Center Director's Discretionary Fund Proposal

Title

Request-Oriented Scheduling Engine (ROSE)

Responsible Individuals

Principal Investigator: John Jaap, FD42

Mr. Jaap has been employed at the Marshall Space Flight Center since 1980; he has developed mission planning software since 1972, space activity scheduling software since 1978, and web applications since 1995. Mr. Jaap received a Bachelor of Arts degree in Mathematics from Mississippi State University. Mr. Jaap is the recipient of the Silver Snoopy, the NASA Exceptional Achievement Medal, the Space Act Award, and a Space Flight Awareness Honoree.

Co-Investigator: Elizabeth Davis, FD42

Ms. Davis began her employment by NASA at the Marshall Space Flight Center in 1980. She has developed mission planning software since 1977. In 1978, she began concentrating on space activity scheduling software. She has been involved in web application development since 1997. Ms. Davis holds a Bachelor of Science degree in Mathematics from Middle Tennessee State University. Ms. Davis is the recipient of numerous awards, including the Silver Snoopy and the Space Act Award.

Shared Qualifications and Experience:

During the Spacelab era, the investigators were the principal developers of the scheduling software used for all MSFC-managed missions and some of the missions managed by DLR and JSC. They were also part of the team that scheduled payload activities and worked at console positions on various missions. Their software for collecting users' requirements for space station payload scheduling was published in *NASA Tech Briefs*. They have published numerous technical papers related to planning and scheduling software and concepts; a combined bibliography is included in Appendix A.

Scope of Work:

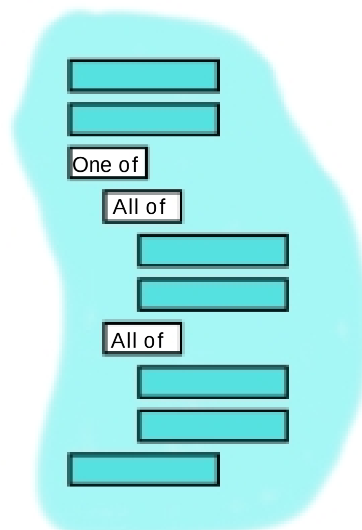
Objective:

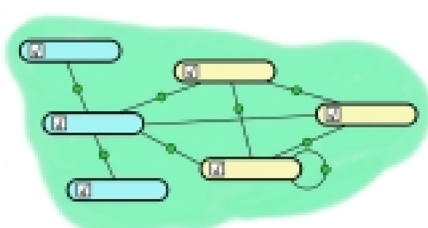
The primary objective of the proposed research is to develop a new scheduling engine that will be superior to any scheduling engine available for space activities, large construction projects, and new product development scheduling. To that end, the following objectives are defined.

Scheduler – The proposed scheduling engine will be an incremental scheduler that processes a single request (adding one or more tasks to the timeline) and then waits for the next request. Of course, each request will be scheduled within the available resources and will meet all the temporal constraints of the scheduling request. An initial profile of resource availabilities will be input from à priori sources. All incoming requests will be handled by a standalone server application written in a language such as C++. The name ROSE is the acronym for Request-Oriented Scheduling Engine, which describes the engine from the user's viewpoint.

Candidate Technologies Investigation – Several candidate technologies are being considered for the scheduling engine. The most likely technology is an algorithm-based search for available "legal" times to schedule all the individual activities of a sequence combined with rule-based selection of the "best" slots in which to place the activities while meeting the temporal (and other) constraints of the sequence. The search for legal slots will incorporate search-space limiting methods such as tree trimming and heuristics. To provide the needed response time of an interactive system, multi-threading will be investigated.

Innovative Modeling – Scheduling requests are database ***models*** of the sequences of activities to be scheduled. The modeling methodology that the proposed scheduling engine will use is based on activities and sequences. *Activities* define the resource requirements (with alternatives) and other quantitative constraints of tasks to be performed. Activities are defined by an outline graphics paradigm. Requirements may be grouped into "all-of" groups or "one-of" groups. Groups may be arranged in a hierarchy. Drag-and-drop is used to define the hierarchy. Constraints are selected from predefined lists. The values of constraints are entered via dialog boxes. *Sequences* define the relationships between activities. Sequences may also define relationships with other sequences, as well as with activities and sequences of other payloads or systems. Sequences are described by a network graphics paradigm. The spatial location is used only as a visual aide; actual sequencing constraints are entered into the nodes (the small circles on the figure) via dialog boxes. Temporal relationships include during, sequential, disjoint, overlap, carry-through, and interruptible. Other relationships are resource lock-in and repetition count.





ROSE will schedule the following modeling innovations:

- Activities can include non-homogeneous optional groupings. For example an activity may require crewman 1 and the 35mm camera or crewman 2 and the digital camera.
- Sequences may include sub-sequences.
- Sequences may include "percent-coverage" relationships (downlink must be available for 80% of the time).
- Sequences may specify alternate scenarios (if available, use real-time video; otherwise, record and downlink later).
- Sequences may include optional tasks.
- Sequences may include resource lock-in (if crewman 1 is selected for the first activity, then he must be selected for subsequent activities).
- Sequences may include resource carry-through tasks (if the start of this task is delayed, then specified resources are consumed during the delay).
- Sequences may include preemptable or interruptible activities and specify the resources consumed by and during the interruption.

Web-Based Interface – ROSE will be deployed "on the web" so that geographically dispersed users can submit their scheduling requests. Users with a personal computer or laptop will connect to the ROSE web site via a web browser, thereby automatically downloading a Java applet. Using this applet, they will graphically define the resource and temporal requirements of the tasks to be scheduled and submit them to the scheduling engine. ROSE must be of *transaction quality*, so that when a remote user schedules something, it is certain to be in the timeline no matter what software or hardware failures occur after the user signs off. To accomplish this goal, ROSE will be layered on a COTS database that provides transaction-quality storage.

Future Applications – Not only would the scheduling engine be useful for a major upgrade to the Space Station operations (such as handover to a non-government organization), it would also be valuable for planning and scheduling activities during a Mars mission or for development or construction of a new launch vehicle.

Diversification – While the scheduling engine will be developed with space activity scheduling in mind, recent investigation has shown a strong parallel to construction planning and new product development. In this context, the scope of a new construction project is similar to a large bridge or a skyscraper, and the scope of a new product development is similar to a new airplane design. The planning and scheduling of subcontractor activities in construction planning and new product development is similar to the planning and scheduling of onboard experiments in space activity scheduling. In both cases, planning and scheduling requires the coordination of a large number of interdependent activities. The requirements modeling method of ROSE has been demonstrated to meet the requirements for space activity scheduling and is believed to meet the modeling requirements for construction planning.

Demonstration – ROSE will be demonstrated for scheduling space activities and for either new product development scheduling or construction planning and scheduling. In-house resources will be used to develop and demonstrate the applicability of ROSE to space activity scheduling. The investigators have access to the database of scheduling requests that are being submitted for the International Space Station and have developed an advanced concept for utilizing ROSE for space station. A contract will be let to Mississippi State University to develop a demonstration of ROSE for construction planning or new product development planning. Dr. John M. Usher will be the point of contact at Mississippi State for this contract. During Dr. Usher's recent tour as a Summer Faculty Fellow, he evaluated the ROSE modeling methods and recognized its usefulness for modeling construction projects and new product development.

Plan:

- **Modeling –** Implement additional modeling enhancements as required.
- **Web-Based Architecture –** Evaluate and enhance the existing architecture if necessary.
- **Activity Scheduling –** Design and implement a search algorithm to find the time slots at which activities may be legally scheduled.
- **Sequence Scheduling –** Design and implement a rule-based method of placing all the activities of a sequence onto the timeline.
- **Demonstration –** Demonstrate ROSE for space-activity scheduling using existing database of requirements and demonstrate ROSE for project management (a large construction project or new product development) with the aide of Mississippi State University.

Anticipated Results:

The scheduling engine developed by this research will be a giant step forward in the science of planning and scheduling. This advancement in the state-of-the-art will be demonstrated for space activities and for either a large-scale construction project or new product development.

The current operations paradigm has the Space Station users submit their requirements; NASA schedules the payload tasks and publishes the timeline. The users request changes; NASA changes the timeline; etc. This loop continues up until the day the tasks are done on board.

A ROSE-based system would allow the Space Station users (the payload developers themselves) to directly produce the timeline that best meets their science needs. The feedback loop would be greatly reduced or eliminated, thereby shortening the development cycle. Fewer NASA mission planners would be required. If adopted for space station payload scheduling, the results would be a payload scheduling system that is truly cheaper, faster, and better.

Related Efforts:

Past Efforts:

- The user interface for most of the modeling enhancements has been developed and deployed in the Interim Users' Requirements Collection (iURC) program (Jaap,Meyer,Davis 1997).
- A recent Summer Faculty Fellow has evaluated the modeling methodology for construction projects and new product development, and has suggested improvements to modeling for space activities.
- The groundwork for developing ROSE as a web-based application has been put in place. Two papers have been published which define the web-based architecture (Jaap,Davis, 2000) and a possible web-based application (Jaap,Muery, 2000) of the proposed scheduling engine
- The investigators proposed by this CDDF are the authors of the "Experiment Scheduling Program" which was used to schedule the payload activities of most Spacelab flights including several controlled by DLR (Germany) and JSC. Selected features of that scheduling engine (Jaap,Davis, 1988) will be considered for inclusion in ROSE.
- One of the investigators has researched a scheduling engine based on an object-oriented approach using a multi-leveled requirement hierarchy. Selected results of this research (Davis, 1997) will be considered for inclusion in ROSE.

Current Efforts:

- Continued refinement and enhancement of the user interface as part of normal operations and maintenance of the iURC program.

Proposed Efforts Through Other Channels:

None.

Resource Requirements:

- Q1: $3,000 – Software Subscription
- Q2: $4,000 – Hardware refresh
(Hardware delivered in Q3 or Q4)
- Q5: $3,000 – Software Subscription
- Q6: $20,000 – Contract with MSU
(Task to be completed in Q7 and Q8)

- Continuous: 1.5 FTE

Two-Year Totals:
$30,000
3 FTE

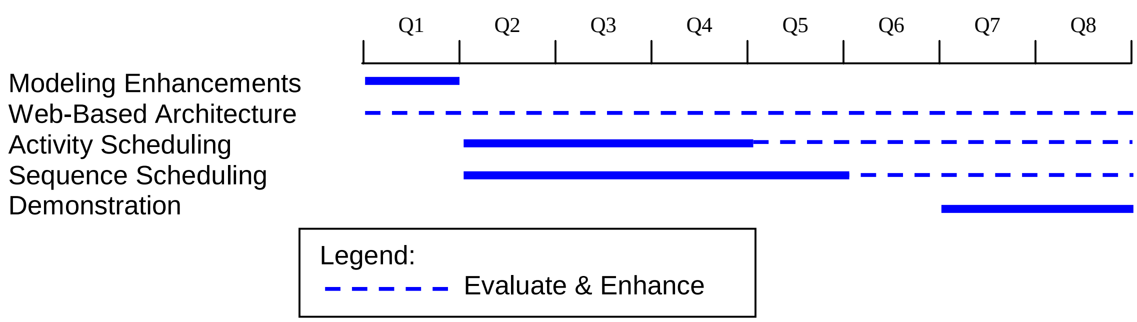
Justification for Use of Discretionary Fund:

As we planned the payload activities for Spacelab missions, we learned a significant and valuable lesson: *regardless of the quality of the automatic scheduling engine, if the modeling methodology does not support the representation of the real-world requirements, then the user must build the timeline manually with an editor*. It is also true that having a good modeling methodology without an automatic scheduling engine forces the user to build the timeline manually with an editor. The key to reducing costs is a *usable* automatic scheduling engine coupled with a modeling methodology that supports modeling the real world with high fidelity.

An innovative modeling methodology that will support automatic scheduling has been defined and partially implemented. This proposal requests funding to develop an automatic scheduler, ROSE, to match the modeling. ROSE can eliminate the costs and impacts of building a timeline manually. Because ROSE is web-based, it could be deployed so that the payload developers themselves schedule their payload activities – further reducing cost and compressing the program schedule. With forethought applied during the definition and development phase, ROSE can also be used to schedule construction projects and new product development such as those that might materialize as part of the Space Launch Initiative.

Schedule:

Estimated completion date: September, 2003



Appendix A: Combined Bibliography

Papers:

- J. Jaap, E. Davis, Can Customers Schedule Their Own Payload Activities? 2nd International NASA Workshop on Planning and Scheduling for Space, March, 2000, (online: http://payloads.msfc.nasa.gov/ROSE/publications/workshop2000.html)
- J. Jaap, K. Muery, Putting ROSE To Work: A Proposed Application of a Request-Oriented Scheduling Engine for Space Station Operations, SpaceOps 2000, June, 2000 (online: http://payloads.msfc.nasa.gov/ROSE/publications/ROSEconcept.html)
- J. Jaap, P. Meyer, E. Davis, Using Common Graphics Paradigms Implemented in a Java Applet to Represent Complex Scheduling Requirements, NASA Workshop on Planning and Scheduling for Space, October 1997, (online: http://payloads.msfc.nasa.gov/FD40/papers/iURC-1/iURC.html)
- E. Davis, Progress Report on the Development of a Windows NT Scheduler, September 1997, (unpublished), (online: http://payloads.msfc.nasa.gov/rose/project/NTscheduler.html)
- J. Jaap, P. Meyer, Graphical Timeline Editing, 3rd International Symposium on Artificial Intelligence, Robotics, and Automation for Space (i-SARIAS 94), October 1994. (online: http://payloads.msfc.nasa.gov/FD40/GTE/gte_paper.html)
- E. Davis, J. Jaap, A Format for the Interchange of Scheduling Models, 3rd International Symposium on Artificial Intelligence, Robotics, and Automation for Space (i-SARIAS 94), October 1994.
- E. Davis, J. Jaap, The Scheduling Techniques of ESP, Second Annual Workshop on Space Operations Automation and Robotics (SOAR '88), July1988. (online: http://payloads.msfc.nasa.gov/FD40/papers/SOAR-1/soar.html)
- J. Jaap, L. Stacy, Space Station Payload Operations Scheduling with ESP, Second Annual Workshop on Space Operations Automation and Robotics (SOAR '88), July 1988. (online: http://payloads.msfc.nasa.gov/FD40/papers/SOAR-2/soar2.html)
- E. Davis, J. Jaap, Experiment Scheduling for Spacelab Missions, Conference on Artificial Intelligence for Space Applications, November 1986.

Documents:

- E. Davis, J. Jaap, Experiment Scheduling Program User's Manual; MSFC Document ESP-USERS-02, September, 1989
- J. Jaap, Experiment Scheduling System (ESS) User Interface Standards; MSFC Document ESS-UIS-01, April, 1990.
- E. Davis, J. Jaap, Experiment Scheduling System (ESS) Library; MSFC Document ESS-LIB-02, November, 1990.
- E. Davis, J. Jaap, Software Requirements Specifications for the Experiment Scheduling Program (ESP) Computer Software Configuration Item of the Initial Mission Planning System; MSFC Document ESP-SRS-02, January, 1993.

- E. Davis, J. Jaap, Model Interchange File Format - Generic; MSFC Document MIF-GENERIC-01, (unpublished).
- E. Davis, J. Jaap, Experiment Scheduling for Spacelab Missions, Conference on Artificial Intelligence for Space Applications, November 1986.

Presentations:

- “Web-based Request-Oriented Scheduling Engine (ROSE)” The Sixth Annual Meeting and Exposition of The American Telemedicine Association, (2-day continuously running demonstration), June 2001.
- “Java at Work – Web-Based Data Entry of Space Station Payload Requirements for Activity Planning and Scheduling” Technical and Business Exposition and Symposium, (2-day continuously running demonstration and posterboard), May 1997.
- “Experiment Scheduling Program Cans, Can'ts and More”, 3rd Annual Space Station Freedom Scheduling Workshop, December 1992.
- “Payload Activity Scheduling”, Advanced Technology Activity Committee, February 1992.
- “Payload Activity Scheduling for Spacelab Missions”, 2nd Annual Space Station Freedom Scheduling Workshop, September 1991.
- “Interim Mission Planning System - Activity Scheduling”, User Operations Working Group, Mission Planning Workshop, May 1990.
- “‘Final’ Scheduler”, User Operations Working Group, Mission Planning Workshop, May 1990.
- “Experiment Scheduling for Space Missions”, Fourth Conference on Artificial Intelligence for Space Applications, (3-day continuously running demonstration and posterboard), November 1988.
- “Using ESP”, 40-hour training course presented to personnel of Deutsche Forschungs- und Versuchsanslalt für Luft- und Raumfahrt (DLR - the German Space Agency) in Köln-Porz and Oberpfaffenhofen Germany, July 1985