbers and start times for each action designated as "scheduled." These are entered into the schedule file. All requests not scheduled due to conflicts or other causes are passed to the conflict analysis module for detailed analysis.

The schedule module produces the time-ordered listing actions by determining the best combination within each scheduling group, and then forms the union of these combinations. For a given scheduling group, all feasible combinations (i.e., actions that do not cause resource conflicts) and their start times are generated in the following manner. All possible combinations of actions with start times within the scheduling period are generated. The total resources required by each combination of actions in this period is then compared to the total resources available. All combinations (i.e., nodes) that require more resources than are available are eliminated. Nodes from the preceding period are also checked, and those that are unfeasible for the preceding period are also discarded. The remaining nodes from the preceding period and the feasible combinations generated for the current period are matched, and all feasible unions of nodes in these two periods are retained.

Table 1 Example inputs to CANS scheduling algorithm

Action	*Start period*	*Stop period*	*Cost*	*Equipment*
A	2	3	3	X and Y for duration of action
A	4	5	1	X and Y for duration of action
B	2	5	1	Z for duration of action and Y in last period of action
C	1	3	2	X and R for duration of action

The resultant set of nodes consists of the feasible combinations generated for the current period, the feasible combinations remaining from the previous periods, and the feasible combinations obtained by matching pairs of nodes in the preceding two groups. This process is repeated for every scheduling period. The feasible combinations remaining at the end of the last scheduling period then comprise the set of all feasible combinations of events in the scheduling group.

example

As an example of the CANS scheduling algorithm, assume that there are five scheduling periods and three events A, B, and C in the scheduling group with start periods, stop periods, costs, and equipment requirements as given in Table 1. (Costs are generated according to predefined parameters and decision rules used by NASA.)

We assume that there is no combination of conflicts with previously scheduled equipment, and define the following three categories of nodes:

- Set 1 are the feasible combinations generated at the start of the current period.
- Set 2 are the combinations remaining from the preceding period.
- Set 3 are all feasible combinations at the end of the current period.

CANS generates nodes for each period as shown in Table 2, wherein subscripts denote start periods.

When all combinations of actions in the scheduling group have been generated, the best combination is determined as the node with the greatest number of actions. If several "best" combinations have been generated (i.e., having equal numbers of actions), the one with the lowest cost is selected. In the example A_2B_2, B_2C_1, B_2C_3, and A_4C_1 all satisfy the best-combination criterion. Since their relative costs are 4, 3, 2 and 3, respectively, B_2C_3 is finally designated.

Table 2 Nodes generated by CANS for the example inputs

Period	*Set*	*Nodes*
1	1	C_1
	2	None
	3	C_1
2	1	A_2, B_2, A_2B_2
	2	C_1
	3	A_2, B_2, A_2B_2, C_1, B_2C_1
3	1	C_3
	2	A_2, B_2, A_2B_2, C_1, B_2C_1
	3	C_3, A_2, B_2, A_2B_2, C_1, B_2C_1, B_2C_3
4	1	A_4
	2	C_3, A_2, B_2, A_2B_2, C_1, B_2C_1, B_2C_3
	3	A_4, C_3, A_2, B_2, A_2B_2, C_1, B_2C_1, B_2C_3, A_4B_2, A_4C_1, $A_4B_2C_1$
5	1	None
	2	A_4, C_3, A_2, B_2, A_2B_2, C_1, B_2C_1, B_2C_3, A_4C_1
	3	A_4, C_3, A_2, B_2, A_2B_2, C_1, B_2C_1, B_2C_3, A_4C_1

After the best combination has been determined, the actions in that combination are designated as "scheduled" by adding their action numbers and start times to the array of scheduled actions. The next scheduling group is then entered into the schedule module, and the algorithm is repeated.

When an action has been scheduled, succeeding scheduling groups do not affect its status. However, the equipment required by scheduled actions is not available to actions in succeeding scheduling groups.

After all the scheduling groups have been considered, the start time for each action in the array of scheduled actions is placed in the schedule file, and the schedule module returns control to the control module.

conflict analysis

The function of the conflict analysis module is primarily to identify conflicts and the actions involved. Of course, any action for which there are no equipment conflicts is scheduled. In case of conflict, the conflict analysis module first seeks alternative start times. Messages are printed giving suggested alternative start times or the conflicting action, station, and equipment. The conflict analysis module then checks modified actions that could also introduce conflicts into the network schedule file.

The conflict analysis module receives control either during actual or simulated runs in which primary action scheduling is accomplished by the schedule module. During the modification of either type of run, the schedule module is bypassed and scheduling is effected during conflict analysis. In actual or simulated actions, the inputs to be analyzed are those actions that could not be scheduled because of conflicts. For modifications, each input action is received directly from the input stream for addition to a schedule if no conflict exists, or for analysis if conflicts do exist. When modifying a simulation, conflict analysis has the effect of determining whether the action fits into the schedule, but no scheduling is performed. A third case in which conflict analysis receives control is that of file modification, in which the module validates changes to the network schedule file.

For each action analyzed, the start times provided are considered one at a time in order of preference. Equipment required by proposed but conflicting actions is retrieved, subsystems of equipment are broken down to unit requirements, and the total resources inventory of the network is retrieved. Conflict analysis then proceeds through the action and examines the utilization of each type of resource at each station involved, and identifies and records each conflict that occurs.

If at least one conflict is identified, the process is repeated for each start time until all start times have been treated or until a start time is found to involve no conflicts. If a conflict-free start time is identified, and the conflict analysis module has not been called by a simulated schedule modification, the action is added to the schedule for that start time. A message is created if execution is for a modification run, and the conflict analysis module proceeds to the next event to be analyzed. For runs other than simulation runs, all conflict-free start times are accumulated for later output.

If no proposed start time for an action is found to be free of conflicts, the module attempts to locate alternative start times at which the action could be scheduled. This is accomplished by defining an appropriate time span about each given start time, and designating certain times within that span as alternative start times. Each alternative start time is analyzed in the same way as the original start times. When this process has been completed for all the original start times, the conflict analysis module prints analysis messages for the original proposed start times, and for conflict-free alternative start times that were located. Finally, the conflict analysis module prints the detailed analysis, which lists information on all equipment conflicts for all given start times of the action analyzed. This analytical process is repeated for each action not scheduled. When no actions remain to be analyzed, the conflict analysis module returns control to the control module.

system subroutines

The preceding program modules use a group of standard programs that are known collectively as the CANS system subroutines, and which are attached to the user program by CALL statements. System subroutines perform such functions as: add, delete, or modify entries in direct-access storage; make conversions; maintain indexes; read from direct-access storage into main storage and the reverse; and obtain and enter the correct time into calling programs.

Concluding remarks

The Computer Assisted Network Scheduling System (CANS) performs the major functions of producing conflict-free schedules, and maintaining files of Manned Space Flight Network resources, schedules, and requested actions. The system also performs the functions of simulating requested actions and displaying their effects. All CANS functions may be performed interactively or in the batch mode of computing system operation. Although CANS was originally designed and used on System/360 Model 95 with OS/360 in the MVT environment, it can be used with the System/360 Model 40 (or larger) with the storage specified in the paper and can be modified to operate in the DOS, MFT, or other configurations.

The system described is presently in operation at the NASA Goddard Space Flight Center. Although it is highly oriented toward scheduling the Manned Space Flight Network, the CANS design concepts are potentially transferable to many complex scheduling problems. Most scheduling applications have in common with CANS such capabilities as schedule preparation, file maintenance, and conflict analysis. Although CANS is optimized around the concepts of mission safety, priorities, and user specification of the time of action, the system could be optimized for a different set of constraints just as well. Under the constraints given in this paper, the lowest cost or most efficient schedules cannot be produced; under other conditions, these factors could be improved. Since CANS performs the functions of planning and simulation, it can be adapted to model, simulate, modify, and determine system loading for an unknown or hypothetical system.

CITED REFERENCES

1. M. N. Perry and W. R. Plugge, "American Airlines SABRE electronic reservation system," *AFIPS Conference Proceedings, Western Joint Computer Conference* **19,** 593–601 (May 1961).
2. R. W. Conway, W. L. Maxwell, and L. W. Miller, *Theory of Scheduling,* Addison Wessley, New York, New York (1967).
3. *Proceedings of the First International Congress on Network Analysis (INTERNET)* held in Vienna, Austria, June 1967, Dunod Publisher, Paris, France (1969).
4. J. Horowitz, *Critical Path Scheduling—Management control through CPM and PERT,* The Ronald Press Company, New York, New York (1967).
5. G. H. Mealy, "The functional structure of OS/360, Part I Introductory survey," and B. I. Witt, "The functional structure of OS/360, Part II Job and task management," *IBM Systems Journal* **5,** 1, 3–29 (1966).