The EO-1 Autonomous Science Agent

Steve Chien, Rob Sherwood, Daniel Tran, Benjamin Cichy,
Gregg Rabideau, Rebecca Castano, Ashley Davies, Rachel Lee
Jet Propulsion Laboratory, California Institute of Technology
Dan Mandl, Stuart Frye[1], Bruce Trout[2] , Jerry Hengemihle[2], Jeff D'Agostino[3],
Seth Shulman[4], 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
Contact: steve.chien@jpl.nasa.gov

`Abstract—` An Autonomous Science Agent is currently flying onboard the Earth Observing One Spacecraft. This software enables the spacecraft to autonomously detect and respond to science events occurring on the Earth. The package includes software systems that perform science data analysis, deliberative planning, and run-time robust execution. Because of the deployment to a remote spacecraft, this Autonomous Science Agent has stringent constraints of autonomy, reliability, and limited computing resources. We describe the constraints and how they were addressed in our agent design, validation, and deployment

1. INTRODUCTION

The Autonomous Sciencecraft Experiment (ASE) is currently flying autonomous agent software on the Earth Observing One (EO-1) spacecraft. This software demonstrates several integrated autonomy technologies to enable autonomous science. Several science algorithms including: onboard event detection, feature detection, and change detection, are used to analyze science data onboard. These algorithms will be used to downlink science data only on change, and will detect features of scientific interest such as volcanic eruptions, flooding, ice breakup, and presence of cloud cover. These onboard science algorithms are inputs to onboard decision-making algorithms that then modifies the spacecraft observation plan to capture high value science events. This new observation plan is then be 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 Autonomous Sciencecraft Experiment (ASE) effort to develop and deploy the Autonomous Science Agent on the Earth Observing One spacecraft.

The ASE onboard flight software includes several autonomy software components:

- Onboard science algorithms that will 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) [10] package to enable event-driven processing and low-level autonomy
- The Continuous Activity Scheduling Planning Execution and Replanning (CASPER) [5] software that will replan activities, including downlink, based on science observations in the previous orbit cycles

The onboard science algorithms will analyze the images to extract static features and detect changes relative to previous observations. Prototype 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 and ice melt) as well as regions of activity (such as lava flows). Using these algorithms onboard will enable retargeting and search, e.g., retargeting the instrument on a subsequent orbit cycle to identify and capture the full extent of a flood. On future interplanetary space missions, onboard science analysis will enable capture of short-lived science phenomena. These can 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 and resurfacing on Europa.

1- MitreTek, 2- Microtel LLC, 3 – the Hammers Company, 4- Honeywell.

AAMAS'04, July 19-23, 2004, New York, New York, USA.

The onboard planner (CASPER) will generate mission operations plans from goals provided by the onboard science analysis module. The model-based planning algorithms will 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 will accept 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.

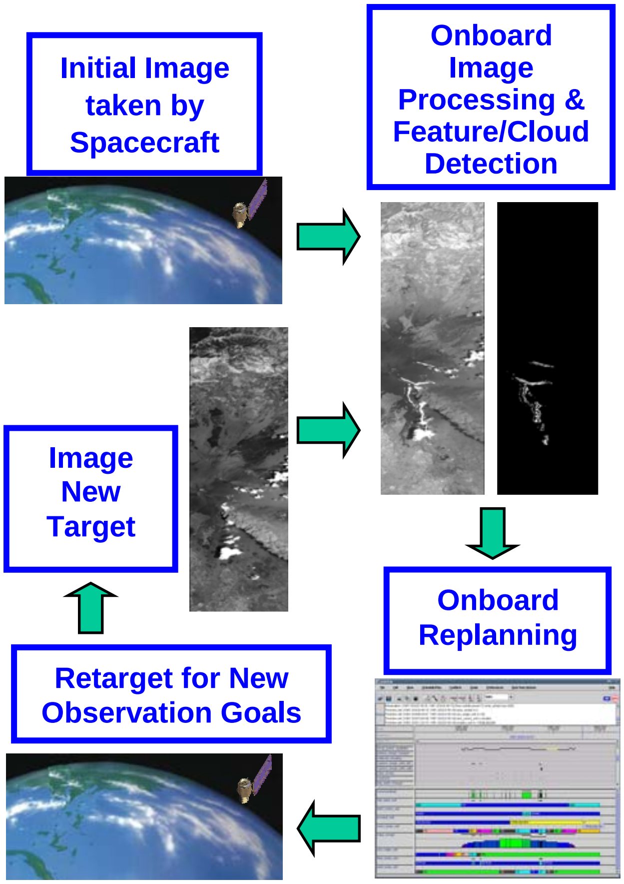


Figure 1. Autonomous Science Scenario

A typical ASE scenario involves monitoring of active volcano regions such as Mt. Etna in Italy. (See Figure 1.) Hyperion data have been used in ground-based analysis to study this phenomenon. The ASE concept will be applied as follows:

1. Initially, ASE has a list of science targets to monitor that have been sent as high-level goals from the ground.
2. 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 infra-red and near infra-red bands are used.
3. During execution of this plan, the EO-1 spacecraft images Mt. Etna with the Hyperion instrument.
4. The onboard science algorithms analyze the image and detect a fresh lava flow, or active vent. If new activity is detected, a science goal is generated to continue monitoring the volcanic site. If no activity is observed, the image is not downlinked.
5. Assuming a new goal is generated, CASPER plans to acquire a further image of the ongoing volcanic activity.
6. The SCL software executes the CASPER generated plan to re-image the site.
7. This cycle is then repeated on subsequent observations.

The basic software architecture used by ASE on EO-1 has been described in several prior papers [3,18], thus in this paper we concentrate on how the software was modified to deal with the unique challenges of flight on EO-1 (most of which apply to other space missions).

Building autonomy software for space missions has a number of key challenges, including the following:

1. Limited, intermittent communications to the agent. A typical spacecraft in low earth orbit (such as EO-1) has 8 10-minute communications opportunities per day. This means that the spacecraft must be able to operate for long periods of time without supervision. For deep space missions the spacecraft may be in communications far less frequently. Some deep space missions only contact the spacecraft once per week, or even once every several weeks.

2. Spacecraft are very complex. A typical spacecraft has thousands of components, each of which must be carefully engineered to survive rigors of space (extreme temperature, radiation, physical stresses). Add to this the fact that many components are one-of-a-kind and thus have behaviors that are hard to characterize.

3. Limited observability. Because processing telemetry is expensive, onboard storage is limited, and downlink bandwidth is limited, engineering telemetry is limited. Thus onboard software must be able to make decisions on limited information and ground operations teams must be able to operate the spacecraft with even more limited information.

4. Limited computing power. Because of limited power onboard, spacecraft computing resources are usually very constrained. An average spacecraft CPUs offer 25 MIPS and 128 MB RAM – far less than a typical personal computer. Our CPU allocation for the Autonomous Science agent on EO-1 is 4 MIPS and 128MB RAM.

5. High stakes. A typical space mission costs hundreds of millions of dollars, any failure has significant economic

AAMAS'04, July 19-23, 2004, New York, New York, USA.

impact. The total EO-1 Mission cost is over $100 million dollars. Over financial cost, many launch and/or mission opportunities are limited by planetary geometries. In these cases, if a space mission is lost it may be years before another similar mission can be launched. Additionally, a space mission can take years to plan, construct the spacecraft, and reach their targets. This delay can be catastrophic.

N the remainder of this paper we first provide background information:

1. describe the basic characteristics of the EO-1 mission and spacecraft
2. review the basic ASE on EO-1 software architecture
3. describe how we updated the science event detection algorithms from the earlier Techsat-21 version of ASE for the EO-1 science instruments

Then in the remainder of the paper we provide information on how our software dealt with three key aspects of software agents for spacecraft namely:

1. We describe how our onboard planning software can generate mission plans despite the limited EO-1 CPU processor (our allocation is about 4 MIPS)
2. We describe how the ASE telemetry was designed to provide sufficient information to track the ASE software performance within very limited bandwidth
3. We describe our layered, redundant agent and how that enables additional agent safety – critical to the operations of mission with cost over $100 Million dollars.

2. THE EO-1 MISSION

Earth Observing-1 (EO-1) is the first satellite in NASA's New Millennium Program Earth Observing series. 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.

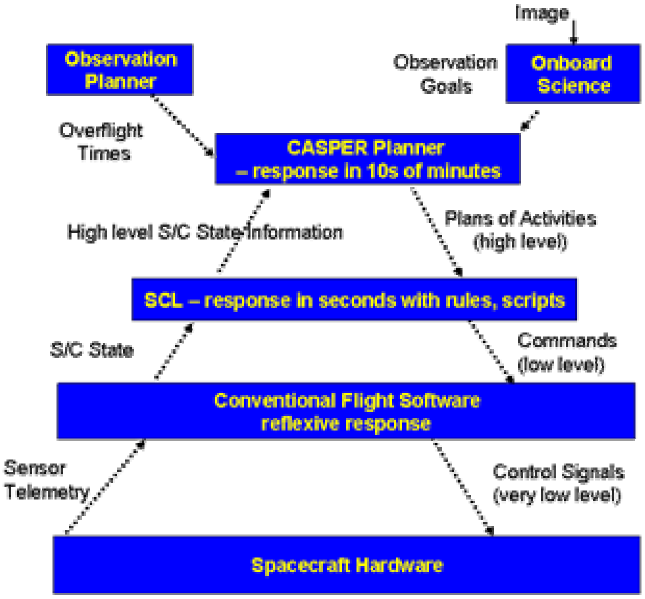


Figure 2. Autonomy Software Architecture

3. AUTONOMY SOFTWARE ARCHITECTURE

The autonomy software on EO-1 is organized into a traditional three-layer architecture (See Figure 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

AAMAS'04, July 19-23, 2004, New York, New York, USA.

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 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 (in seconds) quickly 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.

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

4. 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 will be flown including:

- Thermal anomaly detection – uses infrared spectra peaks to detect lava flows and other volcanic activity. (See Figure 3.)
- Cloud detection [17] – uses intensities at six different spectra and thresholds to identify likely clouds in scenes.
- Flood scene classification – uses ratios at several spectra to identify signatures of water inundation as well as vegetation changes caused by flooding. (see Figure 4.)
- 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.
- Generalized Feature detection – uses trainable recognizers to detect spatial features as sand dunes and wind streaks.

All of these science algorithms use the Hyperion instrument as the ALI data is not available for processing onboard.

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.

Figure 3 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.

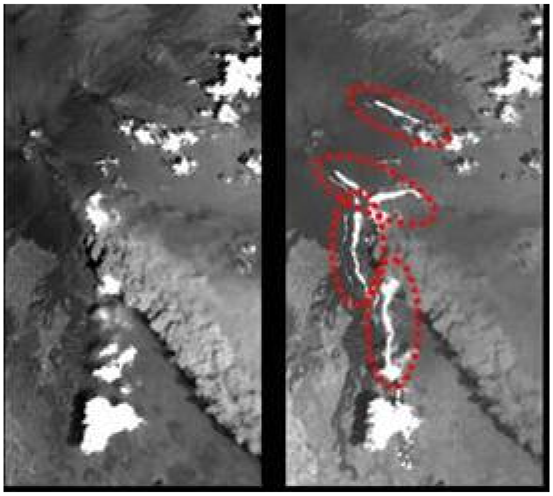


Figure 3. Thermal Anomalies associated with volcano activity at Mt. Etna, visual spectra at left and infra-red spectra with labeled lava flows at right.

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 baseline image to determine if flooding has occurred (or is receding). 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.

AAMAS'04, July 19-23, 2004, New York, New York, USA.

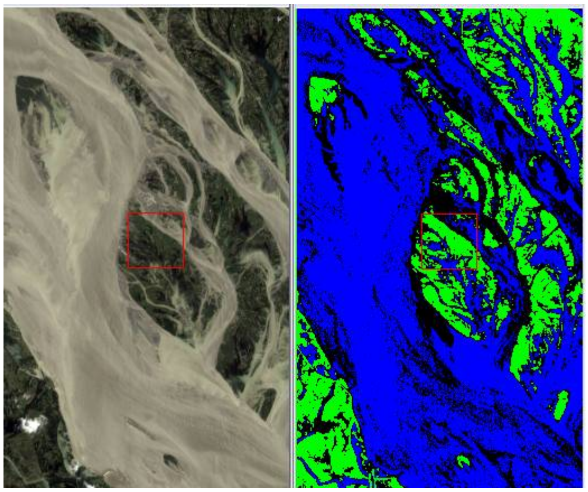


Figure 4. Flood detection with visual spectra at left and flood detection map at right.

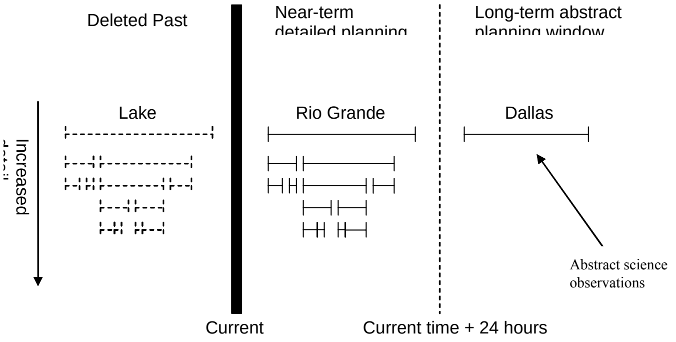
Later flights will validate as many science analysis algorithms as resources allow. These flights will begin by validating change detection on multiple science phenomena, spatial feature detection on Aeolian (wind) features such as sand dunes, sand shapes, and wind streaks, and the Discovery algorithm. Validating this portfolio of science algorithms will represent a valuable step forward to enabling future autonomous science missions [6].

5. CONTINUOUS 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 CASPER [5] 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 approach [15] 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 which only about 4 MIPS are available to the ASE software.

CASPER plans within limited CPU resources by using a hierarchical, continuous [4] planning paradigm. Rather than attempt to plan out an entire week of operations in a single batch timeslice, it utilizes a long-term, more abstract plan for the longest planning horizon (one week), and plans at a detailed level for the next day of operations. As time proceeds forward, it incrementally replans for the new observations that fall within this one-day horizon (see Figure 5). Consequently, CASPER CPU usage is spread more evenly than in a batch planning paradigm.



6. TELEMETRY MANAGEMENT

The ASE software can only send limited data to the ground to enable the operations team to track ASE operations. First, real-time telemetry is logged by the EO-1 flight software every 1-8 seconds depending on spacecraft operating mode and the subsystem. Unfortunately this telemetry is available in real-time only when the spacecraft is in direct contact with the operations center via ground station of TDRSS satellite relay. Because of limited bandwidth ASE is allocated several packets, one for each ASE subsystem (CASPER, SCL, Science, Band Stripping, and Bridge). These packets range in size from 40-350 bytes each and are logged infrequently on change, every 30 seconds, or every minute. The second form of telemetry is archived real-time telemetry. This is the real-time telemetry for the periods in between the ground contacts (e.g. the same packets but stored onboard during times when there was no ground contact). Because of logistical constraints, this data is only available 24 hours after the ground contact (in emergencies it might be accessible within 4 hours). The third form of telemetry is log-files that are stored explicitly by ASE in onboard memory. However log-files require manual actions by operators to downlink so are not viable except to diagnose a spacecraft anomaly.

Because archived real-time telemetry is delayed, the real-time telemetry must provide a good picture of the operations of the ASE software since the last downlink. For example, the real-time telemetry for the CASPER packet is 248 bytes and contains three parts: *Health and Status* provides an indication that the CASPER software is running correctly by heartbeat, counting warnings and errors, and stack and heap usage. *Decisions* provides information on the last repair iterations taken by CASPER to finalize the plan and the number of conflicts before and after the iteration. *Inputs* provides information on the last state and/or resource changes as these changes would cause CASPER to invoke repair. For further details on the ASE telemetry see [19].

AAMAS'04, July 19-23, 2004, New York, New York, USA.

7. AGENT SAFETY REQUIREMENTS

Because of significant concerns for spacecraft health, ASE implements a layered redundant approach to enforcing spacecraft safety. This means that whenever possible *at every level* of the agent architecture, redundant checks are implemented to enhance spacecraft safety. Each of these safeguards has been reviewed by EO-1 spacecraft engineers, EO-1 operations personnel, as well as ASE team members (for a more detailed description of the model development, validation, and testing process, see [17]). In addition, automated code generation techniques were used to develop SCL state & resource constraint checks directly from the CASPER model.

Table 1 below shows analysis of two spacecraft safety constraints. As shown, the operations team, the CASPER planner (via its model), SCL (via scripts and rules), and the EO-1 flight software (FSS) all implement constraints to protect the spacecraft from damage due to faulty commands or anomalies. In this manner, even if one of the layers malfunctions, the spacecraft may still be protected.

Table 1. Sample safety analysis for two risks.

	Instruments overheat from being left on too long	**Instruments exposed to sun**
Operations	For each turn on command, look for the following turn off command. Verify that they are within the maximum separation.	Verify orientation of spacecraft during periods when instrument covers are open.
CASPER	High-level activity decomposes into turn on and turn off activities that are with the maximum separation.	Maneuvers must be planned at times when the covers are closed (otherwise, instruments are pointing at the earth)
SCL	Rules monitor the “on” time and issue a turn off command if left on too long.	Constraints prevent maneuver scripts from executing if covers are open.
FSS	Fault protection software will shut down the instrument if left on too long.	Fault protection will safe the spacecraft if covers are open and pointing near the sun.

Because of the high stakes of EO-1 operations, significant effort also went into validating that the implemented ASE software enforced all of the designed constraints. The testing plan includes a number of cases to verify each constraint is enforced, as well the following general classes of test cases:

1. Coverage test cases that attempt to exercise a representative sample of all possible parameter-value assignments.
2. Stochastic test cases that verify nominal-operation scenarios.
3. Environmental test cases that evaluate how our agent performs in an uncertain environment.

Each build of the ASE software must be rigorously tested in a range of testbeds of increasing fidelity before flight (see Table 2.). The Solaris and Linux testbeds can be run at faster than real-time, however the GESPAC and EO-1 testbeds operate only at real-time.

For each build of the software, it must pass a pre-specified number of runs in order to be accepted for the next level of testbed. This begins with unit testing on workstations and culminates with integrated system runs on the EO-1 testbed prior to flight. These test are quite time consuming. Typically a build requires 100 systems level tests on workstations. Each of these tests may represent hours to a week of operations time and several hours of CPU time. In order to investigate all anomalies from test runs and update software, it may take several hundred runs. Thus for each build the testing scheduled time is measured in weeks or months.

Table 2. Testbeds available to validate EO-1 agent.

Type	Number	Fidelity
Solaris Sparc Ultra	5	Low – can test model but not timing
Linux 2.5 GHz	7	″
GESPAC PowerPC 100-450 MHz	10	Moderate – runs flight OS
EO-1 Flight Testbed Mongoose M5, 12 MHz	3	High – runs Flight Software

7. FLIGHT STATUS

The ASE software has been steadily progressing to full operations with the major milestones listed below.

Test Description	Test Date
Onboard cloud detection	March 2003
Onboard commanding path	May 2003
CASPER ground generated commands executed onboard	July 2003
Software jumping and loading	August 2003
ASE autonomously acquires calibration image and performs downlink	October 2003
ASE autonomously acquires science images and performs downlinks	Jan-Feb 2004 -
ASE autonomously analyzes science	April 2004 -

data onboard and triggers subsequent observations	

The only step remaining for full operations is the flight of the integrated science with autonomous planning and execution. This software is currently in integration and test and is expected to be ready for flight in the April 2004 timeframe. When this software build is ready it will be flown until September 2004 and will be used to acquire as many science –triggered scenes as resources allow.

An additional effort includes teaming with the NASA Ames Research Center to fly the Livingstone 2 Mode Identification and Diagnosis software [16] to be added to ASE in the June 2004 timeframe. The Livingstone 2 experiment would demonstrate tracking of multiple fault hypotheses, a capability not demonstrated in the Remote Agent Experiment in 1999. This effort is in earlier stages but is making good progress.

8. CONTRIBUTION TO FUTURE SPACE MISSIONS

The ASE enables demonstration of onboard science in an Earth-directed mission, but has direct relevance to a large number of deep space missions throughout the solar system. Specifically, the ASE onboard science processing has numerous applications to Space Science Missions. For example, in Europa orbiter and lander missions, onboard science processing could be used to autonomously:

- Monitor surface change as function of changing tidal stress field
- Monitor areas of greatest tidal stresses
- Search for surface change, that is, evidence of recent activity
- Search for landing sites that have a high probability of lander survivability and where the crust is thin enough for deployment of a sub-crust submarine explorer

The ASE Team has identified the NASA Mars Program as an ideal candidate for technology infusion of the ASE software. As a result, we have been working closely with the Mars Odyssey Project to identify and ground test science analysis algorithms that could be used for discovery of high-value science on Mars. The goal of this work is to have an existing or future Mars mission infuse the ASE software into their baseline flight software.

9. RELATED WORK & SUMMARY

In 1999, the Remote Agent experiment (RAX) [13] executed for a few days onboard the NASA Deep Space One mission. RAX is an example of a classic three-tiered architecture [8], as is ASE. RAX demonstrated a batch onboard planning capability (as opposed to CASPER's continuous planning) and RAX did not demonstrate onboard science. PROBA [14] is a European Space Agency (ESA) mission that usesing onboard autonomy and launched in 2001. However, ASE has more of a focus on model-based autonomy than PROBA.

The Three Corner Sat (3CS) University Nanosat mission will be using the CASPER onboard planning software integrated with the SCL ground and flight execution software [3]. 3CS has been delayed several times because it is a shuttle launch and currently scheduled for launch in July 2004. The 3CS autonomy software includes onboard science data validation, replanning, robust execution, and multiple model-based anomaly detection. The 3CS mission is considerably less complex than EO-1 but still represents an important step in the integration and flight of onboard autonomy software.

More recent work from NASA Ames Research Center is focused on building the IDEA planning and execution architecture [12]. In IDEA, the planner and execution software are combined into a "reactive planner" and operate using the same domain model. A single planning and execution model can simplify validation, which is a difficult problem for autonomous systems. For EO-1, the CASPER planner and SCL executive use separate models. While this has the advantage of the flexibility of both procedural and declarative representations, a single model would be easier to validate. We have designed the CASPER modeling language to be used by domain experts, thus not requiring planning experts. Our use of SCL is similar to the "plan runner" in IDEA but SCL encodes more intelligence. The EO-1 science analysis software is defined as one of the "controlling systems" in IDEA. In the IDEA architecture, a communications wrapper is used to send messages between the agents, similar to the software bus in EO-1. In the description of IDEA there is no information about the deployment of IDEA to any domains, so a comparison of the performance or capabilities is not possible at this time. In many ways IDEA represents a more AI-centric architecture with declarative modeling at its core and ASE represents more of an evolutionary engineered solution.

ASE was originally scheduled for flight on the Techsat-21 mission [18]. However this mission was cancelled and the software was adapted for flight on EO-1. The principal changes from the Techsat-21 to EO-1 are that the science payload was changed from a synthetic aperture radar (SAR) to a hyperspectral imaging device (Hyperion). This change requires significant alteration to the science targets and analysis algorithms. The basic software architecture and components (e.g. CASPER and SCL) have remained the same. This paper also reports on some of our experiences in getting the software to flight and operations.

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 will lead to a reduction in data downlink. In addition, the ASE will increase science return through autonomous retargeting. Demonstration of these capabilities onboard EO-1 will enable radically different

AAMAS'04, July 19-23, 2004, New York, New York, USA.

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.

REFERENCES

1. M.C. Burl, L. Asker, P. Smyth, U. Fayyad, P. Perona, J. Aubele, and L. Crumpler, “Learning to Recognize Volcanoes on Venus,” Machine Learning Journal, April 1998.
2. S. Chien, B. Engelhardt, R. Knight, G. Rabideau, R. Sherwood, E. Hansen, A. Ortiviz, C. Wilklow, S. Wichman , "Onboard Autonomy on the Three Corner Sat Mission," Proc i-SAIRAS 2001, Montreal, Canada, June 2001.
3. S. Chien, et al., “The Techsat-21 Autonomous Space Science Agent,” International Conference on Autonomous Agents and Multi-agent Systems (AAMAS 2002). Bologna, Italy. July 2002
4. S. Chien, R. Knight, A. Stechert, R. Sherwood, and G. Rabideau, "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. (also casper.jpl.nasa.gov)
5. A.G. Davies, R. Greeley, K. Williams, V. Baker, J. Dohm, M. Burl, E. Mjolsness, R. Castano, T. Stough, J. Roden, S. Chien, R. Sherwood, "ASC Science Report," August 2001. (downloadable from ase.jpl.nasa.gov)
6. Davies, A. G., E.D. Mjolsness, A.G. Gray, T.F. Mann, R. Castano, T.A. Estlin and R.S. Saunders (1999) Hypothesis-driven active data analysis of geological phenomena using semi-autonomous rovers: exploring simulations of Martian hydrothermal deposits. EOS, Trans. Amer. Geophys. Union, 80, no. 17, S210.
7. E. Gat et al., Three-Layer Architectures. in D. Kortenkamp et al. eds. AI and Mobile Robots. AAAI Press, 1998.
8. Goddard Space Flight Center, EO-1 Mission page: http://EO-1.gsfc.nasa.gov
9. M. Griffin, H. Burke, D. Mandl, & J. Miller, “Cloud Cover Detection Algorithm for the EO-1 Hyperion Imagery,” Proceedings of the 17th SPIE AeroSense 2003, Orlando, FL, April 21-25, 2003.
10. Interface and Control Systems, SCL Home Page, sclrules.com
11. M. C. Malin and K. S. Edgett, “Evidence for recent groundwater seepage and surface runoff on Mars,” Science, 288, 2330-2335, 2000.
12. N. Muscettola, G. Dorais, C. Fry, R. Levinson, and C. Plaunt, “IDEA: Planning at the Core of Autonomous Reactive Agents,” Proceedings of the Workshops at the AIPS-2002 Conference, Tolouse, France, April 2002.
13. NASA Ames, Remote Agent Experiment Home Page, http://ic.arc.nasa.gov/projects/remote-agent/. See also Remote Agent: To Boldly Go Where No AI System Has Gone Before. Nicola Muscettola, P. Pandurang Nayak, Barney Pell, and Brian Williams. *Artificial Intelligence* 103(1-2):5-48, August 1998
14. The PROBA Onboard Autonomy Platform, http://www.estec.esa.nl/proba/
15. G. Rabideau, R. Knight, S. Chien, A. Fukunaga, A. Govindjee, "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.
16. J. Kurien and P. Nayak. “Back to the future for consistency-based trajectory tracking.” In Proceedings of the 7th National Conference on Artificial Intelligence (AAAI'2000), 2000.
17. B. Cichy, S. Chien, S. Schaffer, D. Tran, G. Rabideau, R. Bote, Dan Mandl, S. Frye, S. Shulman, J. Van Gaasbeck, D. Boyer, “Validating the EO-1 Autonomous Science Agent,” International Workshop on Planning and Scheduling for Space, Darmstadt, Germany, June 2004.
18. S. Chien et al., “Autonomous Science on the EO-1 Mission,” International Symposium on Artificial Intelligence, Robotics, and Automation in Space (i-SAIRAS 2003). Nara, Japan. May 2003
19. D. Tran et al., “Flight Software Issues in Automated Onboard Planning: Lessons Learned on EO-1,” International Workshop on Planning and Scheduling for Space, Darmstadt, Germany, June 2004.

ACKNOWLEDGEMENT

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.