In-Space Crew-Collaborative Task Scheduling

John Jaap, Patrick Meyer, Elizabeth Davis, and Lea Richardson
National Aeronautics and Space Administration,
Marshall Space Flight Center, Alabama, 35812

I. Introduction

FOR all past and current human space missions, the final scheduling of tasks to be done in space has been devoid of crew control, flexibility, and insight. Ground controllers, with minimal input from the crew, schedule the tasks and uplink the timeline to the crew or uplink the command sequences to the hardware. Prior to the International Space Station (ISS), the crew could make requests about tomorrow's timeline, they could omit a task, or they could request that something in the timeline be delayed. This lack of control over one's own schedule has had negative consequences [1]. There is anecdotal consensus among astronauts that control over their own schedules will mitigate the stresses of long-duration missions. On ISS, a modicum of crew control is provided by the "job jar." Ground controllers prepare a task list (a.k.a. "job jar") of non-conflicting tasks from which jobs can be chosen by the in-space crew. Because there is little free time and few interesting non-conflicting activities, the task-list approach provides little relief from the tedium of being micro-managed by the timeline.

Scheduling for space missions is a complex and laborious undertaking which usually requires a large cadre of trained specialists and suites of complex software tools. It is a giant leap from today's ground-prepared timeline (with a job jar) to full crew control of the timeline. However, technological advances, currently in-work or proposed, make it reasonable to consider scheduling a collaborative effort by the ground-based teams and the in-space crew. Collaboration would allow the crew to make minor adjustments, add tasks according to their preferences, understand the reasons for the placement of tasks on the timeline, and provide them a sense of control. In foreseeable but extraordinary situations, such as a quick response to anomalies and extended or unexpected loss of signal, the crew should have the autonomous ability to make appropriate modifications to the timeline, extend the timeline, or even start over with a new timeline.

The Vision for Space Exploration (VSE), currently being pursued by the National Aeronautics and Space Administration (NASA), will send humans to Mars in a few decades. Stresses on the human mind will be exacerbated by the longer durations and greater distances, and it will be imperative to implement stress-reducing innovations such as giving the crew control of their daily activities.

A. Major Consideration

Implementation of crew collaboration will be driven by one major circumstance – the round-trip light-time delay between Earth and Mars can be up to 44 minutes and will never be less than 6 minutes (Figure 1). Every 26 months, Mars cycles between close approach and far retreat. Barring some breakthrough in propulsion technologies, a mission to Mars will take longer than two years during which the light-time delays will average over 20 minutes. Normal voice conversations will be impossible.

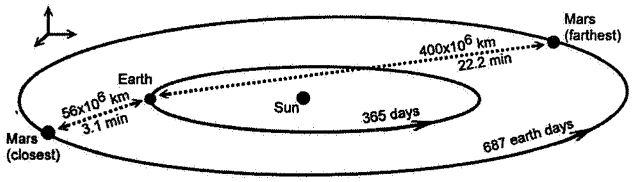

Figure 1. Light-Time Delays to Mars

Instant transfer of information as provided by today's Earth-based internet will also be negated. The communication delays will change the way human missions are operated. The crew will be the first responders to emergencies and mundane anomalies; they will attend autonomously to all alarms, switching the troublesome system to a safe mode and/or making quick repairs and reconfigurations. What we now think of as ground *control* teams will become ground *support* teams.

The inability to have normal conversations with the ground, seeing the Earth only as a point of light, having no immediate return-to-Earth options, interacting only with their companions, having limited food choices, drinking recycled water, and doing the same maintenance tasks each day will make life stressful for the crew. Crew collaboration on the development of the daily timeline for themselves and for the systems they use and maintain will provide them necessary knowledge to execute the timeline and necessary influence so that they will feel that they are in control of their own destiny.

B. Collaboration

Collaboration is working jointly to produce a product or to attain a goal. Usually collaborators share ideas, compromise on goals or methods, contribute where their expertise allows, and jointly arrive at an objective. There are two types of collaboration, interactive and passive. Interactive implies that the collaborators communicate during the collaboration process. Passive collaboration happens without direct communication. An instance of interactive is a group of authors who brainstorm the contents of a paper. An instance of passive collaboration is the progression of teachers who instruct children from early childhood into adults; another instance is the corps of guards who attend all the gates at an arena so that only ticket holders are allowed to enter.

While passive collaboration is based on a concept of operations rather than communication, there are several methods which support interactive collaboration; often, these are used in combinations. The more common methods are listed below in approximate order of productivity.

- Face-to-face – collaborators talk to one another across the table, use whiteboards, share notes, etc.
- Teleconferencing and web conferencing – collaborators simulate being face-to-face using voice, possibly video, and sometimes an electronic white board or a shared computer "desktop."
- Custom software – an application designed to support collaboration on a specific task or product. Internet games are a common example of custom software.
- Instant messaging and chat rooms – collaborators type messages which are displayed almost instantly for each collaborator.
- File transfer – collaborators post and retrieve files using peer-to-peer transfer or a common drop box. Commercial products are available to automate this method.
- Electronic forums – collaborators post messages on a message board.
- Electronic mail – collaborators correspond by sending messages over the internet.
- Postal mail – collaborators correspond by written mail.

For Earth-based support teams and humans on a mission to Mars, several of these collaboration methods will not be available and others (file transfer, electronic forums, and electronic mail) will be degraded. Custom software and the concept of operations can be tailored to deal with the time delays. Collaboration between the in-space crew located at Mars and the ground support will, no doubt, use all the tools available including a delay-tolerant communication infrastructure, delay-tolerant scheduling software, and a specially tailored concept of operations.

C. Integration

In any large community of collaborators, terminology and interfaces are important. The VSE will be the first time that intelligent robots and rovers have been on the same mission as humans. A "master" timeline will be needed that integrates the timelines of the crew, automatic systems and intelligent systems. Any member of the community of contributors may need to view or change any timeline – for example, the crew may need or want to schedule the activities of the rovers. The scheduling requirements for all systems, including the crew, must be described using the same terminology; likewise for the resulting timelines. The user interfaces to the software must be similar for all collaborators – the timeline for a crewmember and the timeline for an automatic system must be viewable on the same display. Additionally, the interface must have special instances for use by the crew in the cabin of the habitat or in their spacesuit while walking on the surface of Mars.

II. Concept of Operations

Human missions to the Moon and Mars will use extraordinarily complex hardware and software systems and will include significant technological and scientific investigations. The missions will extend for years, and ground support will require collaboration from many widely dispersed contributors. The cost to collect the ground-based planning and scheduling collaborators in a central location will be prohibitive. The only affordable solution will be to distribute the planning and scheduling efforts. It logically follows that if the planning and scheduling effort is to be distributed, it can be distributed to the crew on the Moon, in transit to Mars and at Mars. Distributed planning and scheduling has been used to a small extent for *ISS*. Research into concepts of operations [5] which maximize the distribution and the cost savings has yielded viable results.[1] To be successfully accepted and implemented, a distributed concept of operations must be considered and developed beginning as early as possible. The concept proposed here is intended to be a starting point for the concept which will eventually be used to allow full distribution of the planning and scheduling function to all collaborators including the in-space crew. The concept has the additional advantage of giving the crew full autonomy if they should need it.

A. Enabling Principles

The concept of operations is based on several principles which enable collaboration.

1. Adding or removing items from the timeline by one collaborator does not invalidate the contributions of another collaborator.

2. Collaborators need only minimum expertise in the knowledge realm of other collaborators.

These principles apply to the crew as they modify the timeline. The crew is often tempted to make on-the-fly modification to the timeline as they execute it (sometimes we say that the crew interprets the timeline); these principles require that they allow the system to record and verify proposed changes to the timeline.

B. Overview

Figure 2 shows the contributions to the planning and scheduling effort by different collaborators during the preparation of the preliminary timeline on Earth and after the timeline is uplinked to the crew. Text messages, notes, email, and voice memos are not shown on the chart. The development of the timeline is divided into two phases, preliminary and final. Preliminary timelines are developed on the ground using an Earth-based instance of the scheduling system. Before the beginning of each work day, the timeline is

[1] Fully distributed concepts of operations will not be implemented for *ISS* because the paradigm shift would require rewriting current international agreements, incur appreciable one-time costs including new software, procedures, and training, and the phase over would extend almost to the end of the ISS mission.

transferred to a in-space instance of the scheduling system. Crewmembers have time-delayed access to the ground instance and the ground support has time-delayed access to the in-space instance.

C. Widespread Collaboration on Preliminary Timeline

Collaboration is cost effective because those with the best knowledge are able to contribute that knowledge directly. For the same reason, collaboration produces the best product. Collaboration on the planning and scheduling effort for the preliminary timeline is described by listing what each collaborator contributes.

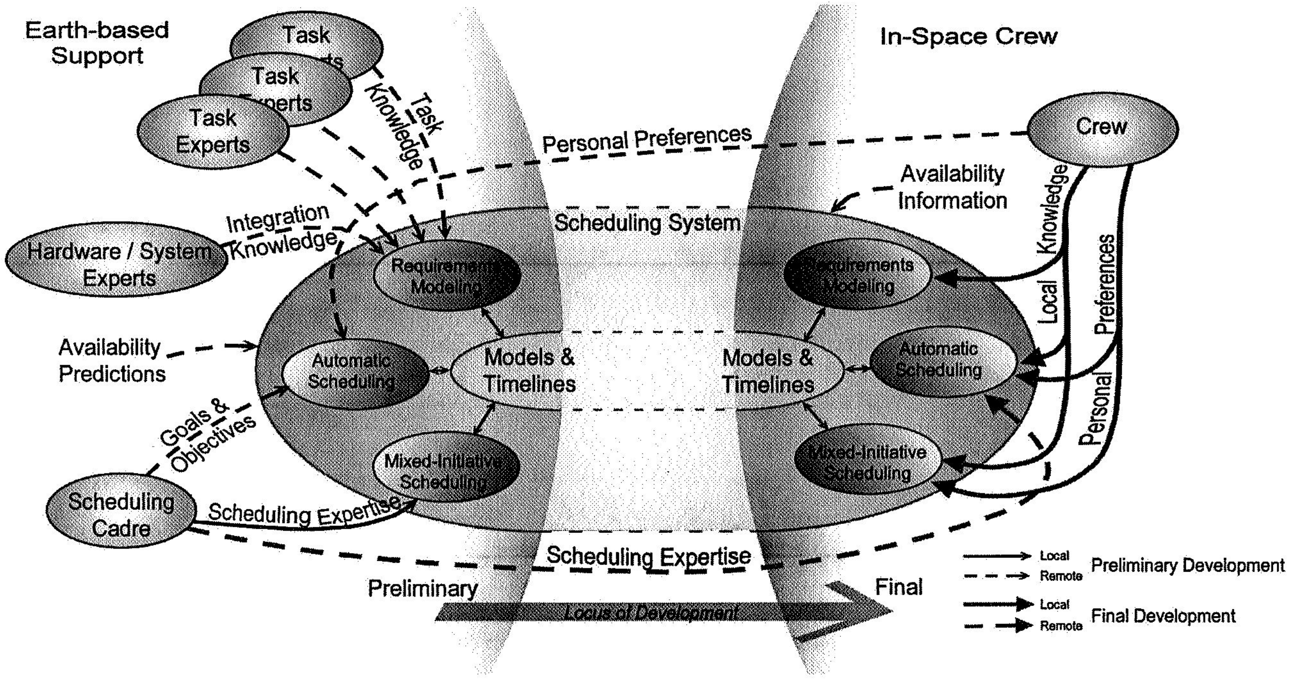

Figure 2. Concept of Operations

- Task Experts – Task experts include hardware developers, scientists, engineering support teams, doctors, astrobiologists and others who have first-hand knowledge about the tasks to be done to maintain the vehicle/habitats and to conduct the exploration and science activities. These experts define the tasks and arrange them into the required sequences in order to accomplish the goals. The task definitions include the procedure descriptions, the equipment (and operation modes) to be used, the conditions required, and the durations. These task models do not directly define the resource amounts required by the tasks; resources are defined by the equipment mode models entered by the hardware and systems experts. The sequence definitions include the temporal relationships between the tasks, relationships to other sequences, windows of opportunity and other data. The information is entered into a central data repository used by the scheduling system.

- Hardware and System Experts – Hardware and systems experts have detailed knowledge about how the hardware is integrated into the vehicle(s) and how that hardware can be used. They also have knowledge about the software systems and how they behave. The experts enter the equipment mode models into the central data repository. For each mode of each piece of equipment, these models define how much of each resource is used. For example, an oven for metallurgical analysis of regolith samples has several operation modes using different resources (power, inert gases, data rates, etc.) and amounts. The hardware and systems experts know how the oven functions and how it is connected to the systems (which power bus, which controller, which video bus, etc.). These equipment mode models are used by the task experts when defining the tasks; therefore, these models eliminate the need for task experts to be hardware experts.

- Scheduling Cadre – The scheduling cadre interacts with program managers, engineering support teams, and the science teams to determine the day-by-day plans. Based on the plan, the cadre submits task sequences to the scheduling engine which adds them to the timeline. The scheduling engine accepts remote requests into a queue of sequences to be scheduled. For each sequence, the engine combines the task models with the equipment mode models to create a complete model including all timing, resources, relationships and conditions, and it then schedules the request. The scheduling cadre can use a timeline editor (mixed-initiative scheduling) to make final tweaks to the timeline. The modeling and scheduling engine are powerful enough that mixed-initiative scheduling will be required only in rare circumstances.

- Crew – Crewmembers can send messages, receive messages, and read the minutes of the various planning group meetings. The crew has first-hand knowledge of the local situation. The crew can collaborate on the development of the preliminary timeline because they can view the task models, the equipment mode models, the developing (partial) timeline and *the crew can submit scheduling requests* to the instance of the scheduling engine currently being used by the ground support teams.

- Other Contributors – Using appropriate software, other contributors provide availability predictions for resources (such as power), conditions (such as daylight or communications), and companion autonomous systems (such as rovers or robots).

In addition to developing the preliminary timeline, the collaborators also develop a task list containing sequences which are candidates for adding to the timeline. These may be purely discretionary or they may be sequences that are planned for the near future.

D. Collaboration on Final Timeline

The final timeline is developed in space. On the day or evening before execution, the preliminary timeline and all supporting data are transferred to an instance of the scheduling system which is co-located with the crew. Crewmembers have immediate and full access to all features of the system and they are the main contributors to the finalization of the timeline. They have first-hand knowledge of the in-space systems; they know their own preferences and needs. They can remove omitted tasks from the timeline so that unused resources are known by the system to be available. They can modify the equipment mode models to reflect actual in-situ configurations, and they can modify task and sequence models to reflect personal experience. They can add to the timeline by selecting items from the proposed task list and submitting them to the scheduling engine so that the tasks are assigned times and so that the resource usage is tracked.[2] The modification to the timeline includes not only tasks done by the crew but also un-attended tasks done by automated or autonomous systems such as robots. In rare circumstances the crew could use mixed-initiative scheduling to modify the timeline; however, mixed-initiative scheduling can consume a lot of valuable crew time, and a system that relies on extensive mixed-initiative scheduling will not be acceptable.

Ground support teams collaborate on the final adjustments to the timeline by reviewing modifications to the models or by updating the models (updating the Earth-based instance and uplinking it). They collaborate on timeline additions by reviewing the crew's modifications and commenting via text

[2] On the *ISS*, task-list items are not submitted to the scheduling engine; the crew selects them and executes them whenever they want (the crew notifies the ground of their actions). For this reason, task list items are limited to those that do not have difficult timing requirements, do not use scarce shared resources, and they are limited to tasks that are done by the crew. Additionally, the *ISS* crew cannot see the scheduling models of the tasks, only the procedures.

messaging. The ground teams can also use remote access to submit new requests to the in-space scheduling engine.

E. Crew Autonomy

When a situation arises in space that necessitates modification and execution of the timeline before the ground can see the change (due to light-time delays), the crew can go ahead and make the change. It is essential for the crew and the in-space vehicle or habitat to be able to function if there is an extended communication loss. The crew may need to extend the timeline for a few hours or for many days. The needed autonomy can be provided by installing in space a complete planning and scheduling system which is sufficiently automated so that the crew can build a timeline with minimum effort.

III. Collaborative Scheduling Software

Enabling widespread and crew-to-Earth collaboration on daily timelines as described in the preceding concept of operations requires major, but evolutionary, changes in scheduling software. In addition, a robust communication infrastructure designed for long round-trip communication delays is needed. Current development efforts are underway on delay-tolerant networking, internet-in-space, and publish/subscribe methods. The supplement section, at the end of this paper, describes interplanetary networks and the message bus architectures. Other communication infrastructure elements are planned. Relay satellites will orbit Mars to provide communication for the twelve hours per Earth day during which the habitat will be on the far side. Additionally, a relay satellite in orbit around the sun will provide communication when the sun is directly between Earth and Mars (in worst cases, the sun-occultation direct communication path could be blocked for two weeks).

The salient features of a scheduling system which can maintain the principles of the collaboration-based concept of operations are use of standard terminology, a comprehensive modeling schema that represents all the constraints, an automatic scheduler that understands the models and produces a desired timeline, remote access to the scheduling system, a human interface that is user friendly to all users including the crew, and the ability to perform over a delay-tolerant network. Figure 3 shows the components of such a scheduling system and how they are used by the collaborators. There would be two instances of the scheduling system, one on Earth and another in space. The Earth-based instance supports widespread collaboration on the preliminary timeline – the ground support teams being the primary contributors and the crew offering limited contributions. The in-space instance supports collaboration on the final timeline with the crew being the primary contributors and the ground support offering limited contributions.

The crew will have little time to work on development of the timeline. Most equipment mode models and task models will be pre-constructed before transferring them to the in-space software. Most of the timeline will be developed on Earth. After the locus of development moves to space, the ground can review timeline additions and changes made by the crew when made far enough in advance of execution; but the crew does not have time to "discuss" the changes via text messages. Some crew changes to the timeline will be in response to the ongoing situation and cannot wait for the light-time delay for review and oversight (these changes can be called real-time replanning). The modeling schema and the scheduling engine must synergistically and automatically produce valid timelines.

A. Equipment Mode Modeling

In space, as on Earth, most tasks are accomplished using equipment. Most types of equipment have modes of operation; e.g., a microwave oven has defrost, reheat, and cook. The microwave oven's power requirement for each mode is predefined. On a space platform, the characteristics of each piece of equipment are well-known to those building and integrating the equipment into the platform systems. The equipment and their modes may be modeled independently of the tasks that will use the equipment. Equipment mode models may use multiple resources and may use other equipment in specified modes. Equipment mode models can describe a hierarchy of constraints and alternate resources. Equipment mode models allow low-level resources such as power to be hidden from the task modeler. The task modeler can only request these low-level resources by selecting equipment that uses them. Using equipment modes to model resource usage is an extension of a method proposed by Hagopian and Maxwell [3].

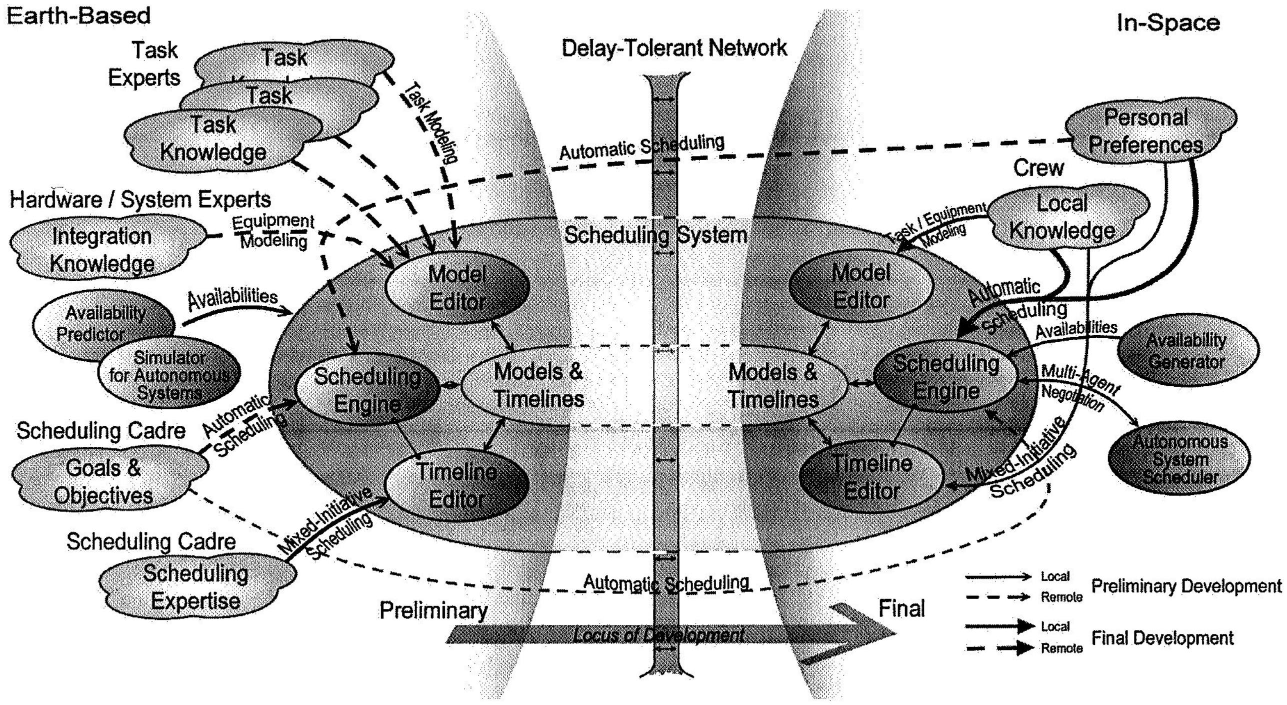

Figure 3. Software and Infrastructure

Earth-based collaborators will build these models using a distributed-via-remote-access[3] feature of the scheduling system to build models in a central data repository. Using a record locking or record checkout method allows multiple users to work at the same time as long as they work on different equipment mode models; revision control is built into the system. All collaborators can view the data of other collaborators.

The in-space crew can view the models in the Earth-based repository via the delay-tolerant network. After the data is transferred to the in-space instance, the crew, using their knowledge of the local configurations, can make needed modifications to the models. The model viewing and editing software must be usable by the crew. The Earth-based support teams can review the crew's changes to the models by either remote display or by transferring a copy of the datasets back to Earth.

[3] The *function* is distributed to remote collaborators by providing remote access to the central location. Accessing the central location could include automatic download and/or update of a local client. In-space software would not be automatically updated.

B. Task Modeling

A specification of the types of goals, states, activities, equipment (resources) and constraints necessary to achieve an objective is called a task model. Task models can range from simple to complex and can depend heavily on the capabilities of the scheduling system that will use them. For the most part, Earth-based task experts will construct, review, and test the models before flight. The modeling schema needs to represent all the requirements (and their flexibilities) so that automatic scheduling can be employed; it also needs to have a straightforward representation so that the task experts (technicians, scientists, etc.) can easily enter the models; and it needs to be easy-to-use so that all collaborators, including the crew, can understand the models. A possible modeling schema has been previously proposed [4]; some of the key features are:

- Decomposition of the operations into salient components – Operations are decomposed into tasks that define resource requirements and sequences that define relationships between tasks. Sequences can also contain other sequences, repeated tasks and sequences, and optional tasks and sequences. Sequences can have multiple scenarios, that is, alternate ways to accomplish the same goal. Tasks can specify alternate resources and variable durations with preferences.
- Rich expression of the relationships between components – Common-sense representations of temporal relationships use everyday concepts like follows, during, and overlap. Innovative enhancements to represent the continuance of resource usage between tasks, the interruption of tasks, minimal percent coverage, and temporal relationships to outside tasks have been added to the modeling schema. The timing of a relationship can have minimum, maximum, and preferred values.
- Public services – The modeling schema also includes the concept of public services – models that are scheduled at the request of another model.

The collaborators use the same type distributed-via-remote-access features as the equipment mode model builders to access a central data repository. Normally, all collaborators can view the data of other collaborators; however, models containing sensitive information may be hidden to some collaborators, but never to the crew. The builders of task models can submit their models to the scheduling engine to test the models for validity and usability. Depending on the concept of operations, the results of scheduling can be part of the developing timeline or they can be tested only.

Like equipment mode models, task models can also be viewed by the in-space crew using the delay-tolerant network. After the data is transferred to the in-space instance, the crewmembers, based on their local knowledge or personal preferences, can modify the task models. Complex syntactical languages, difficult-to-navigate user interfaces, or a modeling schema that does not match the real world could render such a system nearly useless to the crew. The Earth-based support teams can review the crew's changes to the models by either remote display or by transferring a copy of the datasets back to Earth.

C. Automatic Scheduling

Automatic scheduling will be the primary mode of developing the task timelines. An incremental scheduling engine has characteristics which enable supporting multiple collaborators simultaneously contributing to the development of one timeline. An incremental engine is fed by a queue of scheduling requests (sequences of tasks). The engine adds each scheduling request to the timeline without adjusting the times of already-scheduled tasks and without introducing constraint violations or resource overbooking. An incremental engine implements both of the enabling principles of the concept of operations – it add to the timeline without invalidating what is already on the timeline and it doesn't require a user to have global expertise on the items in the timeline. The core logic of an incremental engine is usually an implementation of a greedy algorithm [2]; that is, it makes choices based only on scheduling the current request. These engines may use analytical, heuristic, algorithmic and/or artificial intelligence techniques. Incremental scheduling engine usage is depicted in Figure 4. Some scheduling requests are emphasized to show that they may be very complex. The figure also shows that multiple queues from multiple remote users may be merged into a single queue.

As an incremental scheduling engine processes a sequence, it will search for a near-optimum placement for the multiple tasks of the sequence being scheduled.[4] The tasks always have temporal relationships to each other and may share the same resources. However, all tasks scheduled by previous scheduling requests are locked, and the residual resource profiles are treated as initial resource profiles for the current request.

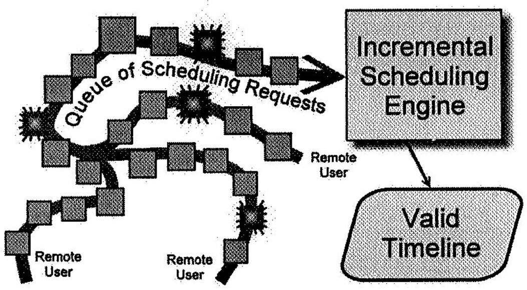

Figure 4. Incremental Engine Usage

Incremental engines can process a request to delete the results of a previous scheduling request. A request to delete an item on which another items depends (reverse dependency) will fail. Deleting only the requested items would create an invalid timeline; deleting dependent items would violate the tenant that processing a request does not affect what is already on the timeline. Of course, after reviewing the failure report, the user could formulate a deletion request to include all the dependent items. Deletion request are moved to the head of the queue to free up resources and reduce the likelihood of a dependency being established.

Incremental engines can also process a request to replace a previous scheduling request. When replacing items on the timeline, resources are reused and reverse dependences become additional constraints. For example, suppose sequence A and D were scheduled by different requests. Suppose sequence A contains task T and sequence D had a requirement to be scheduled after task T. Further suppose that a request is made to replace sequence A with sequence B. Then B must include a replacement task T and the engine will schedule sequence B so that the replacement task T is scheduled before sequence D.

Collaborators use remote access to submit scheduling, deletion, and replacement requests from the repository of models. Multiple collaborators can submit to the queue simultaneously. The scheduling system provides the ability for all users to see the current timeline as it is being developed. The crew can used the delay-tolerant network to submit scheduling requests and to view the timeline.

Incremental scheduling engines and their usage is discussed in detail by Jaap and Phillips [6].

D. Mixed-Initiative Scheduling

Mixed-initiative scheduling refers to building a timeline using a timeline editor; i.e., it is a manual process. Mixed initiative is used when the contributor knows requirements that are not described in the scheduling requests, the scheduling engine is weak, only a few new requests are to be added to the timeline, or the user wants to fully control the results (Figure 5). Reckless user of mixed initiative scheduling will violate the enabling principles of the concept of operations – moving already scheduled items can invalidate what is already scheduled or require the user to have global knowledge of the scheduling requirements.[5] Mixed-initiative scheduling is also discussed by Jaap and Phillips [6].

[4] Incremental engines do not provide global schedule optimization.

[5] Mixed-initiative scheduling does not automatically provide global schedule optimization. However if the user is an expert and the problem is straight-forward, global optimization might be achieved.

Mixed-initiative schedulers usually have logic to help the user avoid violating constraints and allow the user to override constraint limits. If the models are complete, the editor might invoke iterative-repair logic to move other tasks and eliminate constraint violation introduced by a manual edit. The editor might invoke an incremental scheduler to make slight adjustments to the user's input; this feature is called "snap-to." Additionally, the editor might use an incremental engine to suggest times where tasks could be placed without introducing constraint violations.

The timeline editor will display considerable information about the timeline, the models, and the availabilities. It will be closely coupled to the data repository and the scheduling engine. It will not operate over the delay-tolerant network and will be only available to local users. During the development of the preliminary timeline, a cadre of highly skilled users, called the "scheduling cadre," will be the primary users. After the locus of development moves to space, the crew can use mixed-initiative scheduling, but will do so rarely.

E. Resources, Conditions, and Autonomous Systems

Scheduling depends on the availabilities of resources which are scarce and shared, on conditions which occur intermittently, and on the available of other, possibly autonomous, systems. Power, cameras, and storage space are examples of resources; daylight, communication links, and sand storms are examples of conditions; and self-scheduling robots and rovers are examples of autonomous systems. During the development of the preliminary timeline, availabilities are predicted by Earth-based software and autonomous systems are simulated. During the finalization of the timeline in space, the availabilities of resources and future conditions are predicted by adjunct software. For example, on Mars a weather prediction program will warn of approaching sand storms.

The in-space scheduling systems will negotiate with the rovers and robots to jointly arrive at a common timeline. The negotiation might go like this:

- The main scheduling system will ask a robot if it is available to do a certain task at a certain time.
- The robot scheduler will attempt to the schedule the requested task; if possible, give an affirmative answer; if not, an alternate time might be suggested.
- After the main scheduler has found acceptable times for all tasks of the scheduling request, it would send the robot a commit message and the robot would complete its scheduling.

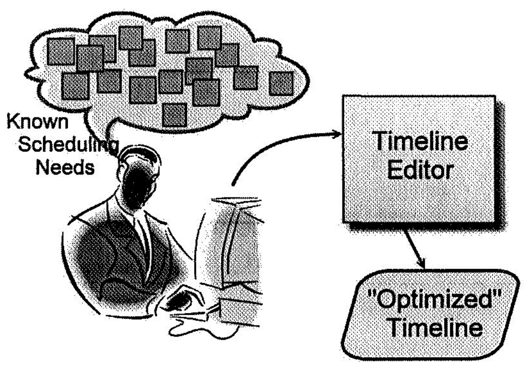

Figure 5. Mixed-Initiative Scheduling

F. Terminology and Standards

Historically, the planning and scheduling for space domain has been fragmented along human vs. non-human flights and also by the sponsor of the missions (NASA center, ESA, etc.). The fragments have each evolved their own terminology and standards for expressing planning and scheduling requirements, results and displays. Many of the differences are semantic – a consumable resource and a depletable resource have the same characteristics, a state is a condition, etc. Some of the differences are substantive – activities have durations and are delimited by start and stop events. In one scheme, temporal relationships may be expressed relative to the activity; in another scheme, they are relative to the delimiting events. "Activity A occurs during activity B" can be directly expressed in one scheme but in the other becomes "start of B occurs on or after start of A and stop of B occurs before or on stop of A." Goals may be chains of events or sequences of activities. Similar differences exist for the results and the displays. Human missions to the Moon and Mars will bring together contributors from many backgrounds because the missions will include humans, robots, rovers, automatic systems and remote-commanded systems.

Collaboration by a large diverse group of ground support persons and by the crew will require a standard terminology be adopted by all. It is especially important that the crew, at a minimum, be able to look at any timeline (for any system), understand it, comment on it; and, in some cases, modify it. The crew is dedicated to planning and scheduling and should not devote time and effort to learning different scheduling terminologies or methods. Fortunately, existing standards bodies, such as the Consultative Committee for Space Data Systems (CCSDS), are available to apply their expertise in establishing, negotiating, and documenting the needed data standards.

G. User Interfaces

Good user interfaces with the planning and scheduling software are necessary for successful collaboration between diverse contributors. The user interface needs to allow those who are not experts in planning and scheduling to perform as "virtual" experts. Like the driver of a car who is able to steer the car without concern for how to manipulate the steering wheel, the user of the planning and scheduling software should be able to produce the desired timeline without concern about the user interface.

To support widespread collaboration and remote access across the solar system, the scheduling software user interfaces will need some special features. One might say the software must have delay-tolerant user interfaces. Delays will be caused by light-time and by loss of signal to relay satellites. Some of the ameliorating features of the software are:

- Maximizing local processing to avoid delays.
- Continuing to interact while waiting for a reply to a message.
- Providing visibility into the stack of sent messages showing time remaining until a response is expected.

The crew will have special user interface needs. The console in the habitat or vehicle may have the standard user interfaces, but the interfaces used when outside will have special requirements. Figure 7 shows a PDA-like device attached to a crewmember's sleeve. This device is in communication with the planning and scheduling system in the habitat. The crewmember should be able to look at the timeline and possibly modify the timeline using the sleeve-attached device.

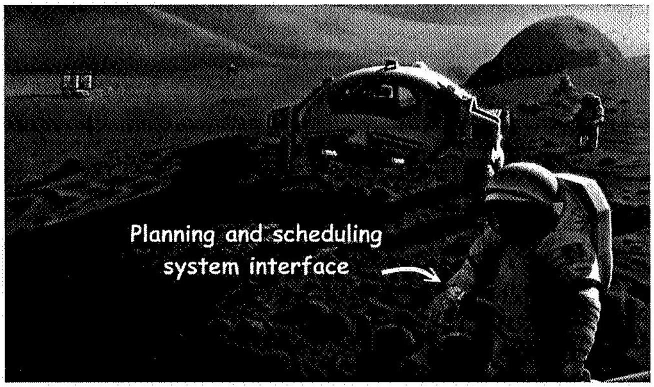

Figure 7. User Interface for the Crew

IV. Conclusion

As humans explore regions of space where the Earth appears as a mere point of light, and round-trip communication delays exceed half an hour, it will be imperative that the humans on the journey exercise significant control over their daily timeline and the timelines of their companion systems. Yet the crew does not have the time to do all the scheduling. Scheduling will be a collaborative effort between multiple ground support teams and the crew.

A new concept of operations is based on the principles that adding to the timeline doesn't invalidate what is already on the timeline and contributor do not need global expertise about the items on the timeline. The concept calls for developing the preliminary timeline on Earth, having the Earth-based teams as the primary contributors and the crew providing only minimal input. Once the preliminary timeline is uplinked to the in-space location, the crew will become the primary contributors and the Earth-based teams will provide only minimal input. This concept is tailored to function effectively in the delayed communication environment.

Planning and scheduling software will evolve to meet the needs of the new concept of operations. Automatic scheduling will become the normal way to produce the timeline. Modeling will be partitioned into equipment mode modeling and task modeling so that different experts with different knowledge can contribute effectively. An equipment mode model will reflect the different modes in which the equipment operates and will book all resource and condition requirements. Task models will be grouped into sequences showing the temporal relationships of the tasks. An incremental approach to scheduling will enable multiple experts to schedule sequences without impacting sequences scheduled by other experts (the crew being among these experts). Because the models reflect all the requirements and the scheduling engine can produce good timelines, mixed-initiative scheduling will be used rarely.

The synergy of the new concept of operations, delay-tolerant communication and software (including user interfaces), and advances in scheduling technology will allow the in-space crew and Earth-based support teams to collaborate on the scheduling of the tasks done by the crew, the operation of the vehicle or habitat, and the operation of the various near-autonomous robots and rovers.

V. Supplement – Delay Tolerant Networking

Planning and scheduling is only one application that will depend heavily on a communications framework that will carry data between Earth and Mars or in between. Without such a framework, in-space collaborative scheduling would be impossible. The cost of spaceflight being prohibitive as it is, much work will be done to leverage existing technologies and commercial hardware and software to use them and manipulate them to work in these foreign environments. Accomplishing this will necessitate standardizing communication between applications using concepts like a message bus which travels over ubiquitous protocols such as the internet modified for light-time delay.

A. Inter-Planetary Internet

Standardized digital communication devices have revolutionized the way we communicate and interact with information. But the foundation of the technology, TCP/IP, does not translate well in a space environment where line-of-sight connectivity is intermittent, communications have delays and data may have high error rates requiring retransmissions. However, the technology of TCP/IP can easily be leveraged to create Mars-based internet that connects remotely to today's Earth-based internet. The interoperability between the networks is where new technology is needed. Based on on-going research, a new concept, Delay-Tolerant Network (DTN), is evolving to mitigate the delay problem. DTN could employ concepts such as store-and-forward message delivery.

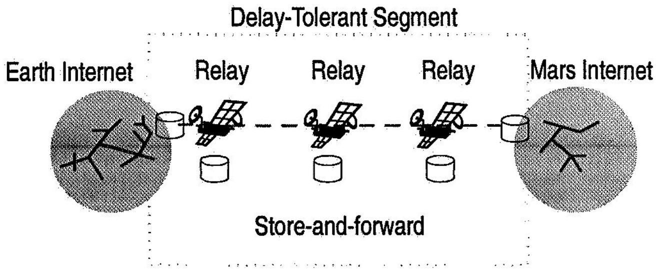

Figure 7. Inter-Planetary Internet

In store-and-forward, relay systems capture incoming data and store it locally before transmitting it to the next link in a chain (Figure 7). If there is a transfer failure, the data will not have to travel the complete distance again. DTNs can be used to route data between two traditional TCP/IP networks almost transparently to the TCP/IP packets concept. However, many of today's applications are not designed with such communication delays in mind. Loss of continual heartbeats or continuous data streams would cause many of today's applications to fail as they wait for the next packet to arrive. Therefore, any applications themselves must be built with data delays in mind. The applications must be more robust and fault tolerant and should not malfunction when there is a long data delay.

B. Message Bus

Message buses standardize information exchange as well as data pathways between two end points. Buses can be layered upon DTNs to provide more abstraction between the software and the communication infrastructure, allowing for the flexibility of architecture upgrades. Most message bus modes include a publish/subscribe protocol, sometimes called "pub/sub," in addition to more direct request/reply connections.

In the publish/subscribe mode, client applications subscribe to information they would like to be provided. As messages pass by, messages that match the requested type are captured by the client API and passed to the application.

VI. References

[1]Compton, W.D., and Benson, C.D., *Living and Working in Space: A History of Skylab*, National Aeronautics and Space Administration, Washington, D.C., 1983.

[2]Cormen, T.H., Leiserson, C.E., Rivest, R.L., and Stein, C., Introduction to Algorithms, 2nd Edition, Sec. 16, The MIT Press, Cambridge, Massachusetts, 2001.

[3]Hagopian, J., and Maxwell, T., "Explicit and Implicit Resources: A Simplified Approach to User Requirements Modeling," SpaceOps 96, Fourth International Symposium on Space Mission Operations and Ground Data Systems, September 1996.

[4]Jaap, J., Davis, E., and Richardson, L., "Maximally Expressive Modeling," Fourth International Workshop on Planning and Scheduling for Space, Darmstadt, Germany, June, 2004.

[5]Jaap, J., and Muery, K., "Putting ROSE to Work: A Proposed Application of a Request-Oriented Scheduling Engine for Space Station Operations," Sixth International Conference on Space Operations (SpaceOps 2000), Toulouse, France, June 2000.

[6]Jaap, J., and Phillips, S., "On Using an Incremental Scheduler for Human Exploration Task Scheduling," 2006 IEEE Aerospace Conference, Big Sky Montana, March 2005.