Lessons learned from the Autonomous Sciencecraft on Earth Observing One

Steve Chien, Rob Sherwood,
Daniel Tran, Benjamin Cichy,
Gregg Rabideau,
Rebecca Castaño,
Ashley Davies

Jet Propulsion Laboratory,
California Institute of
Technology

Dan Mandl, Stuart Frye[1],
Seth Shulman[4]
Bruce Trout[2],
Jeff D'Agostino[3]

Goddard Space Flight Center

Darrell Boyer

Interface & Control Systems

Abstract

The Earth Observing One spacecraft has been under the control of AI software for several years – experimentally since 2003 and since November 2004 as the primary operations system. This software includes: model-based planning and scheduling, procedural execution, and event detection software learned by support vector machine (SVM) techniques. This software has enabled a 100x increase in the mission science return per data downlinked and a >$1M/year reduction in operations costs. In this paper we discuss the AI software used with a particular focus on issues related to the use of onboard mission planning.

Introduction

Space autonomy presents significant challenges for Artificial Intelligence systems.

-Limited, intermittent communications - a typical spacecraft in low earth orbit (such as EO-1) has 5 x 10-minute communications tracks per day. The spacecraft must be able to operate for long periods of time without supervision. Deep space missions may command weekly or even bi-weekly.

-System complexity - a typical spacecraft has thousands of components, each of which was carefully engineered to survive rigors of space (extreme temperature, radiation, physical stresses). Most of the components are one-of-a-kind and thus are not well characterized behaviorally by historical data.

-Limited observability – onboard computing to process telemetry is limited, onboard storage is limited, and downlink bandwidth 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.

-Limited computing power. Limited power and the need for radiation hardened processors limits onboard computing. A typical spacecraft CPUs offer 25 MIPS and 128 MB RAM – our EO-1 computing budget was offers 4 MIPS and 128MB RAM.

-Reliability due to high stakes - a typical space mission costs hundreds of millions of dollars. The total EO-1 mission cost is over $100 million dollars. Beyond financial cost, because of design and construction lead times, if a mission is lost it is typically years before another similar mission can be launched. Therefore reliability is paramount in space missions.

The Autonomous Sciencecraft Experiment (ASE) [Chien et al. 2005] has flown AI software onboard the Earth Observing One mission for several years. This software is so successful that it is now the primary operations software (for up to date mission status see - ase.jpl.nasa.gov) and ASE was a co-winner of the 2005 NASA Software of the Year Award. We briefly describe the ASE software, impact on the EO-1 mission, lessons for AI research with a particular focus on automated planning & scheduling, and future directions for AI in space exploration.

The ASE software represents a traditional three-layered architecture with the CASPER continuous planning system as the deliberative layer [Chien et al.2000], Spacecraft Command Language (SCL) [ICS 2006] as the execution layer, and the original EO-1 flight software as the skills layer. A version of the Livingstone 2 mode identification software also flew as an experiment during 2005 [Hayden et al. 2004]. The ASE software enables the spacecraft to:

- Process imagery onboard to detect science events such as volcanic activity, flooding, and cryosphere events. Some of these classifiers were developed using Support Vector Machine (SVM) machine learning techniques [Mazzoni et al 2005].
- Replan future observations using CASPER based on the priority of observations triggered detected events relative to prior scheduled observations.

- Robustly execute plans using SCL despite timing uncertainties and anomalies endemic in robotic real-time systems.

This autonomous operations is illustrated by a sequence of events that occurred on the 7th of May 2004. In this series of events, the EO-1 spacecraft, under the control of the ASE software, detects increased volcanic activity at the Mt. Erebus Volcano in Antarctica and responds without human intervention.

Time (in GMT)	Event
1340	Under ASE control, EO-1 images Mt. Erebus
1350-1418	ASE extracts image to RAM for processing
1418-1447	Onboard processing detects increased thermal activity
1447-1503	Planner selects reaction observation and schedules repeat observation, bumping Stromboli observation
1558	Summary of detection downlinked at first ground contact
2010	EO-1 images Erebus on day overflight

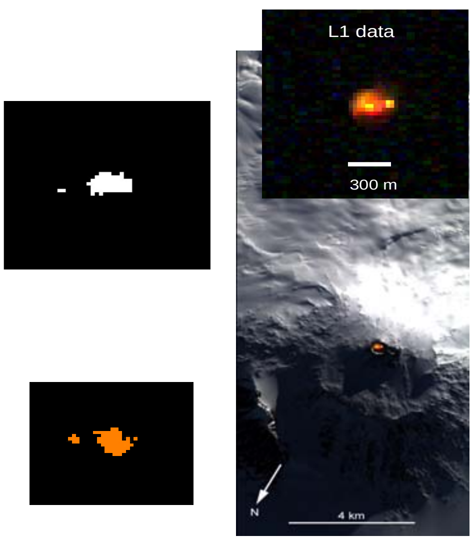

Figure 1: Actual Imagery taken by EO-1 of Mt Erebus 7th May 2004. Summary images at left, final science product at right with context image.

The key points from this actual example are:

- ASE enabled rapid notification of the volcanic event (ground alert within about 2 hours). Prior to ASE downlink and processing of data nominally would require 7-14 days.
- ASE enables rapid re-imaging of the volcanic event – in this case in about 6 hours from detection. Prior to ASE nominal observation planning would be 5-12 days in advance. Therefore closed loop response would be 7-14 + 5-12 days or 12-26 days.
- All of the retargeting is automated driven by scientist priorities. With manual operations often for high priority targets data can be rush downlinked, processed, and observation plans manually changed but at significant effort. ASE has executed thousands of observations autonomously.

Impact

The ASE software has had several benefits:

- By moving decision-making onboard, ASE has enabled a 100x increase in science return per Mbyte downlinked. This is because ASE can detect and downlink just the relevant data – the thermal signature of the volcano, the flood extent map, the sea/ice extent boundary, avoiding downlinking the entire image. For further details of the science applications see [Davies et al. 2005, Doggett et al. 2005, Ip et al. 2005].

-By removing the dependency on ground contacts to downlink the data, process, decide and uplink changes ASE has reduced the routine response time for images from days to several hours.

-By automating large parts of the observation planning and command generation process, ASE enabled a reduction in the EO-1 mission operations costs of over $1M/year. This reduction was instrumental in the extension of the EO-1 mission from 2005-2007.

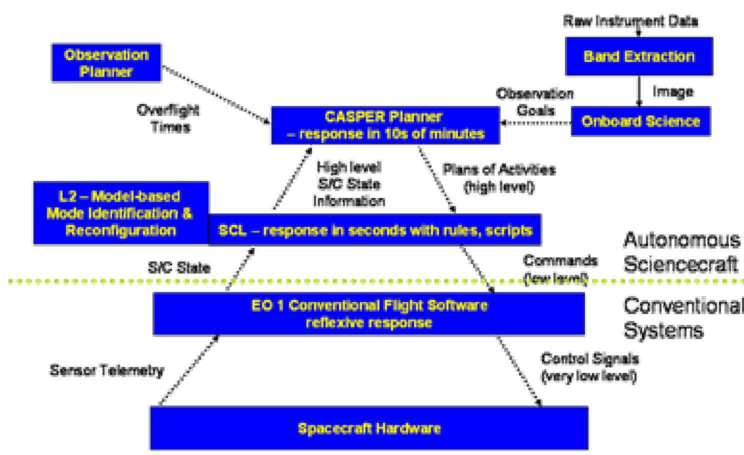

Figure 2: ASE Flight Software Architecture

Lessons Learned for Planning & Scheduling for Space

Several major lessons learned for planning & scheduling as well as for autonomous systems have emerged from the ASE project.

- In order to achieve extremely high levels of reliability, autonomy software (and automated planning & scheduling) must exist in an architecture designed for reliability (in our case a robust, layered architecture) and be developed within a rigorous framework for systems engineering and validation (model, software, and process). For details on our process see [Cichy et al. 2004]. The ASE flight software architecture is shown in Figure 2.

- Iterative repair and continuous planning techniques [Chien et al. 2000] were essential in enabling solution of large-scale planning & scheduling problems within very limited computation(4 MIPS) (for further details see [Tran et al. 2004]). The CASPER onboard planner uses a committed representation of plans that enables rapid repair in response to anomalies or last minute requests. CASPER on EO-1 also uses an abstracted long-term planning representation to fit a large (5000+ activities in a weekly schedule) schedule within restricted RAM constraints (32MB heap for all of ASE). This abstracted representation is shown in Figure 3.

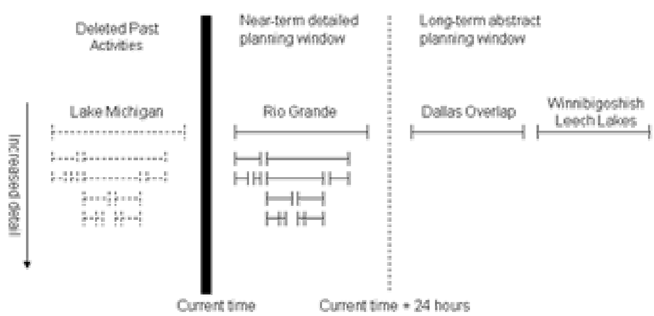

Figure 3: Abstracted Planning Representation used by CASPER on ASE

- Many issues relating to integrating planning & execution had to be engineered for ASE due to time restrictions. For example, CASPER onboard EO-1 uses a series of overlapping windows to manage the continuous planning, optimization, and execution of activities. At the furthest out timeframes CASPER attempts to achieve observation goals by expanding them out but may delete observations goals to resolve conflicts. In this time period, CASPER may add back deleted goals to attempt to pack them in. While CASPER/ASPEN has a general optimization capability, this simpler scheme was flown in an order to reduce the computation needed onboard and to increase the predictability of the system. Figure 4 shows the integration of the planning windows with execution. One issue is poor predictability of the execution times of activities. For example, if a subsystem is scheduled to be turned on at 10 am, the command must be sent out prior to 10 am due to subsystem commanding delays. If the command executes prior to 10 am, the sensed power draw will be added to a predicted power draw starting at 10 am – thus double book-keeping the power utilization. In the worst case, this will cause a power deficit later on in the schedule and the planner will delete activities in order to resolve this deficit. However by 10am, the situation will be reconciled with the predicted power draw being withdrawn. Ideally a state estimation system would receive the early notification of the power draw and note that the most likely interpretation is that the command started early – suppressing the later prediction of power draw. In other cases, operators would command the spacecraft from the ground (in a so-called blind acquisition). In these cases operator actions appear to the autonomy software as unexplained system, state, and resource changes. Supressing autonomous responses to these changes is desirable to avoid large scale schedule disruptions from the planner.

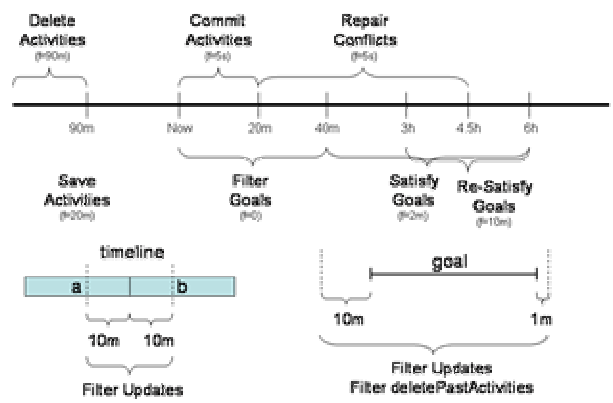

Figure 4: Continuous Planning meets embedded execution

- Complementary representations enabled natural representation of mission operations constraints. The procedural script and rule representations of SCL as well as the procedural HTN, and declarative planner activity representations were all essential to represent the myriad of vehicle and science constraints. This supports earlier work in practical planning [Estlin et al 2001, Wilkins et al, 2001] Examples of redundant constraints checking in both the CASPER and SCL models is shown in Figure 4. This was particularly important as ASE was originally planned for flight on the Techsat-21 mission [Chien et al. 2002] until launch uncertainty forced a switch to the EO-1mission. This required all of the modeling be re-done for the new mission – not possible without the broad expressivity and naturalness of the models.

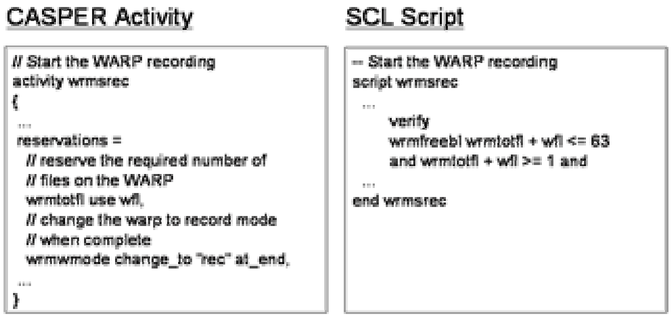

Figure 5: Examples of redundant and complementary CASPER and SCL models for recording data to the Solid State Recorder.

- Model-based programming allowed both rapid deployment and considerable software reuse. From carrier change from Techsat-21 to first test flight was approximately 18 months. The vast majority of the flight software (over 80%) was reused from the Techsat-21 deployment to EO-1 (see Table 1 for details).

	EO-1 code (kLOC)	Reused from TS-21 (kLOC)
CASPER Flight	223	200
SCL Flight	214	200
Science Flight	50	0
Ground Automation	25	0
Testing	40	20
Total	593	420

Table 1: Software Reuse from Techsat-21 to EO-1

- All of the AI technologies had to operate extremely reliably for the integrated system to function reliably. This imposed a major burden on the validation of the components and system [Cichy et al. 2004]. We achieved this reliability requirement through the use of: redundant, layered architecture and supporting models, a rigorous systems and requirements engineering process, a multi-level testing process, and an incremental spiral development and deployment process. Even with this strategy, the systems and software engineering challenges were arduous. Also, because we were deploying to a spacecraft in extended mission, it was possible to deploy with greater risk than for a launching spacecraft in prime mission. Because of this experience and in anticipation of future missions expected risk averse posture, JPL and MIT have initiated a program in model-based monitoring and diagnosis of software intended to enable supervisory systems to track, detect, and correct operations of complex onboard software such as autonomy software.

Related Work and The Future

The Remote Agent Experiment [Muscettola et al. 1998] flew onboard the Deep Space One spacecraft in May 1999 for about 48 hours. ASE advances beyond RAX in duration and reliability, integrating science operations and dramatic improvements in science return, and demonstrating over $1M/year in operations costs savings. ASE also demonstrated new technologies (e.g. continuous planning). Several research projects exist in rover autonomy [Castano et al. 2006, Wetergreen et al. 2005], space constraints preclude comparison here.

Onboard science to enable tracking of dust devils and clouds is scheduled for upload to the Mars Exploration Rovers currently at Mars [Castano et al. 2006] (upload June 2006). Onboard science technology is also being infused into the Mars Odyssey orbiter to search for thermal anomalies, polar frost, water ice clouds, and dust storms [Wagstaff et al. 2005, 2006]. These onboard science and closed loop science autonomy concepts are being enhanced for a wide range of future science missions including landers, rovers, aerobots to Titan, submersibles to Europa, and many others [Chien et al. 2003]. The autonomy technologies are also applicable to future Exploration Systems mission involving crewed exploration of the Moon and Mars.

More recently, enabled by ASE automation, EO-1 has been integrated with a wide range of other space-borne and ground-based sensors to track [Chien et al. 2005]. In this effort, other sensors are used to track volcanoes, flooding, lake and sea ice freeze-thaw, and snow fall and melt, with EO-1 being automatically triggered to image specific targets when relevant. This effort has enabled literally hundreds of timely observations of transient science events in a fully automated fashion.

Acknowledgements

ASE is a broad multi-disiplinary, multi-institutional effort (ase.jpl.nasa.gov): JPL: Steve Chien, Rob Sherwood, Becky Castano, Ashley Davies, Gregg Rabideau, Daniel Tran, Ben Cichy, Nghia Tang, Rachel Lee, Russell Knight, Steve Schaffer, NASA Goddard Space Flight Center: Dan Mandl, Stuart Frye (Mitretek), Seth Shulman (Honeywell-TSI), Joe Szwaczkowski (Honeywell-TSI), Josh Bowman (Adnet), Rob Bote (Honeywell-TSI), Tom Brakke, Stephen Ungar, Lawrence Ong, Interface and

Control Systems: Darrell Boyer, Jim VanGaasbeck, Microtel: Bruce Trout, Nick Hengemihle, Jerry Hengemihle, The Hammers Company: Jeff D'Agostino, Kathie Blackman, Arizona State University: Ronald Greeley, Thomas Doggett, University of Arizona: Victor Baker, James Dohm, Felipe Ip.

The Livingstone 2 on EO-1 team (ASE team plus NASA Ames: Adam Sweet, Anupa Bajwa, Lee Brownston, Sandra Hayden, Scott Christa, NASA GSFC: Joe Young, ICS: Phil Courtney).

The MER Onboard Science Team: , JPL: Andres Castano, Alex Fukunaga, Jeff Biesiadecki, R. Castano, S. Chien, S. Schaffer, Arizona State University: Ronald Greeley, Lynn Nekrease, Patrick Whelley, Texas A&M: Mark Lemmon/

The THEMIS Onboard Science Team: JPL: R. Castano, K. Wagstaff, N. Tang, T, Stough, S. Chien, ASU: P. Christensen, J. Bandfield, G. Mehall, GSFC: M. Smith.

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.

References

A. Castano, A. Fukunaga, J. Biesiadecki, R. Castano, S. Chien, R. Greeley, L. Nekrease, P. Whelley, M. Lemmon, Autonomous Detection of Dust Devils and Clouds on Mars, Intl Conf Image Proc, Atlanta, Oct 2006.

R. Castano, T. Estlin, D. Gaines, A. Castano, C. Chouinard, B. Bornstein, R.C. Anderson, S. Chien A. Fukunaga and M. Judd, "Opportunistic Rover Science: Finding and Reacting to Rocks, Clouds and Dust Devils," Proc 2006 IEEE Aerospace Conference, Big Sky, Montana, March 2006.

S. Chien, R. Knight, A. Stechert, R. Sherwood, and G. Rabideau, "Using Iterative Repair to Improve the Responsiveness of Planning and Scheduling," Proc. 5th International Conference on Artificial Intelligence Planning and Scheduling, Breckenridge, CO, April 2000.

S. Chien, et al., “Revolutionary Deep Space Science Missions Enabled by Onboard Autonomy,” Intl Symp AI, Robotics, & Automation in Space (i-SAIRAS). Nara, Japan. May 2003

S. Chien, et al., Using Autonomy Flight Software to Improve Science Return on Earth Observing One, Journal of Aerospace, Computing, Information & Communications, AIAA. April 2005.

S. Chien, et al., “An Autonomous Earth Observing Sensorweb,” *IEEE Intelligent Systems*, May-June 2005, pp. 16-24. 17.

B. Cichy et al., “Validating the EO-1 Autonomous Science Agent,” Intl Workshop on Planning and Scheduling for Space, Darmstadt, Germany, June 2004.

A. G. Davies, S. Chien, V. Baker, T. Doggett, J. Dohm, R. Greeley, F. Ip, R. Castano, B. Cichy, R. Lee, G. Rabideau, D. Tran and R. Sherwood (2005) Monitoring Active Volcanism with the Autonomous Sciencecraft Experiment (ASE). Remote Sensing of Environment, Volume 101 Issue 4 pp. 427-446.

T. R. Doggett, R. Greeley, A. G. Davies, S. Chien, B. Cichy, R. Castano, K. Williams, V. Baker, J. Dohm and F. Ip (2005) Autonomous On-Board Detection of Cryospheric Change. Remote Sensing of Environment, Volume 101 Issue 4 pp. 447-462.

T. Estlin, S. Chien, and X. Wang, ``Hierarchical Task Network and Operator-Based Planning: Two Complementary Approaches to Real-World Planning," Journal of Experimental and Theoretical Artificial Intelligence, 13:379-395, 2001.

S. Hayden, et al., Livingstone model-based diagnosis of Earth Observing One, AIAA Intelligent Systems, 2004.

Interface & Control Sys, www.interfacecontrol.com

F. Ip, J. M. Dohm, V. R. Baker, T. Doggett, A. G. Davies, R. Castano, S. Chien, B. Cichy, R. Greeley, and R. Sherwood (2005) Development and Testing of the Autonomous Spacecraft Experiment (ASE) floodwater classifiers: Real-time Smart Reconnaissance of Transient Flooding, Remote Sensing of Environment , Volume 101 Issue 4 pp. 463-481.

D. Mazzoni et al, “Learning Classifiers for Science Event Detection in Remote Sensing Imagery,” i-SAIRAS 2005.

N. Muscettola, P. Nayak, B. Pell, and B. Williams., Remote Agent: To Boldly Go Where No AI System Has Gone Before. Art Int Jnl 103(1-2):5-48, August 1998.

D. Tran, et al. “Flight Software Issues in Onboard Automated Planning: Lessons Learned on EO-1” Intl Wkshp Plan Sched for Space. Darmstadt, Ger. June 2004.

K. Wagstaff, J. Bandfield, R. Castano, S. Chien, and M. Smith, “Dust storms and water ice clouds: Feature detection for use onboard THEMIS,” 37th Lunar and Planetary Science Conference, 2006.

K. Wagstaff, R. Castano, S. Chien, A. Ivanov, and T. Titus. An Onboard Data Analysis Method to Track the Seasonal Polar Caps on Mars, i-SAIRAS 2005.

D. Wettergreen, N. Cabrol, V. Baskaran, F. Calderon, S. Heys, D. Jonak, R.A. Luders, D. Pane, T. Smith, J. Teza, P. Tompkins, D. Villa, C. Williams, and M. D. Wagner. Second Experiments in the Robotic Investigation of Life in the Atacama Desert of Chile, Proc. of iSAIRAS, 2005.

D. E. Wilkins, D. E., and M. desJardins. A call for knowledge-based planning. AI Magazine 22(1) 2001.