View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by NASA Technical Reports Server

N91-20675

The Range Scheduling Aid

Eliezer M. Halbfinger
The MITRE Corporation
Burlington Road
Bedford, MA 01730
emhalb@mbunix.mitre.org

Barry D. Smith
The MITRE Corporation
Burlington Road
Bedford, MA 01730
bds@mbunix.mitre.org

ABSTRACT

The Air Force Space Command schedules telemetry, tracking and control (TT&C) activities across the Air Force Satellite Control Network (AFSCN). This is known as range scheduling. The Range Scheduling Aid (RSA) is a rapid prototype developed by the MITRE Corporation combining a user-friendly, portable, graphical interface with a sophisticated object-oriented database.

The Range Scheduling Aid has been implemented as a set of five modules: the object-oriented database, the constraint-based analytics, the user interface, the multi-user blackboard system, and a dispatcher through which all modules communicate.

The objects in the object-oriented database have a one-to-one correspondence with the objects in the real world. They include satellites, tracking stations, pieces of equipment, requests for service and scheduled tasks.

The analytical capabilities of the Range Scheduling Aid include conflict identification, conflict explanation, and conflict resolution. It also has three error checking routines: time checking, location checking, and visibility checking.

The user interface to the RSA prototype is an electronic clone of the current paper chart. The user interface functions as a query/manipulation language for the database. By pointing at an icon with the mouse, all relevant information is displayed across the bottom of the screen. The user can choose operations from context-sensitive menus or drag an icon to a new location with a single mouse button.

To support multiple users and connections to the outside world, the RSA object-oriented database is populated by a subset of data from a commercial RDBMS. Modifications to an object on screen are posted via a modified blackboard architecture to the RDBMS and to all users affected by the change. Periodically the code requests its messages and updates its local database.

INTRODUCTION

The MITRE Corporation is developing an aid for the scheduling of resources in the Air Force Satellite Control Network (AFSCN). The Range Scheduling Aid (RSA), prototyped through frequent contact with the Air Force schedulers, maintains the currently employed scheduling techniques and efficiencies that have developed over the past twenty-five years, while adding the analytical and database capabilities of a knowledge-based system.

The Current Method of Range Scheduling

The Air Force Space Command (AFSPACECOM) schedules telemetry, tracking and command (TT&C) activities across the AFSCN. The resources of the AFSCN, called the "Range", include 15 remote tracking stations (RTSs) and 55 DOD satellites owned by 10 Mission Control Complexes (MCCs). Range scheduling, as this activity is known, is currently done manually on a three foot wide paper chart. The "acquisition chart" represents time across the x-axis and has horizontal bands delimiting the different tracking stations at which requests for service (satellite contacts as well as maintenance and testing) may be scheduled. The schedule of supports is created by placing one-quarter inch wide pieces of adhesive tape of the proper length at a specific time and RTS on the paper chart. The tapes themselves are color and pattern coded to represent specific satellites and their owning MCCs. An 84 foot long segment of the chart represents one week of the schedule to a granularity of one minute.

The schedulers receive paper forms of requests for satellite contacts, called "program action plans" (PAPs), from each of the MCCs as well as other requests for systems maintenance and testing. The schedulers must consolidate all of the requests into one conflict-free schedule. The schedule is created obeying the constraints of the request, including the request time window, the limited line-of-sight visibility of a satellite to an RTS, the specific set of equipment required for a satellite contact at an RTS, and the availability of the equipment.

The scheduling method, although optimized through its long period in use, is limited. Much of the data necessary to create a schedule, including the equipment available at each site, the equipment required for a satellite contact, the satellites' visibilities from the tracking stations and many of the PAPs, is available to the schedulers only in hardcopy form. Much training time is spent becoming proficient with the volumes of data. Scheduling depends on the paper chart and can therefore only be done at a single location. The schedule as finalized on the chart must then be manually typed into a computer system to be sent to the RTSs and MCCs. Understandably, many operator errors are initiated in the data input step. As the number of resources and requests in the domain increases, the maximum capacity for manual scheduling is being approached.

Range Scheduling Aid Architecture

The Range Scheduling Aid prototype has been implemented as a set of five modules: the object-oriented database, the constraint-based analytics, the user interface, the multi-user blackboard system, and a dispatcher through which all modules communicate (see figure 1). The majority of code in the RSA is written in Common LISP with only a few graphics functions dependent on the LISP implementation. The single user version of the system, which was the main focus of development during the first two and one-half years of the effort, runs on Symbolics machines, Sun workstations, TI MicroExplorers and Apple Macintosh IIs. The blackboard module for the multi-user version and the interface between the single RSA nodes and the blackboard are written in C. The multi-user version is currently running on Sun workstations.

The Range Scheduling Aid (RSA) combines a user-friendly, portable, graphical interface with a sophisticated object-oriented database. The RSA maintains the appearance and functionality of the paper chart while adding many advanced capabilities to assist the scheduler. A sophisticated object oriented database that represents the real world entities of the Range can be queried directly through the user interface. Analytical routines identify errors and conflicts in the schedule and also generate possible alternative solutions to create a completely conflict-free schedule. The multi-user module allows multiple users to concurrently schedule on separate workstations using data archived and updated to a single COTS relational database management system. The RSA system also allows a schedule to be electronically disseminated to the MCCs and RTSs through the COTS RDBMS rather than requiring error-prone manual input of the data. As up to fifty percent of the tasks are changed in the 24 hours prior to the mission, changes to the schedule in real time can also be done more efficiently and the adjustments disseminated more quickly.

THE OBJECT-ORIENTED DATABASE

The entities in the object-oriented database have a one-to-one correspondence with the objects in the real world. They include satellites and non-flight activities, tracking stations, pieces of equipment, requests for service and scheduled tasks. The database has multiple points of access to facilitate efficient retrieval of data using various identifiers.

A request for service specifies the satellite (or other activity) involved, a list of possible RTSs that will satisfy the request, as well as the time window during which the contact should occur and a preferred start and stop time. Any special equipment needed is also noted. When a request is scheduled, the scheduled task, set for a specific time, duration and location has pointers to the satellite, the RTS, the visibilities for the satellite at the chosen RTS, the equipment specified for the contact, and any other task which is in conflict with it. The

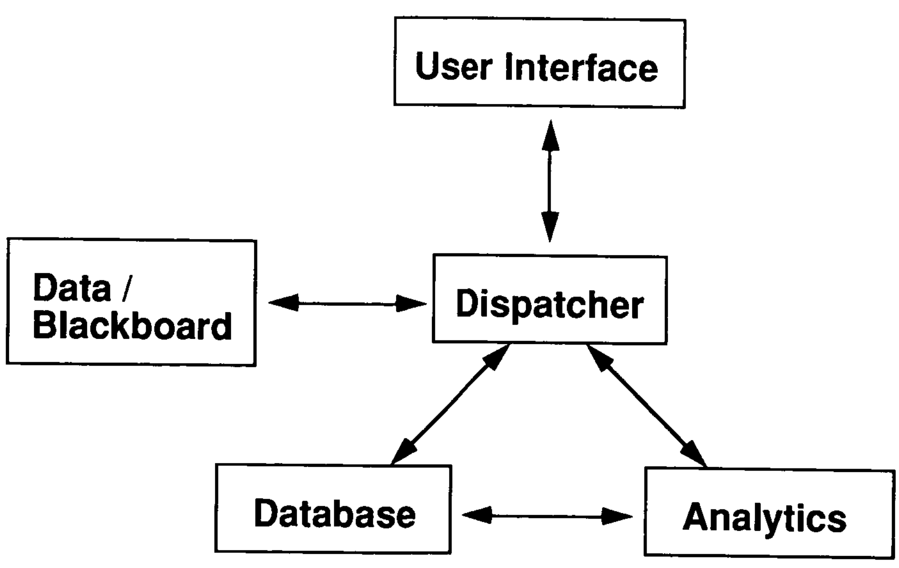

Figure 1: The RSA single node architecture

equipment necessary for the support of each satellite at all of the different tracking stations is stored in the environment table.

Queries to the database are performed exclusively through the mouse-driven user interface. By simply highlighting an icon on screen or bringing up a popup menu all pertinent data is displayed on screen. Data manipulation is also handled through the graphics by either dragging an icon to a different time and RTS or by accessing the menu for an on-screen object.

Analytical Functions

The analytical capabilities of the Range Scheduling Aid include conflict identification, conflict explanation, and conflict resolution. Scheduled requests, also called tasks or supports, are in conflict when they require the same resources for an overlapping period of time. There are three varieties of conflicts: equipment, turn-around-time (TAT), and range conflicts.

An equipment conflict occurs when two tasks at the same RTS are allocated the use of a given piece of equipment at the same time. A turn-around-time conflict is a similar overlap when the time required to set up for a support is subtracted from its begin time. A range conflict tests globally available equipment that can simultaneously support a finite number of tasks across the Range. The conflict identification routine is run whenever a change is made to a support; this may be a change in time, RTS, or equipment.

The conflict resolution procedure will find all of the possible conflict-free options for locating a scheduled task. The user is shown a pop up menu and can choose one option to relocate the task to a new position. When the conflict resolution algorithm is run for a given task, the task is first deleted from the database. A temporary task is created that has a length of the complete time window of the original request of that task. This task is placed at all possible RTSs to identify the other tasks which may possibly lead to a conflict. With this set of pertinent tasks, the algorithm searches for a time gap large enough to accommodate the length of the original task including its turn-around-time. Each of these solutions is returned and the original task is replaced in the database.

The RSA also has three error checking routines: time checking, location checking, and visibility checking. The time of the support is checked with the time window specified in the request for service and an error is flagged if the task is not completely within the window. The RTS at which the task is scheduled is compared with the request's list of possible RTSs for this task. The time of the task for a flight task is compared with the satellite's visibility data to ensure that the satellite has line-of-sight visibility with the RTS throughout the time period when it has been scheduled.

THE USER INTERFACE

The user interface to the RSA prototype is an electronic clone of the current paper chart (see figure 2). The user interface also functions as a query/manipulation language for the underlying database. The user can choose operations from context-sensitive menus or drag an icon to a new location with a single mouse button. The RSA

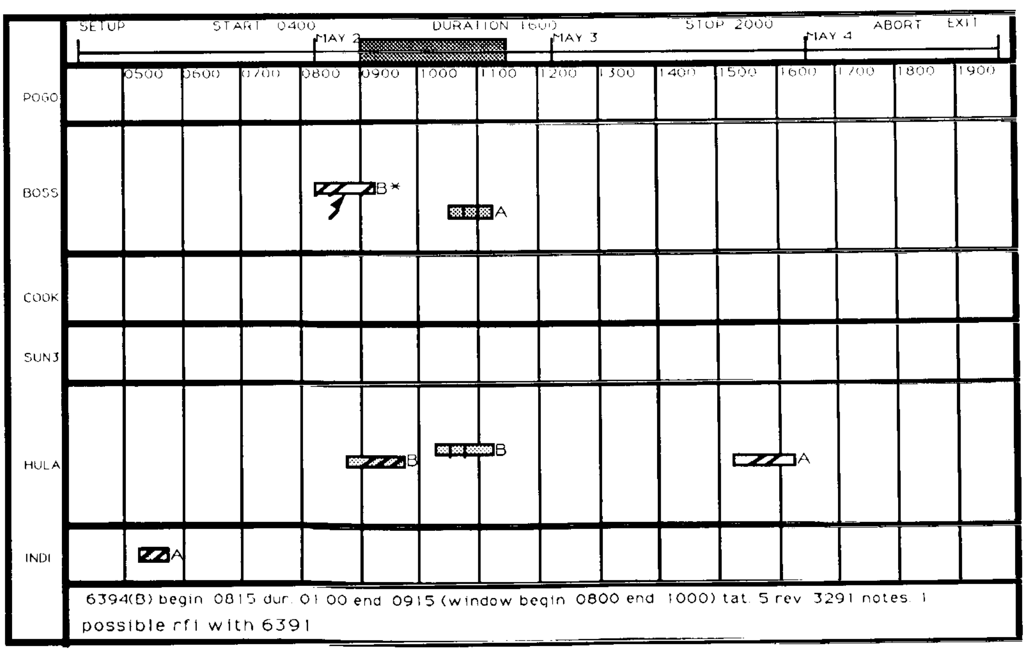

Figure 2: The User Interface

includes its own generic window system supporting several types of pop up menus, color bitmap manipulation and mouse-tracking across panes.

The layout of the screen has been adapted to allow the clear representation of the chart on a commercially available color monitor. Across the top of the screen is a pane with an adjustable timeline marking the time section displayed in the main screen area. The displayed work area can be adjusted by clicking the mouse in the time-pane and then dragging the shaded region to the desired time period. Any length period of time can be selected and the display can represent the schedule clearly with a typical data set for a 24 hour period. Discussions with the schedulers have revealed that although they currently have the ability with the paper chart to stand back and see the complete view of a whole week's schedule, they typically work on periods of twelve hours or less. The horizontal panes on the display coinciding with the available RTSs are also filtered in the RSA through a popup menu. As it is not feasible to show all 15 RTSs with ample work space for each, there is a menu for the user to selectively set the relative heights of each RTS to be displayed. RTSs can be completely hidden from view by setting the relative height to zero. In this way a scheduler focuses on a smaller portion of the scheduling problem but can easily change the display to refer to other data.

On the paper chart, a change in the schedule is accomplished by physically repositioning the tape. Notations concerning a task are written down in pencil on the paper near the placed tape. Changing the position of a tape requires erasing and rewriting its accompanying notes. The RSA defines these tapes as graphical objects. By pointing at an icon with the mouse, all relevant information is displayed across the bottom of the screen. In moving an icon by simply holding down the mouse button and dragging it, all notations for that icon are moved with it. Scheduling a support in error will display a notification menu and a red marker that remains next to the tape. A task in conflict is designated on screen with a pink bar through the patterned icon similar to the method used by the schedulers on the paper chart. To see the explanation of a conflict, a user chooses an option from an icon's menu and the causes of the conflict and the set of conflicting tasks (and equipment involved) are identified.

The RSA also includes icons for service requests, satellite visibilities and notes. Because of the volume of data, the RSA permits the scheduler to display and erase the icons that may not be pertinent to the schedulers current task. The data, however, remains in the database to perform the proper analytical functions.

MULTIPLE USERS - THE BLACKBOARD MODULE

To support multiple users and connections to the outside world, the RSA multi-user version includes the ability to instantiate several scheduling nodes from a single relational database and to manage the concurrency of data among multiple schedulers working on intersecting time periods (see figure 3). The RSA object-oriented database is populated by a subset of the range data from a commercial RDBMS. The central database is completely independent of

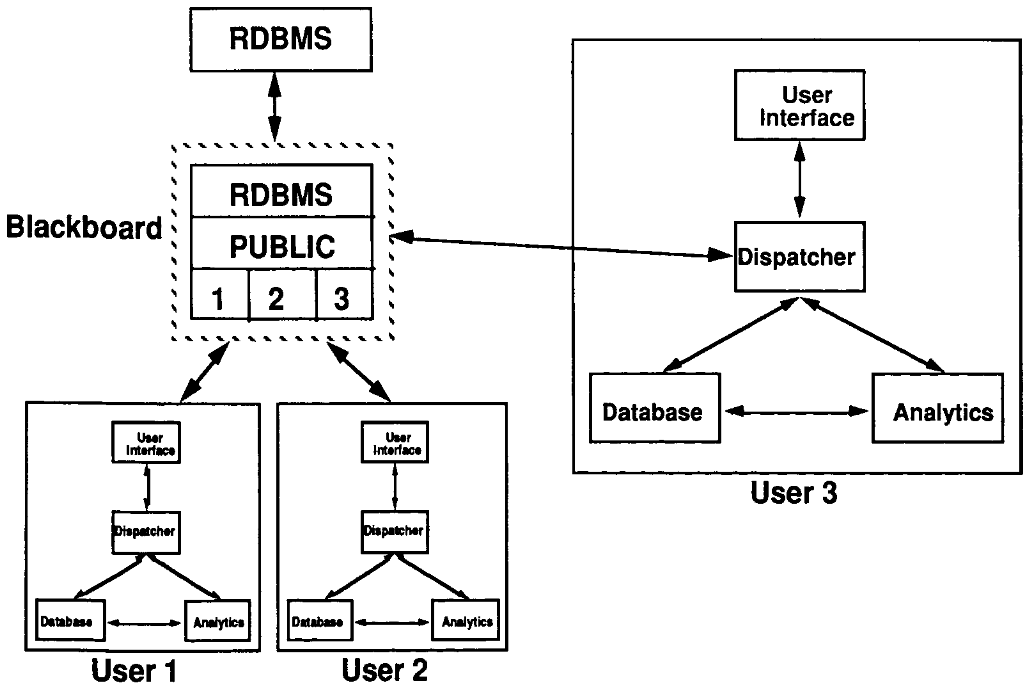

Figure 3: The Multi-User Architecture

the local object oriented databases and does not replace its use in any of the queries or manipulations done by the user. Each of the schedulers can therefore limit its object oriented database to a small portion of data and maximize performance while the complete volume of data remains in the disk-based RDBMS. Modifications to an object on screen (or in the local object-oriented database) are posted via a modified blackboard to the RDBMS and to all users affected by the change. Periodically, each workstation queries the blackboard and retrieves its queued up messages in order to update its local database.

The blackboard module itself has three components: an interface to the relational database management system, mailboxes for each of the active scheduler workstations, and a public area with data accessible to all of the scheduler nodes. The individual workstation passes the data for a message from LISP through a bidirectional stream to a background process. This process (written in C) is awakened and transmits the message across the network to the blackboard server using the SunOS Remote Procedure Call (RPC) interface. The server parses the message, and performs the necessary functions. If an update to the relational database is necessary, the data is formatted as required for accessing the database management system and the Standard Query Language (SQL) code is executed. The message, including any necessary data retrieved from the RDBMS, is then posted to the mailboxes of all other users interested in this message. The blackboard stores the time period for which each of the workstations is logged and checks whether each message pertains to the workstation. Messages concerning the locking of objects for update are posted to the public area. Before a scheduler is permitted to alter the data of an object, an exclusive lock must be successfully placed on the object. If another scheduler has locked that object, the modification is refused and the scheduler is notified with a popup menu.

CONCLUSION

The Range Scheduling Aid has been a rapid prototyping effort whose purpose is to elucidate and define suitable technology for enhancing the performance of the range schedulers. Designing a system to assist the schedulers in their task, utilizing their current techniques as well as enhancements enabled by an electronic environment, has created a continuously developing model that will serve as a standard for future range scheduling systems. The RSA system is easy to use, easily ported between platforms, fast and provides a set of tools for the scheduler that substantially increases his productivity.

Acknowledgements

This work has been funded by the US Air Force Systems Command, Electronic Systems Division at Hanscom AFB.

References

Hayes-Roth, Barbara, "A Blackboard Architecture for Control," Artificial Intelligence 26:3, June 1985.

Smith, Barry D., "A Description of the Range Scheduling Aid Database," MITRE Working Paper No. 28025, The MITRE Corporation, Bedford, MA, January 1989.

Smith, Stephen F., "Constraint-Based Scheduling in an Intelligent Logistics Support System: An Artificial Intelligence Approach," final report for AFOSR Contract No. F49620-85-C-0054 of the Electronics and Material Sciences Department of the Air Force, January, 1987.