# User-Centered Scheduling Support in the Military Airspace Management System Prototype

P. O. Perry
The MITRE Corporation,
Burlington Rd,
Bedford, MA 01730
pop@mitre.org

## Abstract

The Military Airspace Management System (MAMS) is a multi-user distributed scheduling prototype designed to support the scheduling of Special Use Airspace in the CONUS region. The prototype has emphasized the user interface design of the scheduling system as the primary means of producing de-conflicted schedules. This paper reports on work in progress and provides a technical description of the user interface support for the scheduling process.

# 1 Introduction

## 1.1 Problem Description

Nearly 25 percent of continental United States (CONUS) airspace is designated as Special Use Airspace (SUA) for use by the Department of Defense (DOD) for military operational readiness training, research and development, and test and evaluation. Demand for this airspace continues to increase from both the military and the civil sectors resulting in the need for better management.

There are over 200 military airspace scheduling offices of the different services in the United States. In the southwest, where the MAMS prototype is being field tested, there are at least 20 sites where airspace scheduling is performed. The military services differ in the schedule information they report on airspaces, and some military areas report only scheduled-use, not actual-use data. The services therefore have no uniform source of data to determine their actual use, or to compare actual to scheduled use.

The DOD plans to develop the Military Airspace Management System (MAMS) as a solution, automating scheduling and reporting of SUA use and providing near real-time joint use of airspace. The MAMS prototype is being developed to define the requirements for a tool that supports efficient scheduling and utilization data collection and reporting.

## 1.2 History

Schedulers of SUAs currently have limited automated support for scheduling, or simply form a daily flight schedule manually. Airspace users phone or fax their requests for SUA access to the offices with local scheduling responsibility. The initial contact is usually followed by a series of phone calls. Clarifications are made, and eventually a preferred mission time is established. Then, if the local priority rules do not interfere, the scheduler of the airspace will allow the requested mission to take place. In some cases the scheduler can override the local priority rules. For example, when a fleet is conducting maneuvers offshore it expects to receive the highest priority, but may be overridden if a Top Gun class at the Naval Air Station Miramar needs to fly. There are governing rules for airspaces established by the FAA and by letter of agreement. The DOD rules for assigning priorities in an airspace may change from service to service, and sometimes from airspace to airspace. The scheduler currently can resolve conflicts by generating alternatives, assigning priorities, or trying to negotiate a mutually acceptable solution.

## 1.3 Prototype Approach

The MAMS prototype is planned as a widely distributed network of scheduling sites sharing a database of airspace resources. The sites will be part of a national military airspace management system that preserves their local control of resources and provides a hierarchical structure for reporting schedule data. The network will allow DOD airspace managers to quickly request and schedule missions in local airspaces and efficiently request use of remote airspaces.

It was recognized early in user surveys that it would be difficult to capture the many scheduling strategies that a diverse user community had evolved over time, and to establish a consistent set of heuristics that would satisfy most users. Many organizations had developed site specific policies and procedures for scheduling and managing their airspace resources. Scheduling rules and practices are therefore very diverse, as are user interfaces, and there has been some disagreement on a common approach to resolve the differences. Incorporating the daily negotiation process in an automated scheduling system would also be complicated.

The approach taken in developing the MAMS prototype was to provide an intuitive user interface first, and later integrate automated support algorithms. This has had the advantage of providing a method to transition to a more automated scheduling system while extracting from the users their knowledge of scheduling processes.

The variety of organizations and their particular scheduling strategies has also led us to develop a scheduling aid where the user has an explicit role rather than fully automating the scheduling system. In an environment of continuous dynamic rescheduling it seemed more effective to provide the necessary tools via better user interface mechanisms, rather than to incorporate explicit knowledge of numerous considerations of the scheduling process. Since a given schedule is continuously revised due to changing mission requirements, the emphasis on the user interface underscores the role of the scheduler as a problem solver rather than a data entry clerk. The effort therefore centered on providing useful interface components that facilitate forming and maintaining a schedule regardless of local practices or procedures.

In its technical approach, the MAMS prototype addresses the following principal areas: the internal representation of the domain, development and optimization of an efficient user interface, supporting analytic routines, database and the distributed aspects of the application, and data gathering for standardized airspace utilization reporting.

# 2 Domain Representation

MAMS uses an object hierarchy to represent **Resources** and **Activities**. Both resources and activities share a common parent, **Schedule**, that enforces the identification and naming of each object in the hierarchy. Domain specific types of resources and activities are then specialized by inheritance.

## 2.1 Resources

A resource class represents airspace resources. The attributes associated with the class define the state of the resource, which consists of the activities scheduled at the resource. All scheduling functionality is embodied in the resource object. From this class we specialize two classes of airspaces: Special Use Airspaces (SUAs) and Military Training Routes (MTRs).

Most SUAs were created as military areas for training, testing, and national security. There are six different types: Prohibited Areas, Restricted Areas, Military Operations Areas (MOAs), Warning Areas, Alert Areas, and Controlled Firing Areas (CFAs). In 1989 the number of SUAs included approximately 350 MOAs, and 120 Warning Areas. The airspaces are organized hierarchically and are subdivided into further categories designated by the organization controlling a particular airspace.

SUAs can be designated either for exclusive use (only one organization may use the airspace), shared use (several military organizations may share the airspace), or joint use (which allows for simultaneous use of the airspace by military personnel and civilians). Airspaces may be also dynamically created to support special missions.

The DOD also uses point to point air routes known as Military Training Routes (MTRs). There are four types: IRs, which require an Instrument Flight Rules (IFR) flight plan, VRs, which require a Visual Flight Rules (VFR) flight plan, SRs, designated as special routes primarily for slow speed, low altitude operations, and AR routes for air refueling.

## 2.2 Activities

An activity is any operation scheduled within an airspace. It can be scheduled within any of the resources mentioned above, and may contain different domain attributes depending on the desired resource. In general an activity in MAMS is created by a requester, a person desiring the SUA. A scheduler must then examine the activity, evaluate it in the context of the schedule, and act upon it. The activity may be edited and changed through the user interface either by the requester or the scheduler responsible for the relevant resource.

## 2.3 State

State describes the most recently completed action on an activity. An activity's state is changed by the requester or scheduler through the user interface. A requester may create a new activity, delete an existing activity, or modify a current activity. A scheduler may look at an activity, approve an activity or a conflicting activity, or unschedule, deny, or modify an activity. The system also may change the state of an activity if a conflict arises.

The main states of an activity are requests, activities which have not been examined by a scheduler, and tasks, activities which have been acted upon by a scheduler. State is represented in the interface by color coding. Submitted requests are shown in blue, when scheduled they are changed to green, and if conflicting they are changed to red.

# 3 User Interface

In contrast to developing a user interface that provides the user with some insight and some control in the automated scheduling process [Cooper, 1990], the MAMS prototype has approached the problem by first addressing the user interface, then ascertaining how more automated support could be integrated behind the interface. This section highlights some of the key elements that aid the scheduler in establishing a conflict-free schedule.

## 3.1 Design Influences

The MAMS user interface design is based on previous MITRE efforts [Mulvehill, 1986]. It also draws on the Range Scheduling Aid [Smith, 1990] , a prototype designed to support scheduling of the Air Force Satellite Control Network (AFSCN) ground stations and equipment [Smith and Katz, 1990; Halbfinger and Smith, 1990].

All menus and dialogs in the prototype have been developed through extensive interaction with the user community. This feedback has forced many changes, and is the source of much of the volatility in the user interface design.

## 3.2 Visual Representation

The user interface is the scheduler's primary means of establishing a de-conflicted schedule. As in many similar systems, the interface is modeled on an interactive Gantt chart. The main window is divided into horizontal areas, or "panes", associated with the resources being scheduled. The window is divided from left to right by vertical grid lines the user can adjust to represent one hour to one day increments. Each requested activity is represented by a colored bar icon fixed in height and proportional in length to the duration of the mission. Its placement on the screen corresponds to the actual time at which the task is due to take place.

## 3.3 View Control

The user interface is designed to draw the scheduler's attention to those areas in the schedule that need repair. At

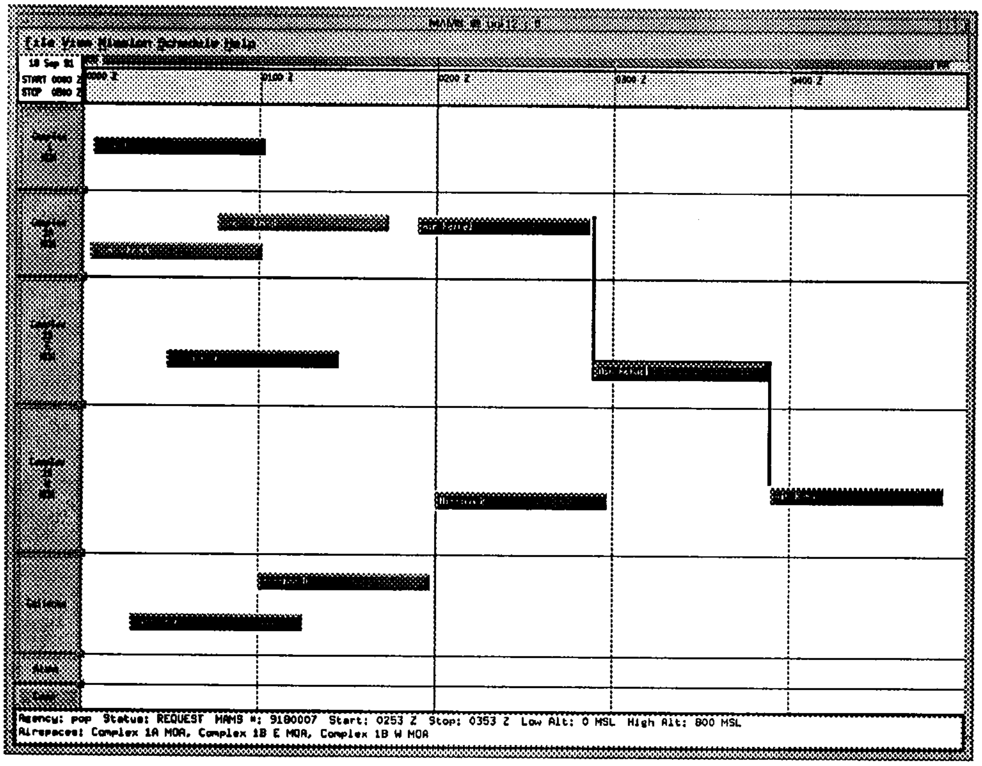


Figure 1: The MAMS Gantt Chart Interface

startup the interface begins with the current day's time frame so that the scheduler can address immediate requests first.

The interface allows the user to focus on a narrow time period, yet gain quick access to distant areas of the schedule. Using the timeline control at the top of the Gantt chart, the user can rapidly view resource data within a sliding time interval that can be varied in duration from one hour to two weeks.

### 3.4 Use of Domain Knowledge

Airspace resources are naturally organized hierarchically: at the top is the controlling organization and in lower nodes are sub-areas that are independently scheduled. A requester is able to submit a request at any level of the resource hierarchy. This action, in effect, is a shorthand for requesting all airspaces below the requested node. The user may view an airspace at any level in the hierarchy. This organization presents an overview of the schedule by displaying a summary of resource usage. The prototype also supports formation of arbitrary groupings of resources to simultaneously schedule at all airspaces of the grouped resources.

### 3.5 User Conveniences

To change the time of an activity, the user drags the icon horizontally with the mouse. To take other actions, the user selects the icon and calls up an appropriate menu. The act of selecting an icon prints information about the activity in a documentation line along the bottom of the screen.

A *find* capability helps the user locate a mission of interest. It displays a list of missions that fit certain search criteria, such as requests not yet acted on by the scheduler, or conflicting tasks in the scheduler's airspace. The user can then select a mission from the tabular list and have the screen reconfigured to the time period of the selected item, facilitating the context switch to a new mission.

The user interface also includes a number of keyboard accelerators, error correction on most input fields, an error reporting dialog, on-line help, and context sensitive help. Commands are supported both through the mouse and through key bindings.

### 3.6 Multi-User Support

Unlike previous multi-user scheduling systems [Beard *et al.*, 1990] the MAMS prototype updates user changes in near-real-time so that changes made by a requester or scheduler will be conveyed immediately to a second scheduler who is working with the same resource for an overlapping time period. However, MAMS provides the two users independent views of the schedule at the same time, with independent screen layouts and time intervals.

The interoperability of the X Window System simplified development of a multi-user scheduling system by enabling the application to open connections to multiple X servers. But supporting multiple users introduces additional complications that need to be addressed in the user interface. We added authentication and authorization checking to give a user permission appropriate for their role as a requester or scheduler [Patterson, 1991]. In addition, the application needs to support individual user preferences so that a user can configure the screen to their liking.

# 4 Analytics

It may be thought that a simple reservation system might have sufficed, since requests are made and serviced by a scheduling authority. However, the scheduler also needs support in maintaining temporal and resource constraints.

## 4.1 Temporal Relationships

MAMS maintains a point-based representation of time for a single activity, and a symbolic representation of time for links between activities. While the prototype does not incorporate a temporal constraint engine for maintaining temporal relations, it needs to support some temporal relations. We have found that users need to represent *before*, *meets* and *equal* relations [Allen, 1983] for a number of situations, but users requested that we represent only three relations between activities because other possible relations do not have a corresponding use in the application. Temporal relations are used when a complex linked mission is being planned, with multiple groups of aircraft operating in multiple airspaces according to an interdependent sequence of events. The scheduler needs to be able to define and maintain these relations. When an activity that is part of a linked mission is created, the scheduler can establish its dependencies to a concurrent or adjacent activity and maintain them graphically in the interface. If one activity is moved to a different time the related activities move with it.

## 4.2 Resource Relationships

MAMS needs to maintain resource relationships for airspaces affected by adjoining airspaces. If, for example, an MTR passes through an SUA, scheduling in the MTR will also schedule the SUA for the duration of activity in the SUA. These kinds of dependencies are automatically maintained; the user need not be aware which airspaces are related.

## 4.3 Conflict Identification

Currently when conflicts in SUAs are detected for activities which overlap in time and altitude within an exclusive use airspace, the conflicting activities are highlighted in red. In addition, if the airspace is an MTR, conflicts are detected if activities are taking place at the same time either at the crossing point of two routes or on the same route. This would occur, for example, if one airplane were to overtake another.Conflict identification is performed each time an activity changes state (is either scheduled or moved interactively).

## 4.4 Conflict Resolution

Thus far, the resolution of conflicts is left to the user. Many types of resolutions simply are not possible to automate because the system does not explicitly represent all factors in a particular scheduling choice. Rather than maintaining continuously changing knowledge in the application, the scheduler is left to resolve those aspects of the schedule that require human judgment, while the prototype maintains consistency in the schedule while managing a large set of scheduling data.

## 4.5 Conflict Description

The MAMS prototype graphically provides an explanation of why two or more activities are considered in conflict by associating them with connected lines. To aid manual resolution of conflicts, the user is then provided with a pop-up window containing a scrollable list of conflicting missions with conflicting field titles in red.

The scheduler may choose to accept a conflict, overriding the conflict detection. The color of the activity's icon is then changed from all red to green with a red border.

# 5 Management

The schedule, and therefore the airspaces, are managed primarily through collection of utilization data. After each mission, the participants are expected to report if they flew the mission as scheduled, and if not, to report any differences. This data is then entered into the system and is used to create utilization data reports. The quality of a particular mission can be recorded, and if the conditions were degraded one can enter the reason. This data can then be used to gather statistics about the number of successful missions run in a particular airspace.

# 6 Database

The prototype interfaces to a relational database for long term storage and management of schedule data. The choice of a relational database was deemed important because the system needs to support arbitrary queries on the data for report generation. Analysis of utilization data using such reports will support long term planning of airspace utilization. To maintain a record of multiple users' actions on the data, all transactions are time-stamped so that a historical trace of changes can be retrieved. This function is considered useful when a scheduler needs to review how a particular request was serviced.

# 7 Data Distribution

One of the primary requirements of the MAMS system, from both a DOD and an FAA perspective, has been timely dissemination of accurate utilization statistics. There is also a perceived practical and technical need to develop a distributed scheduling system. Practically, many site surveys showed that few sites would relinquish to another agency the necessary control to manage their own airspaces. This has led to distribution of the application, so that each site can continue to manage and control its airspaces locally.

Technically, a distributed approach yields a system that is more tolerant of failures. We have experimented with the process group paradigm for developing a distributed application [Birman *et al.*, 91; Makpangou and Birman, 1990]. This has been useful because the programming model directly supports the hierarchical structure of the DOD command. The hierarchy can be implemented as a set of overlapping process groups.

# 8 Implementation

The current prototype was developed on Sun Microsystems Sparcstation 2s, using C++, the X Window System, OSF/Motif, ORACLE RDBMS, and Isis (a toolkit for distributed applications from Cornell University [Birman *et al.*, 91]). The C++ object-oriented programming paradigm supports our problem domain well.

The prototype has been evaluated through quarterly phased deliveries. Installation of each MAMS phase has been accompanied by user feedback meetings that were an important source of system improvements. Some user needs had to be generalized to arrive at a consistent scheduling interface.

# 9 Further Work

The Gantt chart has proven to have limitations as a user interface metaphor for representing and resolving non-temporal constraints. We are therefore considering adding the ability to view the problem space in additional dimensions. To resolve time and altitude conflicts simultaneously, for example, it would be helpful to the scheduler to view a time versus altitude display of a particular airspace. We have applied the Gantt chart to linear travel routes, such as an MTR, with some success, in that the scheduler has enough information to recognize that there is a conflict, but there may not be enough to gain an intuitive sense of how to resolve a conflict by direct manipulation. We have therefore considered displaying an activity that uses MTR resources along time and distance axis to better represent where a conflict has occurred along a given route. Finally, many airspace managers and some users have expressed an interest in being able to view the use of airspaces on a map in order to better understand the geographic relationships of airspace utilization. This geographic capability would also support interactive designation of new airspace partitions.

It is recognized that some portions of a schedule are repeated weekly or monthly. The users have requested the ability to be able to "paste in" a preset template of events. To support this feature we plan to provide user interface functions to cut a portion of the schedule and save it as a template. The user can then select from a list of templates and paste the events at a new date in the schedule. Invariably users will feel a need to customize their environment, and we plan to evolve the application to incorporate more user preference selections. We plan to run a usability study on the user interface to validate the design thus far.

# 10 Conclusion

The MAMS prototype is a proof of concept system, aimed at improving the scheduling process within a diverse DOD community of schedulers. We found the MAMS scheduling process complex and difficult to specify completely, and thus could not provide a purely automated solution. Our approach has therefore been to support the human scheduler with an integrated, easy to use set of tools. MAMS is an interactive system enabling the scheduler to visualize the interdependence of requested activities and to gauge the impact of modifying a schedule. The user can thus understand the state of a portion of a schedule, and incrementally improve it to develop a fair, conflict-free schedule.

We believe the prototype has helped define the requirements for a future scheduling support system serving a large and diverse user community. The emphasis on the user interface is believed to be appropriate for scheduling problems that have large unstructured domains such as MAMS.

# 11 Acknowledgments

The work described herein is the product of many people. The author wishes to thank the current development team. This work was sponsored by the Electronic Systems Division (AFSC), Hanscom AFB, MA.

# References

[Allen, 1983] J.F. Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[Beard *et al.*, 1990] David Beard, Murugappan Palaniappan, Alan Humm, David Banks, Anil Nair, and Yen-Ping Shan. A Visual Calendar for Scheduling Group Meetings. In *Proceedings of the Conference on Computer Supported Cooperative Work*, pages 279,290, 1990.

[Birman *et al.*, 91] Kenneth P. Birman, Robert Cooper, and Barry Gleeson. Programming with Process Groups: Group and Multicast Semantics. Technical Report TR-91-1185, Dept. of Computer Science, Cornell University, January 91.

[Cooper, 1990] Lynne P. Cooper. User Interface Issues in Supporting Human-Computer Integrated Scheduling. In *Second Annual Conference of the International Association of Knowledge Engineers*, pages 268,274, 1990.

[Halbfinger and Smith, 1990] Eliezer M. Halbfinger and Barry D. Smith. The Range Scheduling Aid. In *Fourth Annual Workshop on Space Operations Applications and Research (SOAR '90)*, pages 280,284. NASA Conference Publication 3103, 1990.

[Makpangou and Birman, 1990] Messac Makpangou and Ken Birman. Designing Application Software in Wide Area Network Settings. Technical report, Dept. of Computer Science, Cornell University, October 1990.

[Mulvehill, 1986] Alice M. Mulvehill. A User Interface for a Knowledge-based Planning and Scheduling System. Technical Report MTR-10175, The MITRE Corporation, Bedford, Massachusetts, November 1986.

[Patterson, 1991] John F. Patterson. Comparing the Programming Demands of Single-User and Multi-User Applications. In *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, pages 87,94. ACM Press, 1991.

[Smith and Katz, 1990] Barry Smith and Joseph Katz. The Range Scheduling Aid. In *The Third International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, pages 275,280, 1990.

[Smith, 1990] Barry D. Smith. A Portable Interface to a Scheduling Aid. In *Second Annual Conference of the International Association of Knowledge Engineers*, pages 208,217, 1990.