N91-22772

Sharing Intelligence: Decision-Making Interactions Between Users and Software in MAESTRO[1]

Amy L. Geoffroy, John R. Gohring & Daniel L. Britt
Martin Marietta Information Systems Group
M.S. XL4370
P.O. Box 1260
Denver, CO 80201-1260

ABSTRACT

By combining the best of automated and human decision-making in scheduling many advantages can accrue. The joint performance of the user and system is potentially much better than either alone. Features of the MAESTRO scheduling system serve to illustrate concepts of user/software cooperation. MAESTRO may be operated at a user-determinable and dynamic level of autonomy. Because the system allows so much flexibility in the allocation of decision-making responsibilities, and provides users with a wealth of information and other support for their own decision-making, better overall schedules may result.

INTRODUCTION

In complex space applications Artificial Intelligence software is often embedded into a large hardware/software/human system to support a target function (Perkins & Truszkowski, 1990), such as fuel loading (Delaune et al, 1985), manifesting (Hankins et al, 1985), ground resource allocation (Durham et al 1990), or scheduling of on-board activities (Ruitberg & Ondrus, 1990). There is a functional division of labor between the various parts of the large system, and AI software can take on any of a variety of roles.

The distribution of *decision-making* responsibilities between AI (or other software) systems and people is of particular interest. A system may be used primarily to automate tasks which are repetitious or boring, e.g. telemetry monitoring (Basile, 1988). Such a system generally provides summarized data, alerts and alarms to a human operator. In this case, a minimum amount of decision-making is performed by the system - it filters and to a limited extent interprets the information for the operator. However, the operator bears the brunt of responsibility for making decisions in this type of system. Sophisticated user interfaces can enhance the overall decision-making performance of the large (hardware/software/human) system by supporting the user with well analyzed, clearly presented and easily accessible information that supports the operator's decision-making process.

An increased role in the decision-making process is realized with advisory AI systems. These systems perform various roles in the decision-making process - some provide occasional critiquing and advice on a user's performance, as with design assistants (Lemke & Fischer, 1990) . Other systems provide constant input and oversight, as in some intelligent tutoring systems (Anderson et al, 1990). Some systems perform most of the requisite reasoning for a task, and recommend courses of action. The user may query the system for lines of reasoning or justifications, and has the final authority to follow or reject the system's advice (Shortliffe, 1984). In these advisory systems, the systems perform much of the reasoning upon which decision-making is based, but the ultimate decision and responsibility rests with the operator.

Few systems fully automate all decision-making. Those that do usually have timing requirements that preclude human interventions, such as process control (e.g. DISPATCHER, described in Kempf et al, 1991), or have strict requirements for autonomous operations, such as some concepts

[1] MAESTRO is a proprietary product of Martin Marietta Corporation.

for the Mars Rover. This is partly due to the brittleness of AI techniques in boundary cases, and partly due to reluctance on the part of potential users to accept totally autonomous decision-making by software systems.

In recent work with our scheduling system, MAESTRO, we have been developing interface support which allows the user to flexibly and interactively vary the division of decision-making responsibility between the user and the system, from use of the system as a decision support tool to operation of the system in a fully autonomous mode. (Fox [1989] also discusses mixed-initiative scheduling, but from a somewhat different perspective). There are two main considerations motivating this approach. The first concerns achieving the best overall system performance - i.e. generating the best schedules possible given both the human and computing resources that can be brought to bear on the problem. The second has to do with user acceptance, i.e. how willingly schedulers will accept the schedules produced by the system.

For many space applications, (e.g. onboard experiment scheduling, ground processing for the shuttle) the quality of the scheduling process has important implications for operational cost, productivity, and crisis avoidance. Scheduling in complex, resource-constrained domains is an extremely hard problem. Currently, realistic scheduling problems are not solved well by either entirely manual or entirely automatic systems. At least in the near term, it does not seem feasible to automate certain aspects of human scheduling performance. This appears to be due to the multiplicity of potentially conflicting goals for a given schedule, and the many and diverse strategies that people can flexibly bring to bear on a particular scheduling problem. On the other hand, efficient scheduling in complex domains is beyond the capabilities of people unaided by computer systems. Even with aiding systems that perform primarily bookkeeping and/or follow standard operations research techniques, scheduling performance can be poor for complex applications.

People and software systems, even artificial intelligence systems, display strikingly different strengths and weaknesses. In many cases the best performance for an AI system is found by *not* performing tasks the same way a person would (e.g. chess programs, Hsu et al, 1990), but by taking advantage of the strengths inherent in computers, such as enormous amounts of memory. People, on the other hand, perform far better than software systems in a variety of ways. Most importantly for scheduling problems, they are far more flexible in their reasoning strategies. By combining the best of automated and human decision-making in scheduling many advantages can accrue. The joint performance of the user and system is potentially much better than either alone. Users can act as high level managers, selecting, enforcing, and switching between global strategies, or tweaking system performance according to their own knowledge and experience. The system, on the other hand, can contribute memory, perform limited brute force searches, do local optimizations, and carry out prespecified scheduling strategies. It is not always the case that joint decision-making will be superior to either fully manual or fully automatic scheduling - e.g. instances where scheduling strategies are easily specified and entirely inflexible can easily be handled by a fully automatic system. However, even in cases where fully automatic and joint decision-making do not produce significantly different schedules, user acceptance for schedules that have been jointly generated will be much higher. There is also potential for migration of user strategies into the software and for increased user directability of the software as it evolves.

Effectively, then, we are trying to create an overall system that gives us the best of both worlds - human intelligence and automated intelligence merged. This paper is a report on what this could mean in practice, and how we are developing a user interface to support this.

The remainder of the paper is divided into four main sections. The first section provides a description of the resource-constrained scheduling problem. The next section describes the MAESTRO scheduling system. In the third section, flexible interactions with MAESTRO supported by the user interface serve to illustrate concepts of user/software cooperation. Examples are given which illustrate the points made above. Finally, we derive some general conclusions from this work.

THE SCHEDULING PROBLEM

Resource-constrained scheduling is the fixing of activities on a timeline such that those activities may be performed at the times specified by the schedule. This entails the coordination of requisite resources, the availabilities of required ambient conditions, and the interleaving of activities which compete for resources.

Take as an example the astronomical mission last December (1990) onboard the shuttle Columbia, STS-35 (ASTRO-1 and BBRXT). For the astronomical mission itself there were four main instruments, three of which shared a pointing system. The instruments were controlled by both ground and onboard crew, and the ground control was often distributed between NASA centers (MSFC, GSFC and JSC). For any given experiment, at least one instrument was in use, with associated requirements for power, thermal, computer and communications. Additional resource requirements for the same experiment included onboard and ground crew support, sometimes from multiple centers, target (star) visibility, and certain other conditions (e.g. pointing system stability, absence of the South American Anomaly). All of these requirements had to be met *simultaneously* in order to achieve a successful observation. Each observation could be in contention with others for resources such as instruments, crew, pointing orientation, or computational time. So a schedule had to ensure both that 1) the resources for one experiment/observation were fully coordinated, and 2) no resource or other conflicts existed between activities as scheduled. In addition, some observations required coordination between instruments, further complicating the scheduling process. Moreover, the astronomical mission was only part of the overall operation of the shuttle. The experimental program competed with other shuttle activities for limited resources such as crew, computer and communications time.

On a mission such as this, every potential observation minute is precious, so schedule validity and efficiency are paramount requirements. Efficiency is produced by packing the schedule as tightly as possible, which generally means performing as many observations/operations concurrently as possible. Unfortunately, scheduling problems such as this are computationally intractable using pure optimization techniques. For a variety of reasons, artificial intelligence techniques provide the most promising approach to automating the scheduling process (Geoffroy et al, 1990).

Over the last several years we have been exploring the use of AI techniques for resource-constrained scheduling problems. We have developed a sophisticated, intelligent scheduling approach embodied in the MAESTRO system, which has been demonstrated in several prototype applications. The challenge of sharing intelligence between the user and the system has been a primary focus of our recent work.

It is important to emphasize that it is the richness of the representation and reasoning strategies already embedded in MAESTRO that make it possible to support profitable interactions with users. Three main elements are required to support joint operator-system decision-making. First, the system must possess a rich, detailed representation of the domain. This indepth representation is required to supply both the user and the system with the basic information that enables intelligent decision-making. Second, the system must possess sophisticated reasoning strategies. For software to be a useful partner in intelligent decision-making (in contrast to being useful as a decision support tool) it must bring its own useful skills to the table. Third, the system must have mechanisms for sharing its information and reasoning processes with the user. Without these three elements, the system will be a weak partner in the decision-making process. We have deliberately aimed for as much system intelligence and power as possible by emphasizing rich, detailed domain representations, and fairly complex reasoning strategies which exploit the richness of those representations. The approach we have followed - information rich representations with sophisticated reasoning strategies - makes the system suitable for joint intelligent operations with users.

In the next section we describe MAESTRO as an approach to scheduling, and in the following section we turn to how decision-making processes of MAESTRO and the user can be flexibly merged at the user's direction.

THE MAESTRO SCHEDULING SYSTEM

MAESTRO is a prototype software system, using standard and artificial intelligence techniques. It has been used to explore some of the basics of scheduling, and to examine ways of reducing the computational complexity of the scheduling task while trying to achieve high-quality schedules. It has been under development since 1986. A detailed description of the MAESTRO system is required here to provide a basis for understanding the kinds of interactions a user may have with the system.

The MAESTRO system takes as inputs activity models, a scheduling period, profiles of available resources, conditions profiles, and a list of activities requested for scheduling. The system outputs a timeline of scheduled activities, updated resource availability profiles, simple evaluations of the completed schedule, and a listing of activities' success level by priority.

Activities are modeled as ordered series of subtasks, each of which must be performed in the order specified to accomplish the activity. Temporal relationships which are required between a subtask and any other subtask (within the same activity or in a different activity), event, or absolute time may be specified within the activity model.

Experiment: CONTINUOUS FLOW ELECTROPHORISIS
Subtasks:
- Run preparation through sterilization
- Flush
- Power up and get ready
- Run
- Flush buffer solution
- Power down and remove sample
- Extract
- Centrifuge
- Culture preparation
- Incubation
- Separation
- Stain and analyze
- Package and freeze

Priority: 1
Number of runs requested: 4
Temporal relations: Packaging subtask must be completed at least 3 hours before loading of returning materials on the shuttle.
Soft constraints: Maximize duration of Run subtask, minimize delay between Extract and Centrifuge subtasks, begin experiment as early as possible.

A)

Subtask: FLUSH BUFFER SOLUTION
Parent task: Continuous Flow Electrophorisis
Min duration: 35 min
Max duration: 35 min
Min delay: 0 min
Max delay: 0 min

Equipment used:	
Fluid handling tools	1
General tools	1
EqA09 chamber	1
Resources required:	
Crew	1
Power	100 w
Heat rejection	100 w
Consumables:	
H20	10 l

B)

Figure 1. Panel A shows an example of an activity description, B shows a sample sub-task description

Additionally, each activity has some baseline priority (e.g., life support activities are more important than experimental activities), and may be requested for some repeated number of performances. Any activity that can be modeled as a series of subtasks, each with resource and environmental requirements, can be represented by the system.

Each of the activity's subtasks is in turn represented as an entity with a specified (but perhaps variable) duration, a specified delay (again possibly variable) between it and the preceding subtask, and a specification of the resource and environmental conditions requirements that must be met to perform that subtask. Figure 1A shows a partial description of the information resident in an activity description, and Figure 1B illustrates a partial representation of a subtask for the activity shown in Figure 1A.

The representation of activities allows for the specification of both hard constraints - those which must be absolutely satisfied, and soft constraints - preferences which can be ignored if necessary. So for instance, a particular experiment may be required to occur during daylight hours only (a hard constraint), with a preference for around 10 a.m. (a soft constraint).

To build a schedule, the system:

(1) creates sets of related activities based on any specified temporal relations between activities;

(2) selects an activity set to place on the schedule;

(3) for each of the activities in the set:

(3a) places the activity in a valid slot (i.e., a region on the timeline where all timing, resource, and condition requirements are satisfiable); and

(3b) updates the resource profiles according to the usage predicted for that activity;

and then repeats this process (steps 2 & 3) until no more activities may be validly scheduled.

One of the most important features of the MAESTRO system is the method by which we determine whether and where on a schedule an activity may be placed. On every scheduling cycle the opportunity for each activity is determined. The opportunity calculation specifies all and only those points on the timeline where each subtask in every activity may start and end, considering all resource requirements for each of the subtasks, and the required temporal relations between an activity's subtasks. This means that all possible solutions are represented, and no point is specified that could not be used in a potential solution. We can use this measure for two purposes. The first is to derive from this a secondary measure of how hard it is to schedule an item, given the current schedule and remaining resources. Difficult to place, highly constrained items will have low opportunity, and this information can be used to make intelligent decisions about what to select to schedule on a given scheduling cycle. Second, given that we know all possible start and end times for each subtask, we can use this to make decisions about how to place items on the timeline. It is possible, for instance, to find the earliest or latest possible start for a subtask, or the longest possible runtime for a data collection subtask. This information is critical in much of the intelligent decision-making that the system currently performs, and is the basis on which many more intelligent heuristics may be built.

Note that the ability to identify all possible placements using this method assumes that each activity may be placed independently of other, as yet unscheduled, activities. This becomes an important point in scheduling activities that bear some temporal relation to one another (e.g., cyclically recurring activities, or multiexperiment campaign science projects). Additional scheduling strategies have been implemented to accommodate such relationships so that items constrained in these ways are likely to be successfully scheduled where possible. Britt et al (1990) gives a more complete description of temporal relations management within MAESTRO.

Contingency rescheduling operations are supported in MAESTRO. The system uses information about the current schedule, the projected violation (e.g., a resource deficit), the time of the violation relative to current real time, and characteristics of the scheduled activities to determine what activities need to be altered or descheduled to accommodate the contingency. Heuristics relating to various rescheduling goals help determine which items will be altered. Knowledge about the activities' current states, potential repairs (e.g., interruptability), and performance requirements are used to make deschedule, repair, and/or reschedule decisions. Activities that are in progress during a real-time contingency receive special treatment to synchronize schedule fixes into real-time operations. The system also synchronizes schedule changes with subsystems when the contingencies are real-time or near real-time.

SUPPORTING THE USER IN JOINT DECISION-MAKING INTERACTIONS

MAESTRO was originally designed to work autonomously, and the description of MAESTRO given above presents MAESTRO as it operates in an autonomous decision-making mode. In this mode, there is little user control beyond some parameter specifications, such as weightings on decision criteria. As the project matured, it became apparent that:

a) MAESTRO occasionally did stupid things, and a little user intervention could go a long way, and

b) experienced schedulers could get a lot of mileage using partial results of MAESTRO's reasoning processes (e.g. identification of opportunity windows), while imposing their own strategies during the scheduling process, and selectively intervening in the scheduling process.

In short, the intelligent behavior of joint user/system decision-making could be a big improvement over their behavior in independent operation. The way to achieve this was to provide the user with the means of selectively and flexibly driving the division of responsibility. The user could then allow the system to perform autonomously when that seemed best, and take over decision-making responsibilities when skilled human judgement would produce superior results.

This required major changes in the user interface and some of the control flow in the system. The user interface must provide support for different allocations of user/system responsibility[2]. This includes support appropriate to each level of interaction, plus mechanisms for transferring decision-making responsibility between the user and the system. Because the system is rich in available information, and in the number of control options available to the user, accessibility of information and control through the user interface is a key issue. Hence, the discussion begins with a description of information accessibility. This is followed by a description of the kinds of support the user receives for different divisions of user and system control. Subsequently, we describe how levels of control can be intermixed and varied during a scheduling session. Finally, an example is given of how this works for an instance of contingency handling.

Information Accessibility

In supporting effective joint decision-making, the access to information is as important as the access to control commands themselves. The structure provided to the user in navigating through the system to access information and control is key to successful overall performance. The MAESTRO user interface provides convenient access to enormous amounts of information and system control. The interface is organized to provide users with guidance and structure without unnecessarily restricting their actions.

---

[2] Note that there is an underlying assumption that the operator of the system is also a sophisticated scheduler. To participate in profitable joint problem-solving, both the user and the system must bring useful knowledge and skills to the table. This is not to say that the user should be obliged to know much about computers, beyond knowing how to run an applications program.

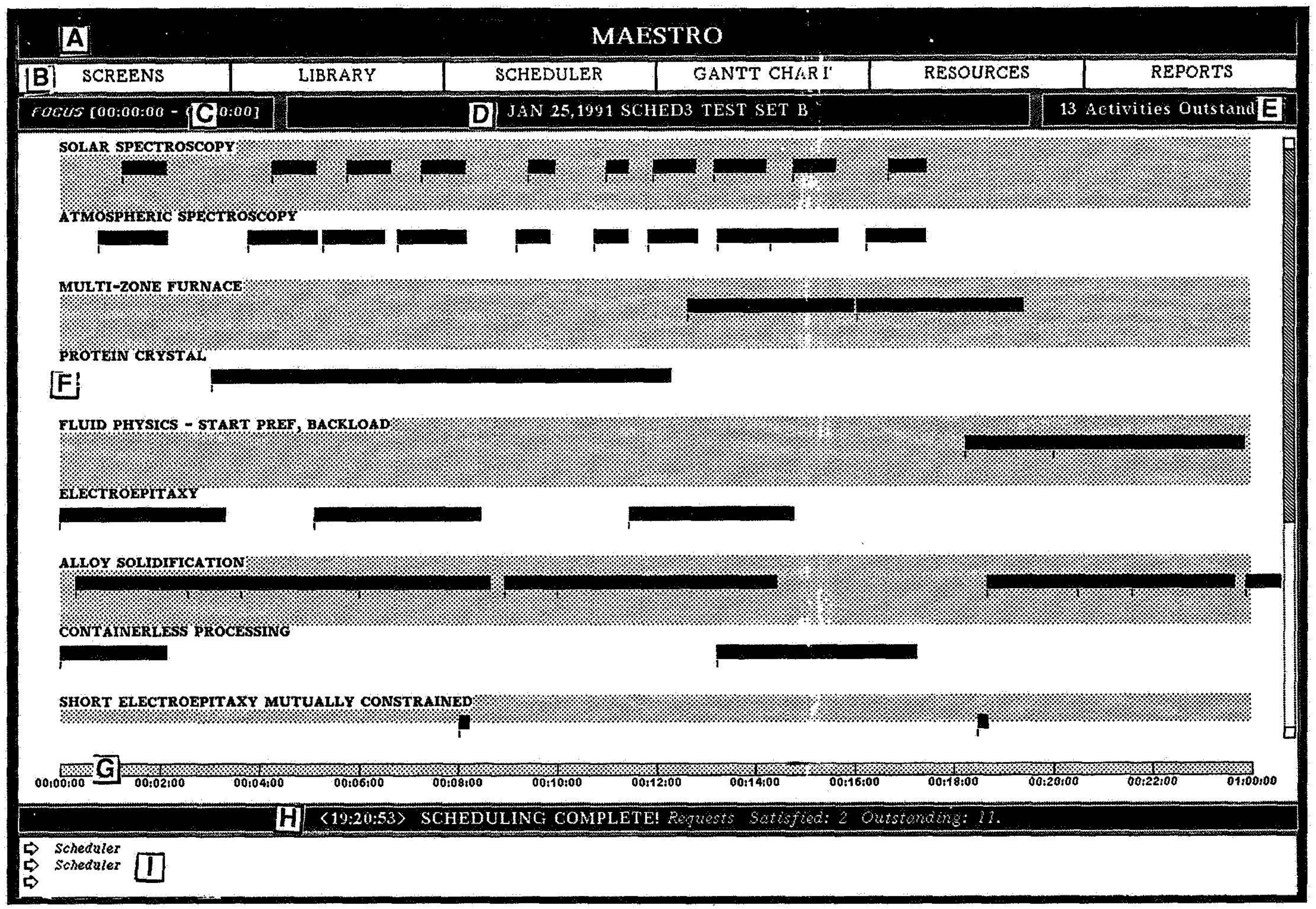


Figure 2. MAESTRO screen display from a typical scheduling run. Lettered regions of the schedule screen schow:

A) Scheduler Name
B) Menu Bar
C) Current Scheduler Focus

D) Schedule Name
E) Scheduler Status
F) Gantt Chart

G) Timeline
H) History of Scheduler Operations
I) Command Type-In

ORIGINAL PAGE IS
OF POOR QUALITY

There are more than 70 commands which a user can access through the user interface. These include both querying functions, such as resource displays, and controlling functions, such as activity selection. The user interface is organized to support easy discovery of, access to and use of the various options. Commands have been topically organized into hierarchical menus. The scheduler screen, for instance, has six major menu headings - "Screens", "Library", "Scheduler", "Gantt Chart", "Resources", and "Reports" (see Figure 2). Each of these provides relevant sub-menus, some of which are themselves topically organized. So, for instance, under the "Library" menu, the items are divided between "Permanent" and "Snapshot" operations. Where a given command requires the user to supply arguments, the system supplies an appropriate dialog prompting the user for the required information.

Each of the menus is context sensitive. The displays only reflect commands which are legitimate to perform given the current state of the system. For instance, you can only retrieve a saved schedule under certain conditions - there must be a schedule already saved, and you can't be in the middle of creating another schedule. The "Library" menu reflects this - it will display the "Retrieve Schedule" option only when the system is not scheduling, and there are items in the schedule library. These menus help to guide the user to appropriate actions given the current state of the schedule and scheduling process.

Many objects on the display are active. Mousing on activity names or bars in the Gantt chart can retrieve a wealth of information about each activity, and parts of the information displays are themselves active. So, for instance, the user can mouse on an activity name to get a list of subtasks, and mouse on a subtask within that list to get a fuller description of that subtask. The potential actions are clearly explained on the mouse documentation line, so that the user can always determine the effect of an action before trying it. Various displays are available which provide information about resources, activities and their sub-components, and general schedule information.

In addition to providing information about a particular schedule, the system also provides for schedule saves, both permanent and snapshot (temporary). Among other things, this allows users to develop a partial schedule, save it, and generate and compare alternative completions of the schedule.

Different Levels of Decision-making Control

The user may choose to exert no control over the system beyond selecting a scheduling period and a set of activities to be scheduled during that period[3]. The system will default to a set of standard parameters' values, and scheduling will be performed entirely autonomously using those defaults.

One level of user control involves the user in setting some or all of these parameters. These parameters include:

a) weightings for activity selection in regular scheduling operations (the relative importance of success, opportunity and priority in determining the next item to be scheduled),

b) weightings for activity selection in contingency situations (the relative importance of ten criteria for determining which item to alter or deschedule in a contingency/resource overbooking situation. In addition to success, opportunity and priority, factors important in descheduling, such as whether the activity is already initiated or easily alterable are considered) ,

---

[3] Note that the activity descriptions (which have been previously specified) contain scheduling directions about both hard and soft constraints. Hard constraints are those which absolutely must be met for the item to be successfully scheduled, and soft constraints represent preferences. MAESTRO always schedules activities fully respecting the hard constraints. In the default mode, MAESTRO uses the model-specified soft constraints to guide placement. Thus, the person who defined the activity is also exerting influence on the scheduler's performance.

c) selection strategies (default is the weighted criteria described in (a), but the user may select a different strategy, such as omitting opportunity calculation for selection),

d) placement strategies (the default is to follow the preferences specified in the activity model, but the user may choose to ignore these preferences), and

e) scheduler focus (the user may instruct the system to only pay attention to a portion of the scheduling period rather than its entire length).

Once these system parameters have been set, MAESTRO can again perform scheduling entirely autonomously. The strategies used in scheduling, while selected by the user, are all part of MAESTRO's repertoire.

Users may take further control of the scheduling process by enforcing their own control strategies in running the system. The mechanism by which the user performs this is by selecting options[4] for:

a) User activity selection - On each cycle the user determines which activity will be scheduled.

b) User activity placement - For a user-specified subset of requested activities the user places each of these activities.

c) User activity deletion - The user may delete specific instances of scheduled activities.

d) Manual altering of resource profiles. This allows the user to increase or decrease the amount of an available resource for any arbitrarily specified portion of the timeline.

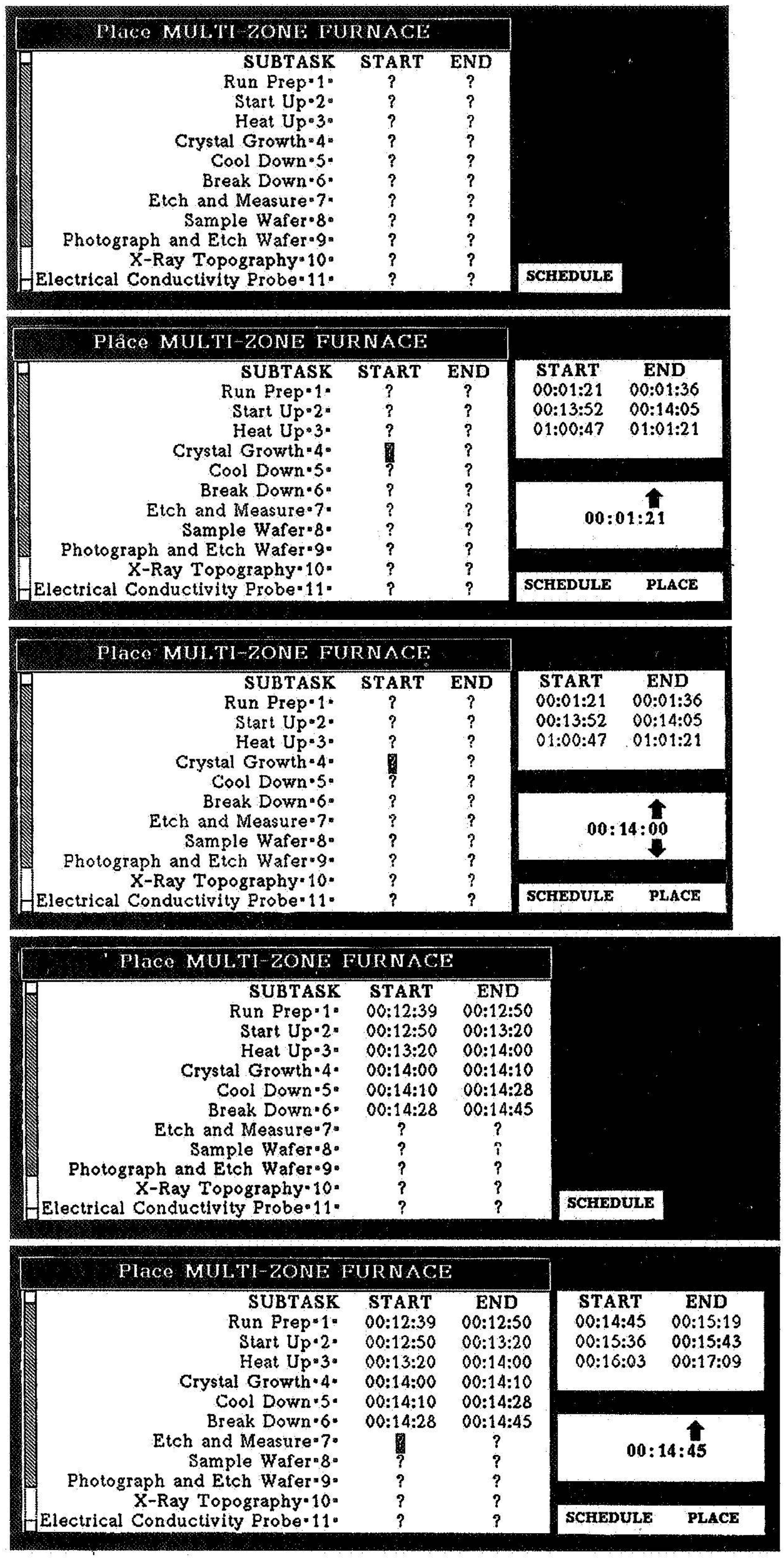
By using these options, the user may take on most of the decision-making responsibility, and MAESTRO then acts more as a decision-aiding mechanism. However, MAESTRO does supply a good deal of support for each of these user operations. MAESTRO provides this support by supplying the user with access to the information the system normally employs in its own decision-making processes. The primary functions used in this way are the system's bookkeeping, constraint propagation, and temporal relations management mechanisms.

a) User activity selection - The system provides a menu from which the user selects an activity to be placed. Only activities which still have opportunity are displayed, so that the user does not try to place an activity which can't fit on the schedule. Additionally, items which are in a related set due to their temporal relations are displayed as a group, so a user will know that all members of that group will be selected for placement on that scheduling cycle. The menu also indicates which activity MAESTRO would select, and the user can accept that option if desired.

b) User activity placement - Recall that activities consist of subtasks with variable durations, and variable delays between subtasks. This allows for flexibility in the scheduling process - more things can get scheduled, and resources can be used more efficiently than if durations and delays are fixed. However, activity placement is much more complex, for the user as well as for the system. To support user placement, MAESTRO provides information about possible placement options based on the previously described opportunity calculation. Whenever an activity designated for user placement is selected for scheduling, an interactor[5] for that activity appears (see Figure 3A). The interactor displays each subtask name and a mouse-selectable icon for each subtask start and end. The user selects the subtask start or end that he wishes to specify, and all possible times for that particular point are provided in the upper right window of the interactor (Figure 3B). The user selects from among

[4] These user options may be used in any combination, e.g. user placement may be used with or without user selection.

[5] An interactor is a window partitioned into subwindows, which provides both information and guides the user through a series of interactions with the system.



A) The interactor presents the user with a list of the activity's subtasks, and icons (the "?"s) to select from to choose the subtask start or end the user wishes to specify first.

B) The user decides to begin by specifying the start time for the Crystal Growth subtask. The interactor displays all possibilities (i.e. this subtask can be started anytime between 1:21 and 1:36, 13:52 and 14:05 or 1:00:47 and 1:01:21) in the upper right subwindow.

C) The user chooses 14:00 as the start time for the Crystal Growth subtask, using the dial in the middle right subwindow. When the user clicks on "PLACE" (lower right) MAESTRO accepts this as the chosen start time, and propagates the constraints this places on the rest of the subtasks' start and end times.

D) The interactor displays all the start and end times that are fully specified. In this case, by specifying the start time for the Crystal Growth subtask, the user has also fully constrained all start and end times for all the subtasks through the end of Break Down. The user may now choose from the remaining subtasks' starts and ends.

E) The user decides to place the start time for Etch and Measure next. The user clicks on the icon, and the system displays the remaining possible times that this subtask may start. The process of user selection and constraint propagation continues until all times are fully specified, or the user directs MAESTRO to complete the process by clicking on "SCHEDULE".

Figure 3. The user places a materials processing experiment, Multi-Zone Furnace, on the schedule. Panels A-E show the sequence of steps the user performs to accomplish this, using the User Placement interactor.

these the desired time for that subtask start or end (Figure 3C), and the system propagates the constraints this imposes on other subtasks' possible start or end times[6]. The interactor is updated, filling in start and end times which have been totally constrained by the user's choice (Figure 3D), and letting the user select the next underconstrained start or end time for specification (Figure 3E). This process continues until the activity placement is fully specified. At this point MAESTRO places the activity and performs the normal bookkeeping associated with activity placement.

User activity placement support is a powerful facilitator for the user. First, it guides the user to valid placement options, so there is not a lot of time wasted in shot-in-the-dark or partially informed guesses about where an activity might fit. The more resource coordination required for an activity, the more critical this facility becomes. Second, the information provided allows users to impose their own placement strategies easily. It is easy to find the earliest possible start time, or to start something as soon as possible after noon, to maximize a data collection subtask duration, or to minimize the delay between calibration and data collection subtasks. MAESTRO is able to provide this support because the information provided to the user is the same as that which the system uses in its own intelligent scheduling. By providing the user with intermediate products of MAESTRO's reasoning processes (in this case opportunity calculation and constraint propagation mechanism outputs) the user is relieved of the burden of operations which are hard or impossible for a person, and left to perform tasks which are more suited to human intelligence.

c) Manual activity deletion - The user may remove any activity from the timeline with a mouse command. A verification menu pops up, to ensure that the user intended to delete the activity (*all destructive operations* require confirmation as a safety feature). Upon user verification, the system removes the activity from the schedule, updates the resource profiles and does other bookkeeping as appropriate. Further, it checks to see if removal has caused any violations of other scheduled activities' temporal constraints. If so, MAESTRO will remove the violated activities and notify the user of any activity's removal. All activities' opportunities are recalculated to reflect the new state of the schedule.

d) Manual alteration of resource profiles - The system provides a dialog through which the user can alter the amount of a resource available. The system automatically updates the resource as appropriate[7]. The system checks to ensure that no resource overbooking has occurred as a result of this change, and notifies the user if there is a contingency. Finally, it updates opportunity for all activities based on the new resource availability profiles. In this case, the functions performed are primarily bookkeeping, but the end result is that the user is informed of any relevant effects of his actions.

Intermixing Control

The division of decision-making responsibility does not have to be fixed at a particular level. Throughout the creation of a given schedule, the locus of responsibility can be flexibly varied. This is achieved by changing parameters, selecting different control options, and selectively accepting defaults during the session. Many of the parameters and control options may be changed while the scheduler is executing. The

---

6 For example, before selection of the start time for crystal growth at 00:14:00, the end time for crystal growth could potentially have been at 00:01:31, 01:01: 31, or at any of a large number of now invalid times. MAESTRO takes care of ensuring that only valid placement options remain for user selection.

7 The actual update method depends on the resource type (e.g. rate-controlled or consumable), projected resupplies, etc. However, since the system already knows which resources are of what type, and appropriate update methods for each type, the system relieves the user of the burden of worrying about these sorts of things.

user may also halt the scheduler at any point in execution[8] and change any parameter or control option, then restart the scheduling process where it left off. Additional flexibility in shifting control is available through the user selection and user placement options. The user selection menu specifies which activity MAESTRO would have chosen next, and the user may accept that default without any deliberation. At any point in the user placement process, the user can allow MAESTRO to complete the placement process, using its normal heuristic strategies. In this case, the user only has to do the work of interest (e.g. specify a particular starting point, or maximizing a subtask duration) and may delegate the rest of the decision-making work to the system.

Example - Contingency Handling

An example of a contingency handling interaction serves to illustrate the concept of flexibly directed mixed control. Let us assume the following scenario. A user, Jane, has previously generated a complete 24 hour schedule for a space laboratory mission that is due to be flown tomorrow. There has been some anomaly in the power system that will cause a 25% across-the-board reduction in the power available to the laboratory from 8:00 to 11:00 a.m. during that scheduling period. The scheduling system is notified of this contingency. The system updates resource availabilities, and checks to see if any resource violations have been induced. If there is a problem, an alert flashes notifying Jane that there is a contingency.

At this point, Jane can follow several courses of action. She may: a) choose to ignore the contingency for the time being, and perform other, more pressing operations with the scheduler; b) ask the system to handle the contingency automatically, in which case it will follow the contingency procedures previously described; c) check to see whether the parameters for the contingency selection mechanism are what she desires, optionally reset them, and then have the system handle the contingency automatically; or d) choose to handle the contingency herself.

To handle the contingency herself, she will typically need more information, so she would probably check the times and levels of the power resource overbooking for the schedule period through the "Resource" menu. After examining the overbooking information, she can continue her investigation into the problem, ignore it, or let the system handle the contingency (optionally checking and altering the contingency selection parameters). If she chooses to continue her control, she can check which activities use power during the violation periods, then ignore or automatically handle the contingency, or continue with control. At this point she may have in mind a particular activity or set of activities to delete. She will delete these by simple mouse operations, and MAESTRO will update the schedule to reflect these deletions. It will check to see if there is still a violation of the power constraints, and if so that will be available in a display. Jane will check the display to see if a violation remains, and either ask the system to complete the contingency handling process, continue with her own contingency handling process, or ignore the remaining problems. At various points in her process of deleting and checking the results, she may save her partial results in the snapshot or permanent schedule libraries. This will enable her to perform comparisons of alternative courses of action.

When contingency handling is completed, some of the deleted activities, or some other activities that may not have been able to get on the schedule previously, may now be schedulable. Because the system has been doing all the usual bookkeeping/updating this can be easily checked. Jane can then, if she chooses, initiate a new cycle of mixed control in adding things to the schedule.

There are two main points in this example. First, the user can hand off control to the system at many points during the contingency

[8] The scheduler will actually take the halt instruction and proceed to the next logical stopping place, usually by completing an activity placement in progress, updating the resources and performing other bookkeeping chores, and then halting the scheduling process.

handling process. Second, in her own reasoning process, the user accesses much of the information that MAESTRO typically generates for its reasoning process, and relies on the bookkeeping and constraint propagation mechanisms of the system to complement her own decision-making.

When would Jane take control? When performance would be significantly improved by use of information unavailable to the system or by following reasoning strategies the system does not possess, and only when time is not a critical issue. In real operations there will probably always be information and reasoning strategies that are unavailable to an automated system, and it is up to the user to try to merge these with the support that the automated system can provide. It is up to the system to provide mechanisms that ensure the user can flexibly access as many of its useful processes as possible.

CONCLUSIONS

Users can access an abundance of raw data and intermediate products of MAESTRO's reasoning processes. They may selectively take over some of the decision-making functions, leaving others to the system. Because the system allows so much flexibility in the allocation of control, and provides users with a wealth of information and other support for their own decision-making, better overall schedules may result.

We have made considerable progress in organizing the user interface to support profitable interactions, but far more could be done. There is still much internal information that either is unavailable or hard to access through the interface. For instance, during contingencies users can find out some of the values for the contingency selection parameters for each activity through the interface (e.g. opportunity, interruptability), but not others (e.g. resource fit). The organization of the parameter values that are available is poor for contingency operations - the information is scattered throughout displays that are typically used for other purposes. Some mixed control that could be supported currently is not. Using contingency handling as an example again, the user cannot currently allow MAESTRO to make only a single deletion and halt, or make a recommendation which the user could accept or reject. We are currently planning enhancements that make more of the system's internal information accessible and to make switching control more flexible, leading to more powerful joint decision-making.

REFERENCES

Anderson, J.R., Boyle, C.F., Corbett, A.T. & Lewis, M.W. (1990) Cognitive modeling and intelligent tutoring. *Artificial Intelligence* 42(1), 7-49.

Basile, L. (1988) Spacelab data processing facility quality assurance/data accounting expert systems: transition from prototype to operational systems. *Proceedings of the 1988 Goddard Conference on Space Applications of Artificial Intelligence.* NASA Goddard Space Flight Center, Greenbelt, MD, 329-341.

Britt, D.L. Geoffroy, A.L. & Gohring, J.R. (1990) Managing temporal relations. *Proceedings of the 1990 Goddard Conference on Space Applications of Artificial Intelligence.* NASA Goddard Space Flight Center, Greenbelt, MD, 123-135.

Delaune, C.I. Scarl, E.A. & Jamieson, J.R. (1985) A monitor and diagnosis program for the shuttle liquid oxygen loading operation. *Proceedings of the 1st annual Workshop on Robotics and Expert Systems*, Houston TX.

Durham, R., Reilly, N.B. & Springer, J.B. (1990) Resource Allocation Planning Helper (RALPH): Lessons learned. *Proceedings of the 1990 Goddard Conference on Space Applications of Artificial Intelligence.* NASA Goddard Space Flight Center, Greenbelt, MD, 17-28.

Fox, B.R. (1989) Mixed initiative scheduling. *AAAI - Stanford Spring Symposium on AI in Scheduling.* Stanford, CA.

Geoffroy, A.L., Britt, D.L., & Gohring, J.R. (1990) The role of artificial intelligence techniques in scheduling systems. *Telematics and Infomatics.* 17 (3/4), 231-242.

Hankins, G.B., Jordan J.W., Katz, J.L., Mulvehill, A.M., Domoullin, J.M., and Ragusa, J.M. (1985) Empress Expert Mission Planning and REplanning Scheduling System. Mitre Corp. Report M85-33, Bedford, MA.

Hsu, F., Anantharaman, T., Campbell, P. & Nowatzyk, A. (1990) A grandmaster chess machine. *Scientific American*, 263(4), 44-50.

Kempf, K., Russelll, B. Sidhu, S. & Barrett, S. (1991) AI-based schedulers in manufacturing practice: report of a panel discussion. *AI Magazine* 11 (5), 46-55.

Lemke, A.C. & Fischer, G.F. (1990) A cooperative problem solving system for user interface design. *Proceedings Eighth National Conference on Artificial Intelligence (AAAI-90)* Boston, MA, 479-484.

Perkins, D. & Truszkowski, W. (1990) Launching AI in NASA ground systems. *AIAA/NASA Second International Symposium on Space Information Systems.* Pasadena, CA. AIAA-90-5055.

Ruitberg, E. & Ondrus, P. (1990) Lessons learned from the Hubble Space Telescope planning and scheduling system implementation and operations. *AIAA/NASA Second International Symposium on Space Information Systems.* Pasadena, CA. AIAA-90-5039.

Shortliffe, E.H. (1984) Details of the consultation system. In B.G. Buchanan & E.H. Shortliffe (Eds.) *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Reading, MA: Addison-Wesley.