Integration of On-board EOS Schedule Revision with Space Communication Emulation System

Lina Khatib[1,2], Robert Morris[2], Rich Slywczak[3], and Thong Luu[3]
[1]Kestrel Technology
[2]NASA Ames Research Center-MS 269/2, Moffett Field, CA 94035
{lina,morris}@email.arc.nasa.gov
(650) 604-2972
[3]NASA Glenn Research Center-MS 142/5, Cleveland, OH 44135
{Richard.A.Slywczak,Thong.Luu-1}@nasa.gov
(216) 433-3493

***Abstract*— The need for on-board decision-making for planning science observations on Earth Observing Satellites is based on the fact that the scientific utility of acquiring an image can change dynamically from the time the observation is scheduled to the time it is taken. Currently, Earth observing satellites cannot communicate directly with each other, and can only communicate with ground stations about 5% to 10% of the time. Because of the limited communication windows a ground-based mission scheduler would have little to no opportunity to revise an observation schedule in response to changes in the observation environment. For this reason, a distributed science planning system combining a ground-based scheduler with on-board schedule revision capabilities is warranted. This paper will focus on on-board autonomous planning capabilities for Earth observing satellites, and the design of a set of experiments demonstrating the effectiveness of on-board decision making using the Space Communication Emulation Facility (SCEF) at Glenn Research Center**

I. Introduction

The need for on-board decision-making for planning science observations on Earth Observing Satellites is based on the fact that the desirability of acquiring an image can change dynamically, because of changes in meteorological conditions (e.g., cloud cover), unforeseen events (e.g., fires, floods, or volcanic eruptions), or unexpected changes in satellite or ground station capability. In such cases, satellite resources, such as power and Solid State Recorder (SSR) capacity, can potentially be better utilized taking other images of higher utility. Currently, typical Earth observing satellites cannot communicate directly with each other, and can only communicate with ground stations about 5% to 10% of the time. Because of the limited communication windows, as well as the cost and effort that would need to be expended in revising a mission schedule, a ground-based scheduler would have little to no opportunity to revise the schedule in response to the contingencies that may arise. For this reason, a distributed science planning system combining a ground-based scheduler with on-board schedule revision capabilities is warranted.

This paper describes algorithms for autonomous on-board science planning and execution. It also describes an collaborative integration effort with the advanced satellite control and communications technology developed at the Space Communication Emulation Facility (SCEF) at Glenn Research Center. This integration will enable a demonstration of how advanced communications and scheduling technology can improve the scientific utility of images acquired by Earth observing systems. This paper will discuss a set of scenarios that will be applied as part of the demonstration.

II. On-board Decision-making and execution

A. EOS Science Observation Scheduling

Observation scheduling solves the following problem: given a set of observation requests, a set of instruments for acquiring images, and a set of other resources required to capture the data (e.g. on-board storage), produce a set of assignments of instruments and viewing times to those requests. In addition to specifying desired viewing location, spectral and spatial resolution, requests are commonly ordered on the basis of their importance. In addition, it is possible to measure the outcome of an observation in terms of its scientific utility; a cloudy image is typically of low scientific utility, for example. Observation scheduling thus can be viewed as an optimization problem of maximizing the overall quality of science data acquired, where quality is a composite function of priority and utility [5].

Future requirements for the coordination of science observations suggest the need for a centralized scheduling system that has complete knowledge of the capabilities and status of all satellites and all data requests. Ideally, such a system could revise a schedule at any time, to take account of unforeseen cloud cover, unexpected events such as floods or volcanic eruptions, new data requests, or unanticipated changes in satellite resources. Unfortunately, achieving these capabilities on a single ground-based system is not feasible. This is due to limited communication windows and the cost and effort of revising a mission schedule to accommodate incremental changes to the schedule in response to the contingencies that may arise.

Therefore, we have developed an architecture for science observation scheduling that consists of two main components:

- a centralized scheduling system for multiple satellites, and
- a responsive, but limited, on-board scheduling system for each individual satellite.

These components would communicate as follows. A set of complete sequences of observations generated by the central scheduler is uplinked to the satellite during its communication window, along with a set of alternative observations. Once uplinked, the on-board system will receive inputs consisting of either updated weather predictions, or data analysis results, which will allow it to revise the expected quality of the nominal schedule. This revision could result in a change in the sequence of acquired observations, the result of choosing from the set of alternatives. After the data is acquired, the images are downlinked, and the central scheduler is notified of any modifications made to the nominal schedule by the revision system. This provides part of the input to the next scheduling cycle [2], [3].

B. An Approach to On-board Schedule Revision

The scheduling conducted on an individual satellite would presumably be limited, due to the lack of current information about all the other satellites and their schedules. Further, it is realistic to assume limited on-board memory and CPU, indicating a proportional limitation in the size or complexity of the scheduling problem the on-board system could solve. For this reason, we assume that the decision-making for planning on-board a satellite is limited to *schedule revision*.

An on-board revision system requires addressing the challenges of fast re-planning during execution, which is an emerging research area in planning. In order to model an on-board schedule revision system, we assume that the following information is available to an individual satellite:

- A schedule produced by a ground-based scheduler
- A set of additional observations that would be desirable
- The expected utility value of each observation in the schedule and in the set.
- Storage limitation and requirements (for indirect downloading).
- Dynamic updates of environment and observation status.

We assume that satellites lack global knowledge. They do not know what images have already been taken by other satellites, the capabilities of other satellites, or the schedules given to other satellites. Therefore, we assume that the schedule produced on the ground would be preferred in the absence of any changes in the environment including the actual or expected values of observations. Also, observations in this uplinked schedule will be assigned a bias over unscheduled, or dynamically added, observations.

An environmental update can be in the form of weather change (e.g., unexpected cloud coverage), downlinking status change (e.g., loss of contact with ground station), and/or serendipitous events (e.g., volcano or fire). An observation status update can be in the form of changes to the actual value of observations just completed (due to on-board analysis), and/or updates on the expected values of observations that could be done in the near future (via communication with other satellites, or forward looking instruments).

The basic approach to schedule revision is a greedy one; more observations are enabled than the system expects to be able to keep, and the system discards those of lesser value, as necessary, in order to take observations of higher value. This over commitment helps ensure that a full complement of useful observations will be collected, even if later scheduled observations turn out to be of low value. If instrument slewing is required between observations, then the selection of one observation may make a succeeding observation impossible. Thus, when slewing is necessary, the myopic greedy rescheduler can make poor decisions. To remedy this, the scheduler needs to perform some degree of *lookahead*; that is, it needs to consider the impact of choosing a current observation on future observations. To avoid non-optimal selections, the lookahead should be of arbitrary degree. Testing has indicated that a short lookahead will be sufficient to provide good performance of the rescheduling algorithm, and that the lookahead distance can be determined using the slew rate and off-track pointing limits of the instruments. We have experimented with various lookahead strategies for which details can be found in [2], [3].

C. Evaluation

In order to identify the value of on-board rescheduling, we need to study the expected gain in the value of observations collected, over those that would be taken if we just followed the schedule produced on the ground. This, of course, depends on the frequency and nature of the revisions. So, more generally, we would like to know the net gain in the value of observations collected, as a function of the frequency and nature of value revisions. We would also like to know how this value is affected as storage capacity changes, ground schedule bias is varied, size of extra observations set increases, or as lookahead strategy and depth changes.

For a constellation of satellites, we would also like to know how the net value of observations collected is affected as we vary the amount of overlap in the set of extra observations given to the satellites. If overlap is allowed, there may be value to allowing satellites to communicate with each other to avoid duplicate observations.

Promising experimental results are reported in [3]. Although these results provide a proof of concept of the ideas underlying the algorithms, the true test of the effectiveness of these ideas will be in integrated platforms running realistic scenarios. The SCEF testbed provides the next step in realizing this overall goal.

III. Space Communications Emulation Facility (SCEF)

The Space Communications Emulation Facility (SCEF) provides an environment for researchers from academia, government, and industry to emulate space missions. NASA is designing and developing advanced space missions that will have communication and coordination requirements that have been previously unseen. On the current horizon, the agency will be proposing constellations missions from loosely coupled, such as the Global Precipitation Mission (GPM) to tightly coupled, such as the Micro-Arcsecond X-ray Imaging (MAXIM) mission to the pervasive Sensor Web which combines satellites, aircraft, balloons, etc. While these missions will focus on Low Earth Orbit (LEO) satellites, another direction, as outlined by the President, will be the return to the Moon and onwards to Mars and other bodies in Deep Space. To ensure the success of these missions, NASA is developing an emulation facility to model these types of missions to determine any weaknesses before they launch. SCEF will also provide a vehicle for researchers to modify components on a satellite to support these complex missions. For example, researchers could replace the Command and Data Handling (C&DH) Software, Scheduling Algorithms or the TCP/IP stack. Using SCEF, researchers can make these changes and model typical missions to determine any aberrations during the algorithm and/or system development [4], [6].

A. The SCEF Testbed

NASA is designing a number of satellite mission concepts that combine current satellite technology with advanced communications techniques. The concept of a single satellite that collects and transmits data is transforming into multiple satellites that work cooperatively. The goal will be to develop a tightly integrated constellation that can not only provide communications among satellites but also provide communications with ground stations using relay satellites. These relay satellites could provide the constellation with almost continuous communications to either transfer data or upload commands. While the individual technologies are well understood, there are still a number of questions that will arise from applying them to space communications. How will the satellites communicate? What information will be transferred between them? What protocols will be the most effective? How will space affect communications? The SCEF testbed will be instrumental in answering these questions by allowing projects to test scenarios and understand the communication infrastructure of their missions.

One objective of SCEF is to create a generic emulation testbed that will focus on the communications path that allows data to be routed, either statically or dynamically, from the satellite to the ground station through a set of relay nodes. Currently, the communications path is to route data through a series of relay nodes. In addition, the emulation environment provides the ability to integrate custom code into the environment to test under user defined mission scenarios. This environment provides networks that are very similar to today's terrestrial network using packet switching technologies and common protocols (e.g., the TCP/IP Suite). The origin of the software was developed by the University of Kansas (UofK) under contract with GRC and was called the Space-Based Internet (SBI). The emulation software is still under development, since SCEF is updating and adding new features to the software provide by UofK.

While the current implementation of the software provides a significant amount of functionality and is being rigorously tested, SCEF is also under going a number of enhancements. Currently, each satellite is viewed as a platform with a number of instruments that can transfer data between itself and other instrument(s) or relay platform(s). For the dynamic on-board scheduling code to be properly integrated, the concept of a satellite must be expanded to more accurately reflect a real satellite. Provisions must be made for Command and Data Handling (C&DH), uplink/downlink interfaces, instruments, and scheduling components. SCEF is expecting to add this functionality to the software.

Eventually, the emulation environment will be accessible either within GRC or remotely from outside of the center. Users will be able to create their own scenarios with an arbitrary number of satellites and the characteristics for each satellite. These characteristics include the number and types of satellites, the number and types of instruments, the composition of the network, etc. In addition, they can select type of communications and protocols. These scenarios will run within the emulation environment and the results will be submitted back to the user.

B. SCEF Architecture

The current hardware architecture of the SCEF facility is comprised of 32 nodes and a controller machine; the overall architecture is depicted in Figure 1. The controller will provide two distinct functions. First, it is responsible for starting the emulation and providing user defined specifications to the each of the nodes, such as the number of instruments, custom algorithms replacements, orbital parameters, etc. Secondly, it will provide the user interface to the emulation system to provide output in both graphical and textual form. Each of the 32 nodes in the emulation represents either satellites or ground stations. The facility emulates LEO based architectures, but eventually the nodes will be able to represent deep space objects with representative characteristics. Each of the 32 nodes in SCEF is a Pentium 4 class machine that is running at 3.06 GHz and has 1 GB of memory. The controller nodes are Pentium III class machines that are running at 900 MHz with 4 GB of memory. The SCEF project has standardized on the Fedora Core I distribution of Linux.

Figure 2 shows the representative architecture for each node in the emulation system; this figure is a representative architecture, since the user can customize the satellite to meet their research objectives. The design of the nodes was based on a component based architecture where each subsystem is modeled as an independent module. The communications mechanism between each of these subsystems is based on the TCP/IP protocol suite, since the general consensus of the satellite community is that the trend of satellite communications will move towards some type of IP-based communications in the near future. For example, each node will contain an on-board clock that can be queried by other on-board components. For example, if the on-board scheduler requires the time, it can issue a request to the on-board clock and receive the current time. The on-board clock will be responsible for keeping the

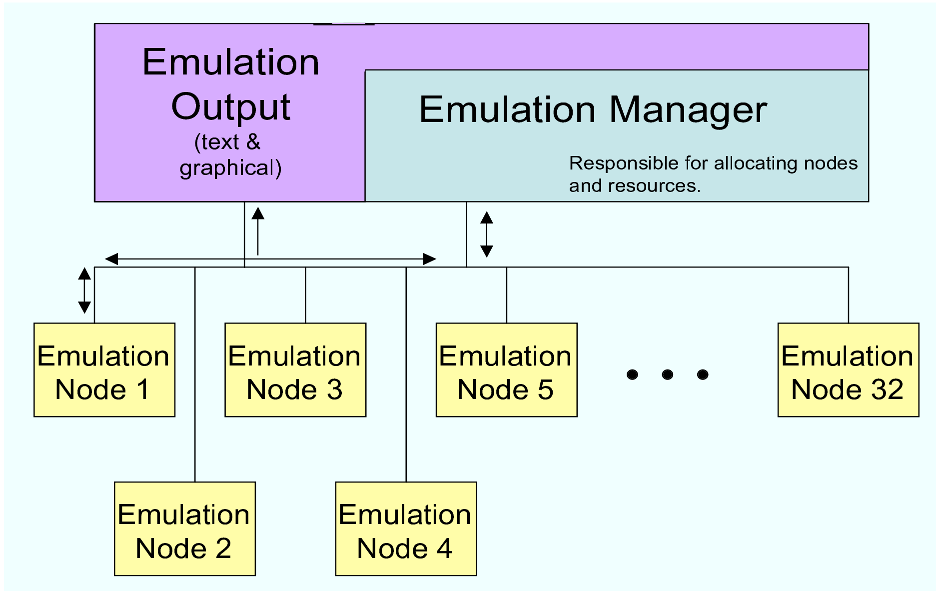

Fig. 1. SCEF Architecture.

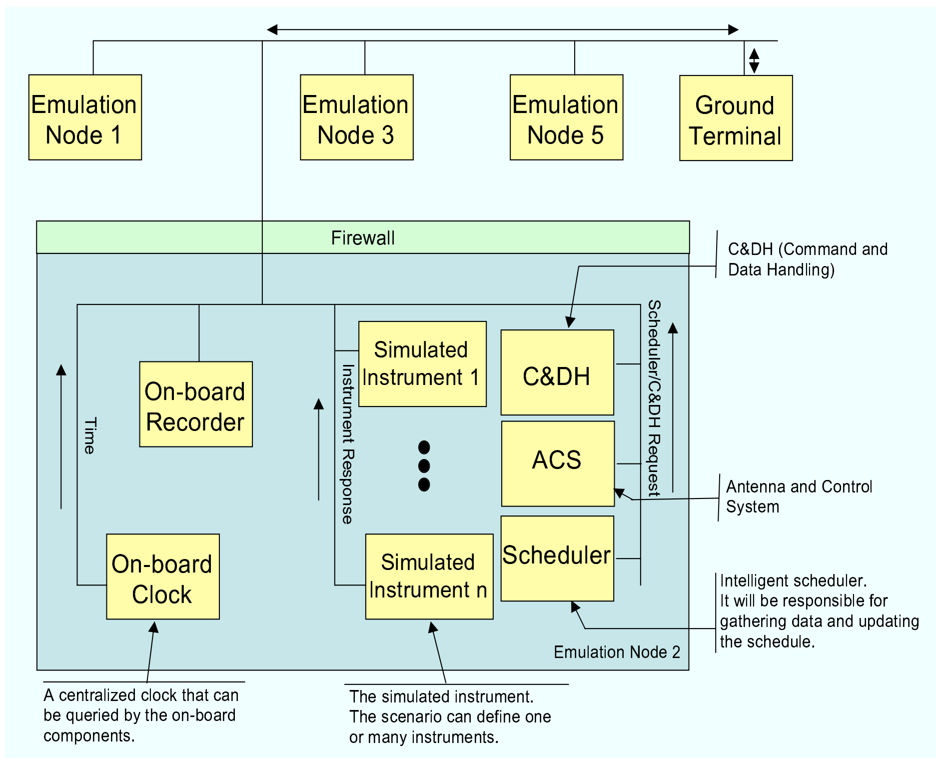

Fig. 2. SCEF Node Structure.

time current by some on-board mechanism or using a ground-based time service. Components in the node software will be the science instruments, C&DH software, on-board scheduler, housekeeping subsystems, and antenna pointing subsystems.

IV. Scenarios for Autonomous On-board Revision

We define a set of scenarios that involves a number of satellites, as well as specifications for each of the satellites. As part of the specifications, a default schedule execution algorithm will be replaced with the on-board schedule revision system just described. Once the scenario is executed, we can evaluate the benefits of the revision system in terms of the improvement in the overall science utility of the acquired images.

Examples of scenarios that will be implemented and demonstrated include:

1) *Nominal execution of ground schedule*. A schedule sequence is uplinked and executed rigorously by the satellite, with the acquired data stored and eventually downlinked as planned. No schedule update or communication involved other than communications with ground stations as pre-scheduled.
2) *Targets of opportunity (TOO)*. Two or more satellites are configured in a train. The lead satellite detects and communicates a TOO (volcano eruption, fire, etc.) to the trailing satellite; the trailing satellite adjusts its schedule accordingly. The trailing satellite might have to remove existing observations in the schedule to accommodate for the high priority target of opportunity. It might also do some local adjustments to the schedule, adding spare observations or interrupting an observation, to keep local optimality in terms of scheduled observations utilities.
3) *Ground station loss of contact*. During a downlink, a satellite unexpectedly loses contact with a ground station, and is unable to dump all its data. It is forced to either revise its schedule of future observations due to depleted SSR capacity, deleting some of the stored images, determine if the data can be dynamically routed to another ground station using one or more relay satellites, or event determine whether the data could be stored on a satellite in close proximity.
4) *Acquisition and utilization of real-time weather data*. A weather satellite monitors changes in weather along the track followed by imaging satellites. This information allows an imaging satellite to revise its schedule based on up-to-date data on expected image quality. In other words, if the lead satellite in a constellation determines that image quality for future images along the tract is poor; this estimate is communicated to trailing satellites which update their schedules accordingly.
5) *Coordination between missions*. A satellite might delete from (add to) its schedule an image that is taken (missed) by another satellite. Also, two satellites might synchronize taking the same image to have the scene taken for different pointing angles, taken by different resolutions, or taken at different time for studying phenomena changes.

V. Integration with the SCEF Emulator

When combining the on-board schedule revision and execution system with the SCEF emulator, we need to consider real-time computational aspects such as decision-making overhead time, communication update delays, and possible execution failures. Towards this end, we introduce the following concepts.

A **Commitment Window** is the interval of time in the near future within which the schedule is not to be changed. During execution, the Commitment Window slides with time to keep its lower bound in line with the current time. The fixed portion of the schedule, which resides in the Commitment Window, allows for handling execution and scheduling delays. (The terminology follows that found in [1].)

Schedule Decision-making Steps

```
While not end-of-horizon
   if (data-update), distinguish type of update:
     1. Weather (e.g., Cloud Coverage):
          update image utilities
     2. Target-of-Opportunity (TOO):
        if within commitment-window
          missed-TOO
        if beyond commitment-window
          place TOO on the schedule
          and plan around it
     3. Downlinking Availability:
        adjust downloading schedule
        and storage content
   adjust-commit-window while
   optimizing its schedule
```

Fig. 3. Observation Scheduling Decision Making in the SCEF framework.

The **Freeze Time** is the minimum amount of time, or smallest interval of time starting from the current time, when the schedule should not be altered.

```
While not end-of-horizon
   for each activity scheduled to start "now"
     initiate the activity
     satisfying all requirements
     and making needed adjustments
   for each activity scheduled to end "now"
     terminate the activity
     satisfying all requirements
     and making needed adjustments
```

Fig. 4. Observation Execution in the SCEF framework.

A Commitment Window size is no less than the Freeze Time. While Freeze Time is constant for a satellite, the Commitment Window changes based on the activity at the end of the window. We assume execution to be non-preemptive and, therefore, the Commitment Window should not end in the middle of a scheduled activity.

A sketch of the on-board decision making algorithm is given in Figure 3. It differs from the algorithm given in [3] in the fact that the objective is to decide which Science observation to schedule for the next commitment window rather than for the next time point. It also takes into consideration dynamic updates (e.g., new observation requests). All other aspects, specifically lookahead strategies, are the same.

The schedule is fixed within the Commitment Window. At every time point, the Commitment Window slides to where its

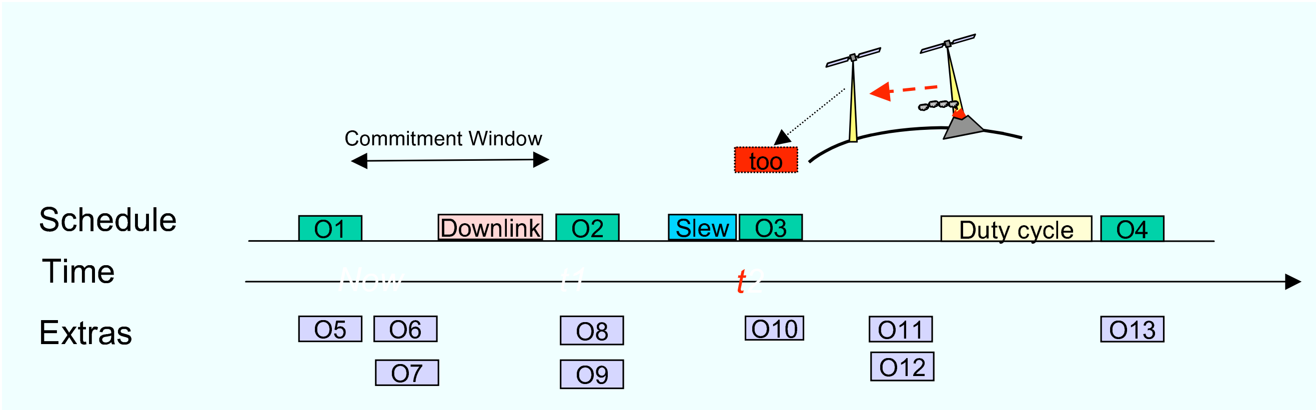

Fig. 5. Example of the On-board Science Observation Schedule and Revision.

lower bound matches the current time and its upper bound is adjusted accordingly[1]. After the adjustment, if there are observations to be added, a greedy selection among competing observations is performed. This is done by applying a lookahead strategy for selecting the best local schedule in terms of observation utility as described in prior work [3]. After the selection, the Commitment Window upper boundary might need to get re-adjusted. If an update is communicated to the scheduler, the update will be handled properly and for the best of the overall utility of revised schedule. When a target of opportunity is detected beyond the commitment window, the scheduler will attempt to fits it in the schedule by removing lower utility observations and their related activities (such as slewing). When weather forecast indicates different cloud coverage than expected, the utility of related images will be adjusted. When loss of contact with ground station is communicated, a downlink activity will be canceled, some scheduled activities might get canceled, and some images stored in the satellite storages could be deleted to free up space for higher utility images.

The schedule executive algorithm is given in Figure 4. The executive keeps track of the next activity to be executed. An activity is either antenna related (downlinking) or instrument related (slewing, taking-image, turning-off, turning-on, and warming-up). For each of the activities, different communication with SCEF is established. If there is an error executing any of the activity, error status will be flagged. The most interesting execution is of the Take-Image activity in the case of indirect downloading. At the start of a Take-Image, the executive requests from the storage manager to reserve the required space. If there is enough available space, the storage manager gives the OK. Otherwise, the storage manager, given it is ok to delete already taken images, will attempt to find stored images of lower utility to delete freeing enough space for the higher utility image. If this can not be done, then the scheduled image will not be executed (execution error). At the end of taking an image, if no execution error so far, the executive will request from the storage manager to finalize the commitment of the reserved space.

[1]The adjustment is based on the status of the last activity in the Commitment Window and the Freeze Time. Details are beyond the scope of this paper

A. Example

To illustrate the entire process, consider the example in Figure 5. A nominal schedule is pictured, along with a set of alternatives. Observations on the same "stack" with respect to the time axis have the same start times. During the set-up phase, the on-board scheduler will boost the utilities for the scheduled activities so that they are biased over the extras. During the execution phase, the current time "now" is within the Commitment Window. In the example scenario, the activity scheduled at the current time, $O1$, is executed. Also, the Commitment Window is about to extend to include time $t1$. For that, the decision making algorithm will assign, using some lookahead strategy, a heuristic value to each observation that can be taken at $t1$. In the figure, this value would be thus assigned to $O2$, $O8$, and $O9$. The observation with highest heuristic value will be selected and committed. Unless there is update in the utilities of some neighboring observations, we expect the selection to be for the nominally scheduled observation $O2$.

To illustrate another aspect of schedule revision, assume a leading satellite had detected a target of opportunity for time $t2$. The schedule will be revised by removing $O3$ from the schedule and adding the target of opportunity which, naturally, will have higher utility value. In addition, the "Slew" activity scheduled for $O3$ will also be removed from the schedule and, possibly, replaced by other pre-requisite activity for the added target of opportunity observation.

B. Current Status

The integration effort is currently at an early stage of development. We are addressing a number of technical challenges, among them the ability to synchronize the on-board science activities scheduler with the executive. One problem is in deciding on the duration of the "Freeze Time", the future time window containing activities which the scheduler is not permitted to alter. Whether the Freeze Time is constant across execution or altered depending on current activities and situation is to be studied and evaluated based on the delay incurred in updating schedules. Other challenges include fitting in a Target Of Opportunity at the expense of pre-preempting an image in execution or deleting an already acquired image. Still other challenges include planning with resources, particularly SSR capacity.

VI. Conclusion

This paper has presented the design of a set of experiments demonstrating the usefulness of autonomous on-board revision of scheduled observations by Earth observing sensors. The paper has described an approach to schedule revision that is compatible with the requirements for fast decision-making with limited computational resources. A cost-effective, yet robust experimental platform is provided by the SCEF emulation facility. Future reports will document the results of the experiments described here.

VII. Acknowledgment

The authors would like to thank Larry McFarland for his review of an earlier version of this paper.

References

[1] Fisher, F., Knight, R., Engelhardt, B., Chien, S., and Alejandre, N., A Planning Approach to Monitor and Control For Deep Space Communications. IEEE Aerospace Conference (IAC00), Big Sky, MT. March 2000.

[2] Khatib, L., Frank, J., Smith, D., Morris, R., and Dungan, J. An Integrated Architecture for Science Observation Scheduling for Fleets of Earth Observing Spacecraft, Proceedings of the Space Mission Challenges and Information Technology (SMC-IT), 2003.

[3] Khatib, L., Frank, J., Smith, D., Morris, R., and Dungan, J. Interleaved Observation Execution and Rescheduling on Earth Observing Systems, Plan Execution Workshop at the 13th International Conference on Automated Planning&Scheduling, Trento, Italy, 2003.

[4] Lansing, F., Lemmerman, L., Walton, A., Bothwell, G., Bhasin, K., and Prescott, G. Needs for Communications and Onboard Processing in the Vision Era, IGARSS Conference, Toronto, Canada, June 2002.

[5] Potter, W. and Gasch, J. A Photo Album of Earth: Scheduling Daily Landsat 7 Activities. In the Proceedings of SpaceOps98, Tokyo, Japan, 1998.

[6] Slywczak, R. Conceptual Design of an IP-based Satellite Bus Using Internet Technologies. 17th Annual AIAA/USU Conference on Small Satellites, Logan, Utah, August, 2003.