

Onboard Autonomy on the Earth Observing One Mission

Steve Chien, Rob Sherwood, Daniel Tran, Benjamin Cichy,
Gregg Rabideau, Rebecca Castano, Ashley Davies
Jet Propulsion Laboratory, California Institute of Technology

Dan Mandl, Stuart Frye, Bruce Trout, Jerry Hengemihle, Jeff D'Agostino,
Seth Shulman, Stephen Ungar, Thomas Brakke
Goddard Space Flight Center

Darrell Boyer, Jim Van Gaasbeck,
Interface & Control Systems

Ronald Greeley, Thomas Doggett,
Arizona State University

Victor Baker, James Dohm, Felipe Ip,
University of Arizona

***Abstract*— The Earth Observing One Spacecraft is currently flying The Autonomous Sciencecraft Experiment (ASE) - onboard autonomy software to improve science return. The ASE software enables the spacecraft to autonomously detect and respond to science events occurring on the Earth. ASE includes software systems that perform science data analysis, mission planning, and run-time robust execution. In this article we describe the autonomy flight software and how it enables a new paradigm of autonomous science and mission operations.**

I. Introduction

In 2004, the ASE running on the EO-1 spacecraft has demonstrated several integrated autonomy technologies to enable autonomous science. Several science algorithms including: onboard event detection, feature detection, change detection, and unusualness detection are being used to analyze science data. These algorithms are used to downlink science data only on change, and detect features of scientific interest such as volcanic eruptions, sand dune migration, growth and retreat of ice caps, cloud detection, and crust deformation. These onboard science algorithms are inputs to onboard decision-making algorithms that modify the spacecraft observation plan to capture high value science events. This new observation plan is then executed by a robust goal and task oriented execution system, able to adjust the plan to succeed despite run-time anomalies and uncertainties. Together these technologies enable autonomous goal-directed exploration and data acquisition to maximize science return. This paper describes the specifics of the ASE and relates it to past and future flights to validate and mature this technology.

The ASE onboard flight software includes several autonomy software components:

- Onboard science algorithms that analyze the image data to detect trigger conditions such as science events, “interesting” features, changes relative to previous observations, and cloud detection for onboard image masking
- Robust execution management software using the Spacecraft Command Language (SCL)¹ package to enable event-driven processing and low-level autonomy
- The Continuous Activity Scheduling Planning Execution and Replanning (CASPER)² software that replans activities, including downlink, based on science observations in the previous orbit cycles

The onboard science algorithms analyze the images to extract static features and detect changes relative to previous observations. This software has already been demonstrated on EO-1 Hyperion data to automatically identify regions of interest including land, ice, snow, water, and thermally hot areas. Repeat imagery using these algorithms can detect regions of change (such as flooding, ice melt, and lava flows). Using these algorithms onboard

enables retargeting and search, e.g., retargeting the instrument on a subsequent orbit cycle to identify and capture the full extent of a flood.

Although the ASE software is running on the Earth observing spacecraft EO-1, the long term goal is to use this software on future interplanetary space missions. On these missions, onboard science analysis will enable capture of short-lived science phenomena. In addition, onboard science analysis will enable data be captured at the finest time-scales without overwhelming onboard memory or downlink capacities by varying the data collection rate on the fly. Examples include: eruption of volcanoes on Io, formation of jets on comets, and phase transitions in ring systems. Generation of derived science products (e.g., boundary descriptions, catalogs) and change-based triggering will also reduce data volumes to a manageable level for extended duration missions that study long-term phenomena such as atmospheric changes at Jupiter and flexing and cracking of the ice crust on Europa.

The onboard planner (CASPER) generates mission operations plans from goals provided by the onboard science analysis module. The model-based planning algorithms enable rapid response to a wide range of operations scenarios based on a deep model of spacecraft constraints, including faster recovery from spacecraft anomalies. The onboard planner accepts as inputs the science and engineering goals and ensure high-level goal-oriented behavior.

The robust execution system (SCL) accepts the CASPER-derived plan as an input and expands the plan into low-level commands. SCL monitors the execution of the plan and has the flexibility and knowledge to perform event driven commanding to enable local improvements in execution as well as local responses to anomalies.

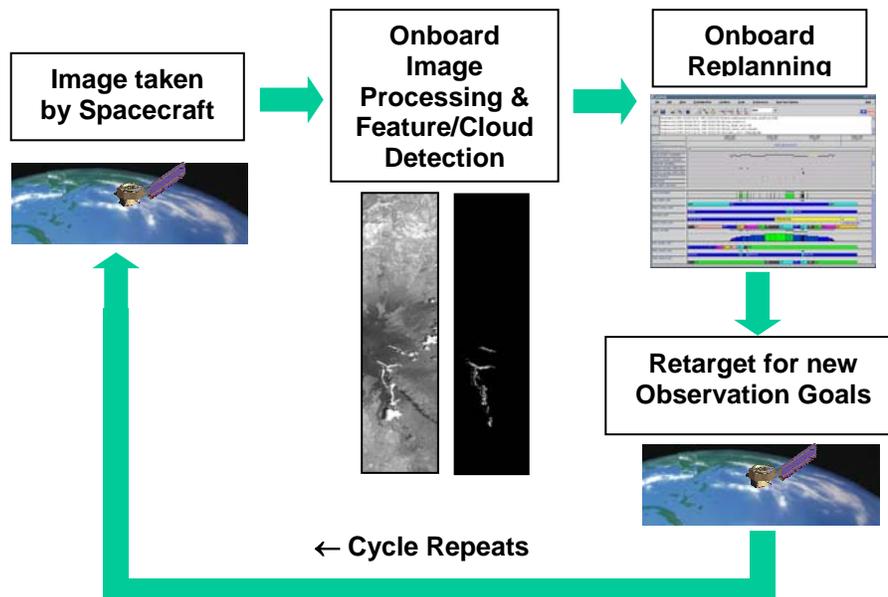


Figure 1. Autonomous Science Mission Concept

A typical ASE demonstration scenario involves monitoring of active volcano regions such as Mt. Etna in Italy. (See Fig. 1.) Hyperion data have been used in ground-based analysis to study this phenomenon. The ASE concept is applied as follows: Initially, ASE has a list of science targets to monitor that have been sent as high-level goals from the ground. As part of normal operations, CASPER generates a plan to monitor the targets on this list by periodically imaging them with the Hyperion instrument. For volcanic studies, the IR and near IR bands are used. During execution of this plan, the EO-1 spacecraft images Mt. Etna with the Hyperion instrument. The onboard science algorithms analyze the image and detect a fresh lava flow. Based on this detection the image is downlinked. Had no new lava flow been detected, the science software would generate a goal for the planner to acquire the next highest priority target in the list of targets. The addition of this goal to the current goal set triggers CASPER to modify the current operations plan to include numerous new activities in order to enable the new science observation. The SCL software executes the CASPER generated plans in conjunction with several autonomy elements. This cycle is then repeated on subsequent observations.

II. The EO-1 Mission

Earth Observing-1 (EO-1) is the first satellite in NASA's New Millennium Program Earth Observing series³. The primary focus of EO-1 is to develop and test a set of advanced technology land imaging instruments. EO-1 was launched on a Delta 7320 from Vandenberg Air Force Base on November 21, 2000. It was inserted into a 705 km circular, sun-synchronous orbit at a 98.7 degrees inclination. This orbit allows for 16-day repeat tracks, with 3 over flights per 16-day cycle with a less than 10-degree change in viewing angle. For each scene, over 20-Gbits of data from the Advanced Land Imager (ALI), Hyperion, and Atmospheric Corrector (AC) are collected and stored on the onboard solid-state data recorder at high rates.

EO-1 is currently in extended mission, having more than achieved its original technology validation goals. As an example, over 5,000 data collection events have been successfully completed, against original success criteria of 1,000 data collection events. The ASE described in this paper uses the Hyperion hyper spectral instrument. The Hyperion is a high-resolution imager capable of resolving 220 spectral bands (from 0.4 to 2.5 μm) with a 30-meter spatial resolution. The instrument images a 7.5 km by 42 km land area per image and provides detailed spectral mapping across all 220 channels with high radiometric accuracy.

The EO-1 spacecraft has two Mongoose M5 processors. The first M5 is used for the EO-1 command and data handling functions. The other M5 is part of the WARP (Wideband Advanced Recorder Processor), a large mass storage device. Each M5 runs at 12 MHz (for ~ 8 MIPS) and has 256 MB RAM. Both M5's run the VxWorks operating system. The ASE software operates on the WARP M5. This provides an added level of safety for the spacecraft since the ASE software does not run on the main spacecraft processor.

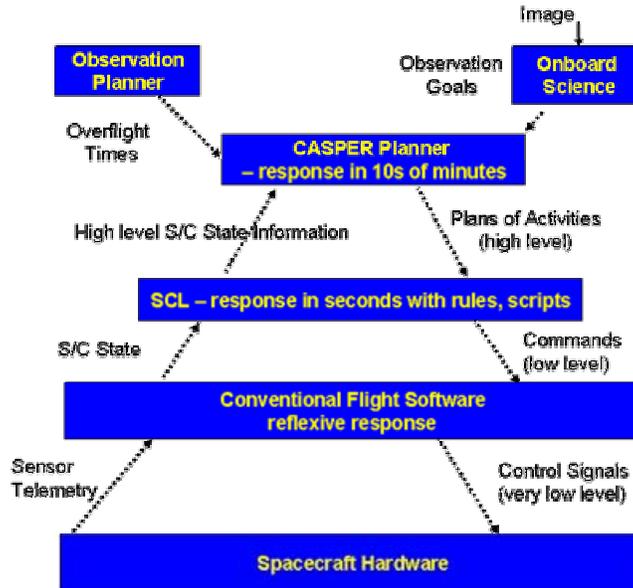


Figure 1. Autonomy Software Architecture

III. Autonomy Software Architecture

The autonomy software on EO-1 is organized into a traditional three-layer architecture⁴ (See Fig. 2.). At the highest level of abstraction, the Continuous Activity Scheduling Planning Execution and Replanning (CASPER) software is responsible for mission planning functions. CASPER schedules science activities while respecting spacecraft operations and resource constraints. The duration of the planning process is on the order of tens of minutes. CASPER scheduled activities are inputs to the Spacecraft Command Language (SCL) system, which generates the detailed sequence commands corresponding to CASPER scheduled activities. SCL operates on the several second timescale. Below SCL, the EO-1 flight software is responsible for lower level control of the spacecraft and also operates a full layer of independent fault protection. The interface from SCL to the EO-1 flight software is at the same level as ground generated command sequences. The science analysis software is scheduled by CASPER and executed by SCL in a batch mode. The results from the science analysis software result in new observation requests presented to the CASPER system for integration in the mission plan.

This layered architecture was chosen for two principal reasons:

1. The layered architecture enables separation of responses based on timescale and most appropriate representation. The flight software level must implement control loops and fault protection and respond very rapidly and is thus directly coded in C. SCL must respond quickly (in seconds) and perform many procedural actions. Hence SCL uses as its core representation scripts, rules, and database records. CASPER must reason about longer term operations, state, and resource constraints. Because of its time latency, it can afford to use a mostly declarative artificial intelligence planner/scheduler representation.
2. The layered architecture enables redundant implementation of critical functions – most notable spacecraft safety constraint checking. In the design of our spacecraft agent model, we implemented spacecraft safety constraints in all levels where feasible.

Each of the software modules operates at a separate VxWorks operating system priority. The tasks are shown below in Table 1 in decreasing priority. The ASE to flight software bridge is the task responsible for reading the real-time flight software telemetry stream, extracting pertinent data, and making it accessible to the remainder of the ASE software. The Band Stripping task reads the science data from the onboard solid state recorder and extracts a small portion of the science data (12 bands of Hyperion data) to RAM. The science analysis software then operates on the extracted data to detect science events.

It is worth noting that our agent architecture is designed to scale to multiple agents. Agents communicate at either the planner level (via goals) or the execution level (to coordinate execution).

Table 1. EO-1 Software Tasks in Decreasing Task Priority (e.g. upper tasks have highest priority for CPU).

Set of Tasks	Rationale for Priority
EO-1 Flight Software	Required for WARP hardware safety
ASE to FSW Bridge	Required to keep up with telemetry stream
Band Stripping	Utilizes WARP hardware while running
SCL	Lowest level autonomy, closes tightest loops
CASPER	Responds in tens of minutes timescale
Science Analysis	Batch process without hard deadlines

We now describe each of the architectural components of our architecture in further detail.

IV. Onboard Science Analysis

The first step in the autonomous science decision cycle is detection of interesting science events. In the complete experiment, a number of science analysis technologies have been flown including:

- Thermal anomaly detection – uses infrared spectra peaks to detect lava flows and other volcanic activity. (See Fig. 3a.)
- Cloud detection⁵ – uses intensities at six different spectra and thresholds to identify likely clouds in scenes. (See Fig. 3b.)
- Flood scene classification – uses ratios at several spectra to identify signatures of water inundation as well as vegetation changes caused by flooding.
- Change detection – uses multiple spectra to identify regions changed from one image to another. This technique is applicable to many science phenomena including lava flows, flooding, freezing and thawing and is used in conjunction with cloud detection. (See Fig. 3c.)
- Generalized Feature detection – uses trainable recognizers to detect such features as sand dunes and wind streaks (to be flown).

Figure 3a shows both the visible and the infrared bands of the same image of the Mt. Etna volcano in Italy. The infrared bands are used to detect hot areas that might represent fresh lava flows within the image. In this picture, these hot spots are circled with red dotted lines. The area of hot pixels can be compared with the count of hot pixels from a previous image of the same area to determine if change has occurred. If there has been change, a new image might be triggered to get a more detailed look at the eruption.

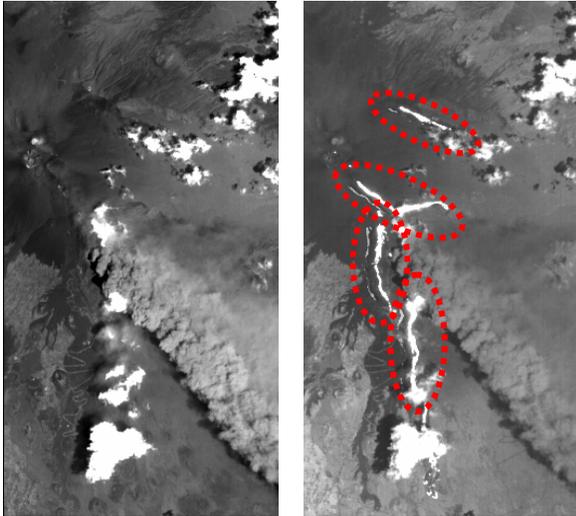


Figure 3a. Thermal Anomalies associated with volcano activity at Mt. Etna, visual spectra at left and infra-red at right.

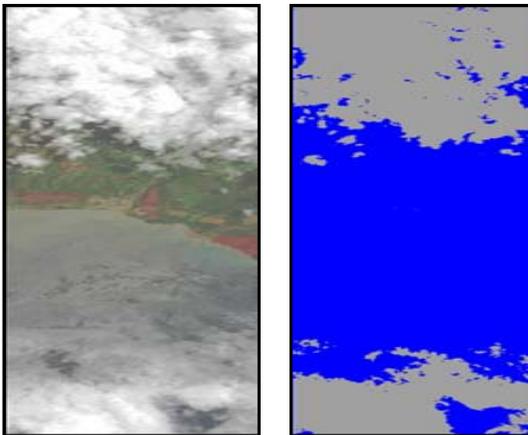


Figure 3b. Cloud Detection of a Hyperion Scene – visual image at left, grey in the image at right indicates detected cloud.

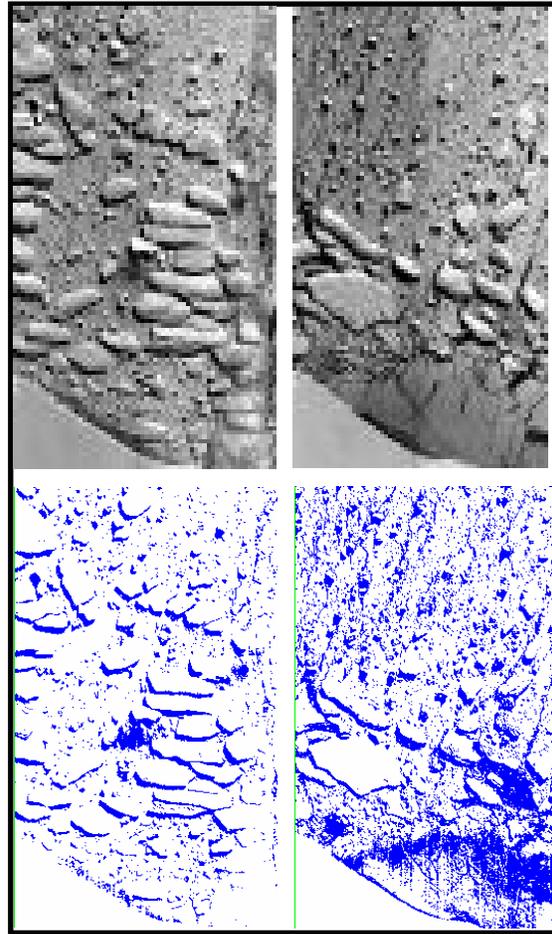


Figure 3c. Change Detection Scenes indicating Ice Breakup in the Larsen Ice Shelf, Antarctica.

Figure 3b shows a Hyperion scene and the results of the cloud detection algorithm. This MIT Lincoln Lab developed algorithm is able to discriminate between cloud pixels and land pixels within an image. Specifically, the grey area in the detection results is clouds while the blue area is land. The results of this algorithm can be used to discard images that are too cloudy.

Figure 3c contains 4 images. The top two are detailed Hyperion images taken of the Larson Ice Shelf in Antarctica on 4/6/2002 and 4/13/2002. A large change in the ice shelf is seen in comparing the images. The bottom 2 images are results of the land-ice-water detection algorithm. The white area of the image is ice and the blue area is water. The ice and water pixels can be counted and compared with the second image to determine if change has occurred. If change is detected, the image can be downlinked and further images of the area can be planned.

The onboard science algorithms are limited to using 12 bands of the hyperion instrument. Of these 12 bands, 6 are dedicated to the cloud detection algorithm. The other six are varied depending on which science algorithm is used. The images used by the algorithm are “Level 0.5,” an intermediate processing level between the raw Level 0, and the fully ground processed Level 1. Each of the science algorithms except the generalized feature detection use simple threshold checks on the spectral bands to classify the pixels.

Initial experiments used the cloud detection triggers. The MIT Lincoln Lab developed cloud detection algorithm uses a combination of spectral bands to discriminate between clouds and surface features. The Hyperion Cloud Cover (HCC) algorithm was run on all images acquired during ASE experiments. In the event of high cloud cover, the image could be discarded and a new goal could be sent to CASPER to reimage the area or image another high priority area. Images with low cloud cover can either be downlinked or analyzed further by other ASE science algorithms.

The JPL developed thermal anomaly algorithms use the infrared spectral bands to detect sites of active volcanism. There are two different algorithms, one for day time images and one for night time images. The algorithms compare the number of thermally active pixels within the image with the count from a previous image to determine if new volcanism is present. If no new volcanism is present, the image can be discarded onboard. Otherwise, the entire image or the interesting section of the image can be downlinked.

The University of Arizona developed flood scene classification algorithm uses multiple spectral bands to differentiate between land and water. The results of the algorithm include are compared with land and water counts from a previous image to determine if flooding has occurred. If significant flooding has been detected, the image can be downlinked. In addition, a new goal can be sent to the CASPER planning software to image adjacent regions on subsequent orbits to determine the extent of the flooding. We have noticed a few problems when ground testing this algorithm with existing hyperion data. The presence of clouds or heavy smoke within an image can cause the algorithm to fail.

The Arizona State University developed Snow-Water-Ice-Land (SWIL) algorithm is used to detect lake freeze/thaw cycles and seasonal sea ice. The SWIL algorithm uses six spectral bands for analysis.

V. Onboard Mission Planning

In order for the spacecraft to respond autonomously to the science event, it must be able to independently perform the mission planning function. This requires software that can model all spacecraft and mission constraints. The CASPER² software performs this function for ASE. CASPER represents the operations constraints in a general modeling language and reasons about these constraints to generate new operations plans that respect spacecraft and mission constraints and resources. CASPER uses a local search approach⁶ to develop operations plans.

Because onboard computing resources are scarce, CASPER must be very efficient in generating plans. While a typical desktop or laptop PC may have 2000-3000 MIPS performance, 5-20 MIPS is more typical onboard a spacecraft. In the case of EO-1, the Mongoose V CPU has approximately 8 MIPS. Of the 3 software packages, CASPER is by far the most computationally intensive. For that reason, our optimization efforts were focused on CASPER. Since the software was already written and we didn't have funding to make major changes in the software, we had to focus on developing an EO-1 CASPER model that didn't require a lot of planning iterations. For that reason, the model has only a handful of resources to reason about. This ensures that CASPER is able to build a plan in tens of minutes on the relatively slow CPU.

CASPER is responsible for mission planning in response to both science goals derived onboard as well as anomalies. In this role, CASPER must plan and schedule activities to achieve science and engineering goals while respecting resource and other spacecraft operations constraints. For example, when acquiring an initial image, a volcanic event is detected. This event may warrant a high priority request for a subsequent image of the target to study the evolving phenomena. In this case, CASPER modifies the operations plan to include the necessary activities to re-image. This may include determining the next over flight opportunity, ensuring that the spacecraft is pointed appropriately, that sufficient power, and data storage are available, that appropriate calibration images are acquired, and that the instrument is properly prepared for the data acquisition.

VI. Onboard Robust Execution

ASE uses the Spacecraft Command Language (SCL)¹ to provide robust execution. SCL is a software package that integrates procedural programming with a real-time, forward-chaining, rule-based system. A publish/subscribe software bus allows the distribution of notification and request messages to integrate SCL with other onboard software. This design enables both loose or tight coupling between SCL and other flight software as appropriate.

The SCL "smart" executive supports the command and control function. Users can define scripts in an English-like manner. Compiled on the ground, those scripts can be dynamically loaded onboard and executed at an absolute or relative time. Ground-based absolute time script scheduling is equivalent to the traditional procedural approach to spacecraft operations based on time. In the EO-1 experiment, SCL scripts are planned and scheduled by the CASPER onboard planner. The science analysis algorithms and SCL work in a cooperative manner to generate new goals for CASPER. These goals are sent as messages on the software bus.

Many aspects of autonomy are implemented in SCL. For example, SCL implements many constraint checks that are redundant with those in the EO-1 fault protection software. Before SCL sends each command to the EO-1 command processor, it undergoes a series of constraint checks to ensure that it is a valid command. Any prerequisite states required by the command are checked (such as the communications system being in the correct mode to accept a command). SCL also verifies that there is sufficient power so that the command does not trigger a low bus voltage condition and that there is sufficient energy in the battery. Using SCL to check these constraints (while included in the CASPER model) provides an additional level of safety to the autonomy flight software.

VII. Flight Status

The ASE software was integrated under the flight version of VxWorks in December 2002, and has been undergoing testing and integration with the WARP flight software. We are testing the individual software components to gain confidence before we perform an integrated flight test.

The cloud detection algorithms were tested onboard in March 2003. The SCL software was tested onboard in May 2003. This test involved starting up the SCL software, testing the software bridge between the SCL software bus and WARP software bus, testing the SCL message and telemetry logs, testing sending commands, and testing sending and executing commands that performed a dark calibration of the Hyperion instrument.

In July 2003, a ground version of CASPER generated several plans that were subsequently uplinked and executed onboard. These plans included image data takes, maneuvers, and telecommunication passes. The purpose of this test was to prove that CASPER could generate valid plans that could be executed by the satellite.

In August 2003, onboard decompression was tested. This capability is used to compress the software before uplink because the uplink rate is only 2 Kb/s. Without compression it would take more than a week to upload the entire ASE software. This test involved uplinking several compressed files, decompressing them onboard, and then downlinking them. The files were then checked for errors.

The ASE software has been flying onboard the EO-1 spacecraft since January 2004. In January and February 2004, we tested several autonomous instrument data take experiments using CASPER/SCL. This test involved uplinking a high level goal that includes a target location and a few instrument mode parameters. We have steadily increased the level of autonomy since this period. In April 2004, we started the first closed-loop execution where ASE autonomously analyzes science data onboard and triggers subsequent observations. So far, we have run over 20 of these trigger experiments with over 100 autonomously planned image data takes. Our most recent focus has been to expand the duration of the tests until ASE is controlling the satellite for 7 days straight. This will involve over 100 image data takes and over 50 ground contacts.

VIII. Impact on Operations

ASE can impact several aspects of spacecraft operations. The mission planning process is simplified because the operations team no longer has to build detailed sequences of commands. The spacecraft can be commanded using high-level goals, which are then detailed by the planner onboard. The processes of planning, build sequence, upload sequence, execute sequence, downlink data, analyze data, and build new sequence are entirely automated using ASE. For example, in the current EO-1 operations, a significant percentage of the images downlinked are of no value because they are mostly covered in clouds. Using ASE, these images can now be discarded onboard and the satellite can acquire another image of a different area. This saves time and labor for the mission planning team, science analysis team, ground station team, flight operations team, and data processing and archive team.

Due to computing limitations, the ASE architecture for EO-1 does not include an autonomous fault protection component. Although this wasn't included for EO-1, it's a natural fit for the ASE onboard autonomy software. In one example, CASPER generates a mission level plan that includes a sequence of behavior goals, such as producing thrust. The SCL executive is responsible for reducing these goals to a control sequence, for example, opening the relevant set of valves leading to a main engine. A device, such as a valve, is commanded indirectly; hence, SCL must ensure that the components along the control path to the device are healthy and operating before commanding that device. Components may be faulty, and redundant options for achieving a goal may exist; hence, SCL must ascertain the health state of components, determine repair options when viable, and select a course of action among the space of redundant options. Adding this level of fault protection autonomy to a future mission could in theory, eliminate the spacecraft analysis team. The team would no longer be required to monitor the spacecraft health because that would be done onboard using model-based mode estimation and mode reconfiguration. The team would also not be required to respond to "safe-hold" periods because anomalies would be handled and reconfigured onboard. Using this software requires a greater up front investment in building the spacecraft models, but much of the underlying software has already been developed in research efforts.

Using the onboard science analysis software can also save time and labor for the science team. The feature detection algorithms can identify specific features of interest within the images. The spacecraft can then downlink the entire image when features are detected, only the detected features, or even a summary of the detected features. Scientists no longer have to analyze many different images to find a feature of interest. In fact, images that do not contain features of interest do not even have to be downlinked. These algorithms can be particularly useful on bandwidth-limited missions by returning the most important science data.

IX. Conclusion

ASE on EO-1 demonstrates an integrated autonomous mission using onboard science analysis, replanning, and robust execution. The ASE performs intelligent science data selection that leads to a reduction in data downlink. In addition, the ASE increases science return through autonomous retargeting. Demonstration of these capabilities onboard EO-1 will enable radically different missions with significant onboard decision-making leading to novel science opportunities. The paradigm shift toward highly autonomous spacecraft will enable future NASA missions to achieve significantly greater science returns with reduced risk and reduced operations cost.

Acknowledgments

Portions of this work were performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. We would like to acknowledge the important contributions of Nghia Tang and Michael Burl of JPL, Dan Mandl, Stuart Frye, Seth Shulman, and Stephen Ungar of GSFC, Jerry Hengemihle and Bruce Trout of Microtel LLC, Jeff D'Agostino of the Hammers Corp., Robert Bote of Honeywell Corp., Jim Van Gaasbeck and Darrell Boyer of ICS, Michael Griffin and Hsiao-hua Burke of MIT Lincoln Labs, Ronald Greeley, Thomas Doggett, and Kevin Williams of ASU, and Victor Baker and James Dohm of University of Arizona.

References

-
- ¹ Interface and Control Systems, SCL Home Page, URL: <http://sclrules.com> [cited 15 September 2004].
 - ² Chien, S., et al., "Using Iterative Repair to Improve Responsiveness of Planning and Scheduling," Proceedings of the Fifth International Conference on Artificial Intelligence Planning and Scheduling, Breckenridge, CO, April 2000.
 - ³ Goddard Space Flight Center, EO-1 Mission page: URL: <http://EO-1.gsfc.nasa.gov> [cited 15 September 2004].
 - ⁴ Gat, E., et al., Three-Layer Architectures. in D. Kortenkamp et al. eds. AI and Mobile Robots. AAAI Press, 1998.
 - ⁵ Griffin, M., Burke, H., Mandl, D., and Miller, J., "Cloud Cover Detection Algorithm for the EO-1 Hyperion Imagery," Proceedings of the 17th SPIE AeroSense 2003, Orlando, FL, April 2003.
 - ⁶ Rabideau, G., Knight, R., Chien, S., Fukunaga, A., Govindjee, A., "Iterative Repair Planning for Spacecraft Operations in the ASPEN System," International Symposium on Artificial Intelligence Robotics and Automation in Space, Noordwijk, The Netherlands, June 1999.