The Techsat-21 Autonomous Space Science Agent

Steve Chien, Rob Sherwood, Gregg Rabideau, Rebecca Castano, Ashley Davies, Michael Burl, Russell Knight, Tim Stough, Joe Roden
Jet Propulsion Laboratory
California Institute of Technology

Paul Zetocha, Ross Wainwright, Pete Klupar
Air Force Research Laboratory

Jim Van Gaasbeck, Pat Cappelaere, Dean Oswald
Interface and Control Systems

ase.jpl.nasa.gov

Firstname.lastname@jpl.nasa.gov

ABSTRACT[*]

The Autonomous Sciencecraft Experiment (ASE) will fly onboard the Air Force TechSat-21 constellation of three spacecraft scheduled for launch in 2004. ASE uses onboard continuous planning, robust task and goal-based execution, model-based mode identification and reconfiguration, and onboard machine learning and pattern recognition to radically increase science return by enabling intelligent downlink selection and autonomous retargeting. In this paper we discuss how these AI technologies are synergistically integrated in a hybrid multi-layer control architecture to enable a *virtual spacecraft science agent.* We also describe our working software prototype and preparations for flight.

Categories and Description Headers

I.2.8 **[Problem Solving, Control Methods, and Search]** - Heuristic methods, Plan execution, formation, and generation, Scheduling

General Terms

Algorithms, Experimentation

Keywords

Space exploration agent, science agent, planning and scheduling, robust execution, mode identification

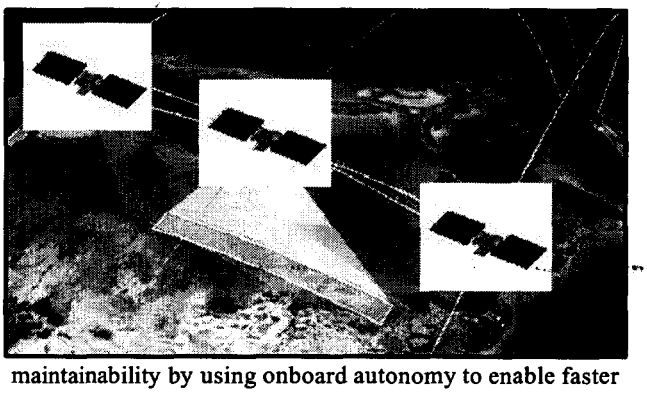
[*] Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS'02, July 15-19, 2002, Bologna, Italy.

Copyright 2002 ACM 1-58113-480-0/02/0007...$5.00.

1. Introduction

There is an increasing desire in many organizations, including NASA and the DoD, to use onboard decision-making to accomplish complex mission objectives. The Air Force Research Laboratory (AFRL) has initiated the TechSat-21 program to serve as a demonstration mission for a new paradigm for space missions. This paradigm seeks to reduce costs and increase system robustness and maintainability by using onboard autonomy to enable faster response times and improve operations efficiency.



TechSat-21 is scheduled for launch in January 2006 and will fly three satellites in a near circular orbit at an altitude of approximately 550 Km (See Figure 1.).

Figure 1 – Techsat-21 Configuration

The primary mission is one-year in length with the possibility for an extended mission of one or more additional years. During the mission lifetime the cluster of satellites will fly in various configurations with relative separation distances of approximately 100 meters to 5 Km. One of the objectives of TechSat-21 is to assess the utility of the space-based, sparse-array aperture formed by the satellite cluster. For TechSat-21, the sparse array will be used to synthesize a large radar antenna. Three modes of radar sensing are planned: synthetic aperture radar (SAR) imaging, moving target indication (MTI), and geo-location.

The principal processor onboard each of the three TechSat-21 spacecraft is a BAE Radiation hardened 175 MIPS, 133MHz PowerPC 750 running the OSE 4.3 operating system from Enea Systems. OSE was chosen because it is

inherently message passing based and particularly suitable for distributed applications. Each satellite will have 256 Kbytes of EEPROM for boot loads and 128 Mbytes of SDRAM. Communications will be through a Compact PCI bus. Eight disk drives per spacecraft will be used for storage of payload data and some large flight software components.

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, and changes relative to previous observations
- *Model-based mode identification and execution (MI-R)* that uses component-based hardware models to analyze anomalous situations and to generate novel command sequences and repairs.
- *Robust execution management software* using the Spacecraft Command Language (SCL) [7] package to enable event-driven processing and low-level autonomy
- The Continuous Activity Planning, Scheduling, and Replanning (CASPER) [4] *planner* that will replan activities, including downlink, based on science observations in the previous orbit cycles
- Observation Planning (OP) software will enable the cluster to predict overflights of targets to facilitate onboard retasking.

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 X-band radar data (from shuttle missions) to automatically identify regions of interest including: regions of change (such as flooding, ice melt, and lava flows), and feature recognition (such as crater and volcano recognition). Such onboard science will enable retargeting and search, e.g., shifting the radar aim-point on the next orbit cycle to identify and capture the full extent of a flood. Onboard science analysis would also enable capture of short-lived science phenomena at the finest time-scales without overwhelming onboard caching or downlink capacities. 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) 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 for the constellation.

The robust execution system (SCL) accepts the CASPER-derived plan as an input and expands the plan into low-level commands.

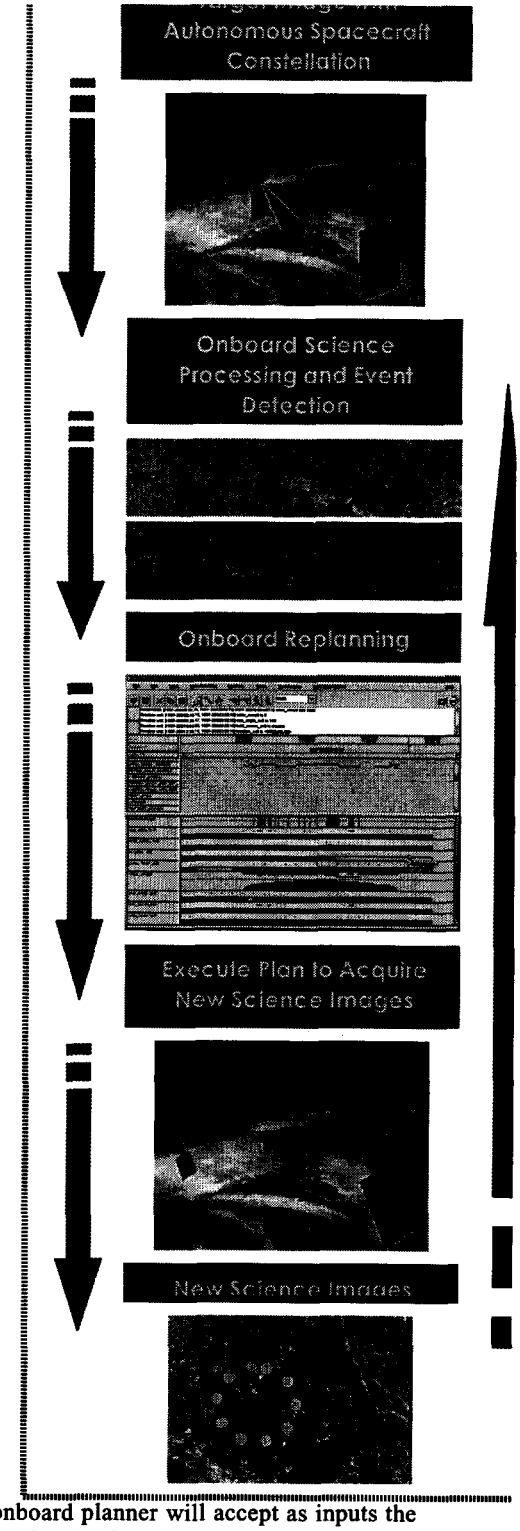


Figure 2 - ASE Mission Scenario

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.

Livingstone 2 performs model-driven estimation of spacecraft state and Burton also accepts configuration goals from SCL.

One of the ASE demonstration scenarios involves monitoring of flooding regions in Arizona (see Figure 2). Radar data have been used in ground-based analysis to study this phenomenon. The ASE concept would be applied as follows:

1. Initially, ASE has a list of science targets to monitor.
2. As part of normal operations, CASPER generates a plan to monitor the targets on this list by periodically imaging them with the radar.
3. During such a plan, a spacecraft images a river area with its radar.
4. Onboard, a reflectivity image is formed.
5. The *Onboard Science Software* compares the new image with previous image and detects that the water region has changed due to flooding. Based on this change the science software generates a goal to acquire a new high-resolution image of an area of flooding.
6. 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. During this process CASPER interacts with the Observation Planner to compute when the spacecraft will overfly the target and determine the required slews to acquire the target.
7. SCL executes this plan in conjunction with several autonomy elements. Mode Identification assists by continuously providing an up to date picture of system state. Reconfiguration (Burton) achieves configurations requested by SCL.
8. Based on the science priority, imagery of identified "new flood" areas are downlinked. This science priority could have been determined at the original event detection or based on subsequent onboard science analysis of the new image.

As demonstrated by this scenario, many different capabilities are used synergistically to enable the spacecraft to behave as an autonomous exploration agent. In our agent architecture, ASE allocates responsibilities both based on abstraction level and domain (e.g., same level of abstraction but a specific discipline such as science or maneuver planning). Specifically, each of the software components has responsibilities as follows. First, for the areas of science decision-making and maneuvers, responsibilities are delegated based on the discipline involved. All of the processing and analysis of science data analysis is performed by the Onboard Science software. This design makes sense because the science processing we are performing is very specialized image processing and pattern recognition and requires special purpose algorithms. Because this is primarily a batch process, there is no real-time decision-making component to the Onboard Science software. However, this is a Techsat-21 specific distinction. Many other autonomous science missions might have a real-time science component, such as to rapidly detect a very short duration science event (such as a supernova) or to control a science instrument rapidly based on science analysis. The Observation Planner software is used to reason about maneuvers, determine when a target can be observed, and determine when communication with the spacecraft is possible. Again, this architecture is chosen because this decision-making capability relies on highly specialized reasoning algorithms to minimize fuel consumption and to reason geometrically about orbits and orbital dynamics. In this case there is both a plan-time and real-time execution component.

In the space operations arena, ASE uses CASPER, SCL, and MI-R to provide distinct, synergistic capabilities. Long-term mission planning, which requires search and the ability to reason about complex states and resources, is performed by CASPER. CASPER is able to respond on a several minute timescale to replan in response to anomalies and science opportunities. CASPER uses a model-based approach to represent operations knowledge. For decision-making at a lower level and requiring a more rapid response, ASE uses SCL and MI-R. SCL and MI-R are able to respond on the order of seconds, and in some cases even more quickly. SCL and MI-R are complementary in that SCL uses a procedural (script and rule based) representation while MI-R uses a declarative stochastic finite state model. These representations are complementary; in different cases one may be more appropriate. Additionally, MI-R's stochastic model is particularly adept at interpreting noisy data from sensors and achieving hardware configurations in the presence of unreliable hardware.

While the Techsat-21 mission is amenable to a multi-agent formulation with each of the three spacecraft being a separate agent, ASE operates the three spacecraft as a single agent exerting centralized control over the three spacecraft. From a mission perspective, operating three spacecraft as self-coordinating agents was viewed as being too risky. After all, ASE will already be performing revolutionary on-board decision-making.

ASE will fly on the Techsat-21 mission and the necessary software is currently being matured and brought into flight readiness. A working version of the flight software described in this paper exists operating on Sun workstations on a wide range of operational scenarios. Already, two out of five components are operational on the flight processor. The remaining three components have been ported to the flight operating system on embedded

processor (completed in the Spring of 2002). Final delivery of the spacecraft and software is expected complete in late 2003. Nominal launch date is January 2006.

2. Onboard Science

There are two components of the onboard science software, the *image formation module* and the *onboard science algorithms*. The image formation module forms a (possibly reduced resolution) SAR image onboard the spacecraft from the raw phase history (demodulated I and Q returns). In the ASE mission concept, we only need to form a few images per orbit cycle (in contrast to a global mapping mission such as Magellan); hence, the necessary processing can be carried out onboard. Our baseline calculations estimate that forming a 15 km diameter spot size (dependent upon grazing angle) 10-meter by 10-meter resolution image can be formed can be formed in at best 18 seconds (with full processor utilization). A 2-meter resolution would require approximately 45 minutes. Both these timescales are considerably less than the 90-minute orbit decision cycle for downlink.

Once the image has been formed, the *onboard science algorithms* can then analyze the SAR image to create derived science products and detect trigger conditions, such as change relative to a previous orbit cycle. For example, fresh lava and old lava have very different backscatter properties; thus, new lava flows can be easily detected and localized. Likewise, water has very different backscatter characteristics than soil, enabling detection of flooding.

Figure 3 – Automatically identified lava cones in X-SAR image of Lava Beds National Monument, CA

We are currently investigating demonstrating several methods of converting images into derived science products. The derived products will in effect be summarizations that are significantly more compact than the raw image (or phase history) data. Intensity and texture-based segmentation (in common use for ground-based processing) will be evaluated for effectiveness in generating terrain boundary descriptions and region summarizations (e.g., a flooded region will be described by an average radar cross-section and a polyline outlining its boundary). Statistical pattern recognition techniques [1] [3] will be used to identify specific types of features such as volcanoes, lakes, and iceberg fragments. Figure 3 shows results from a prototype lava cone recognition algorithm under development for ASE flight. Output from such a module could be used to downlink higher resolution data around items of interest or by downlinking a summary catalog of the interesting features. Recently developed discovery techniques [2] will also be applied to identify "interesting" regions that differ from their local background leading to a compact description of an image in terms of sub-image patches and locations (see Figure 4).

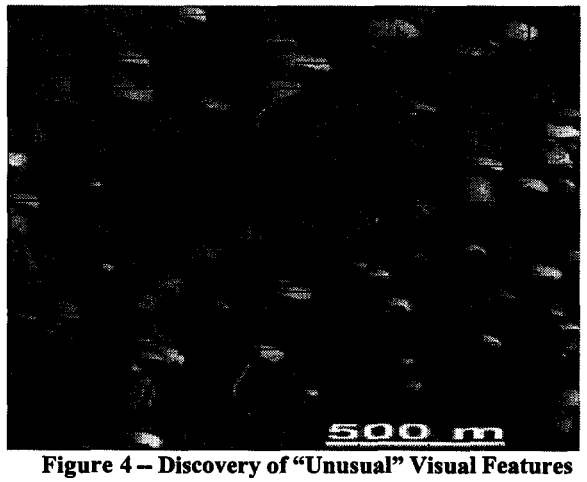


Figure 4 – Discovery of "Unusual" Visual Features

In addition to calculations based on a single image, the onboard science analysis software will include change detection algorithms that compare images of the same region taken at different times. The change detection capability is particularly relevant for capture of short-term events at the finest time-scale resolutions without overwhelming onboard caching systems and for compressing long-term "monitoring" observations in which changes are infrequent. For space science missions, example applications include tracking atmospheric changes on Jupiter, Neptune, or Triton (from optical image data), tracking ice plate movement on Europa, monitoring known (and identifying new) volcanoes on Io, capturing fine time-scale events such as jet formation on comets or phase transitions in ring systems, and detecting new cratering on planets and moons.

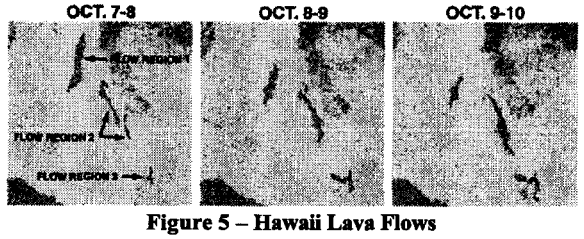


Figure 5 – Hawaii Lava Flows

To detect change, we will test for statistically significant differences in derived descriptors such as region sizes, locations, boundaries, and histograms, as well as in the raw

pixel data. The latter case is complicated by the need to ensure that the two images are approximately co-registered. In part, the orbit repeatability and small absolute positional uncertainty of the TechSat-21 group will help insure approximate co-registration. Also, since the magnitude of change necessary to initiate a trigger event can be specified as a parameter, some degree of robustness to image misalignment will be built in. For change detection, radar observations have the advantage that the illumination, target, and receiver geometry remains basically the same from pass to pass. (In optical imagery, irrelevant change caused by sun position complicates the change detection problem.) Figure 5 contains successive X-SAR radar images indicating lava flow on the Kilauea volcano in Hawaii. The changes in the highlighted areas of the image are indicative of lava flow that occurred in between images. This is the type of change detection that our algorithms will perform onboard Techsat-21.

All of the algorithms described scale linearly in the number of image pixels. Hence, image resolution can be selected appropriately to insure that computational and memory requirements fit within the onboard processing capabilities. For example, a previous study of the recognition algorithm in [1] indicates complexity on the order of 250 operations per pixel to reliably detect a particular type of Venus small shield volcano in the Magellan SAR data. Using this figure as a baseline, we could process approximately 10^5 pixels per second on the PowerPC 750 flight processor.

Detection of the image- and change-based triggers described here will enable a range of automated spacecraft reactions. On the conservative end of the spectrum, triggers can be used to prioritize data for downlink. For example, regions in which change was detected may be downlinked first. (With TechSat-21, it will take a full four days to downlink the entire onboard cache of the three spacecraft.) Early access to the "interesting data" would be especially valuable to the project scientist, potentially enabling a request for modification of the original observation plan.

A slightly more aggressive use of the trigger information involves actually "discarding data". For example, if nothing significant has changed in a region, exclude that region from the downlink. Although the scientist would never like to discard data, the realities of a finite onboard cache and constrained downlink bandwidth will sometimes force a discard to satisfy the primary objective. For example, if the science goal is to capture the fine temporal details of jet formation on a comet, the onboard cache will quickly overflow unless older data that doesn't contain the desired event is discarded or degraded to lower resolution.

A third, more aggressive, but potentially extremely rewarding, use of the trigger information that we will demonstrate onboard TechSat-21 is to autonomously retarget observations. For example, if an image indicates flooding in a region, subsequent orbits will employ the planner to close the loop onboard and use a modified radar aim-point in an attempt to capture the full scope of the flooding. Similarly, since many geological features are spatially clustered (e.g., volcano fields, hydrothermal vents), detection of some features will be used to seed a broad area search (e.g., using the three spacecraft radars in a coordinated effort to look in the surrounding area for additional instances).

3. Robust Execution

Techsat-21 will fly the Spacecraft Command Language (SCL) [7] 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 enables either loose or tight coupling between SCL and other flight software as appropriate. Dynamic messages are supported to allow for future growth as ever-smarter software agents are added to the constellation in different satellites.

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 ASE concept, SCL scripts will also be planned and scheduled by the CASPER onboard planner. The science processing agents, cluster management software, and SCL work in a cooperative manner to generate new goals for CASPER. These goals are sent with the messaging system.

Spacecraft telemetry from all satellites is gathered onboard and fed into the integrated expert system. Significant change in the data will trigger user-defined rules. Those rules can be used for fault detection, isolation and recovery (FDIR). In that case, rules can be used to execute recovery scripts. Another application of rules is for mission constraint checking to prevent operator errors or, more simply, command pre-processing.

SCL is a mature software product, and has successfully flown on Clementine-I and ROMPS. SCL has also been used in a wide range of ground-based control and operations contexts. As such it represents a good basis for integrating the multiple ASE autonomy functions: onboard science, mode identification and reconfiguration, planning, and constellation management.

4. Model-based Monitoring and Reconfiguration

CASPER generates a mission level plan that includes a sequence of behavior goals, such as producing thrust. An 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, the executive 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, an executive must ascertain the health state of components, determine repair options when viable, and select a course of action among the space of redundant options.

Interpreting sensor information and generating sequences to handle a breadth of novel situations requires extensive reasoning about physical processes and state changing actions. A model-based executive performs this reasoning automatically from an onboard model [14]. In particular, a model-based executive is given a model of the spacecraft hardware, including a set of component models and a schematic that describes component interconnections. The executive uses the model to track planner goals, confirm hardware modes, reconfigure hardware, generate command sequences, detect anomalies, isolate faults, diagnose, and perform repairs.

The executive receives a stream of hardware configuration goals and sensor information. It uses sensor information to infer the state of the hardware and then continually tries to transition the hardware towards a state that satisfies the current configuration goals. The model-based executive is reactive in the sense that it reacts immediately to changes in goals and to failures, that is, each control action is incrementally generated using the new observations and goals given in each state.

The executive uses its model to determine the desired control sequence in three stages — mode estimation (ME), mode reconfiguration (MR) and model-based reactive planning (MRP). ME and MR setup the planning problem, identifying initial and target states, while MRP reactively generates a plan solution. More specifically, MI incrementally generates the set of most likely state trajectories of the hardware that are consistent with the hardware model and the sequence of observations and control actions. This is maintained as a set of most likely current states. MR uses the hardware model and the most likely current state generated by ME to determine a reachable hardware state that satisfies the current goal configuration. MRP then generates the first action in a control sequence for moving from the most likely current state to the target state. After that action is performed ME confirms that the intended next state is achieved.

Model-based reactive planning is traditionally viewed as intractable for real world problems. We address this intractability through a set of model-compilation, causal analysis, and online policy construction methods. The result is a model-based reactive planner that is sound and complete. It generates the first control action of a valid plan in average case constant time, and compensates for anomalies at every step. Finally, it will not generate irreversible, potentially damaging sequences except to effect repairs.

Our model-based execution framework uses the Mode Estimation capabilities of Livingstone 1 and 2, described in [9,14] and developed at NASA Ames. The marriage between the model-based executive and SCL provides a powerful hybrid execution capability with an expressive scripting language and an extensive capability to generate novel responses to anomalous situations.

5. Onboard Mission Planning

Traditionally, the majority of planning and scheduling research has focused on a batch formulation of the problem. In this approach, when addressing an ongoing planning problem, time is divided up into a number of planning horizons, each of which lasts for a significant period of time. When one nears the end of the current horizon, one projects what the state will be at the end of the execution of the current plan. (See Figure 6.) The planner is invoked with a new set of goals for the new horizon, and the expected initial state for the new horizon. The planner then generates a plan for the new horizon. As an example of this approach, the Remote Agent Experiment operated in this fashion [8].

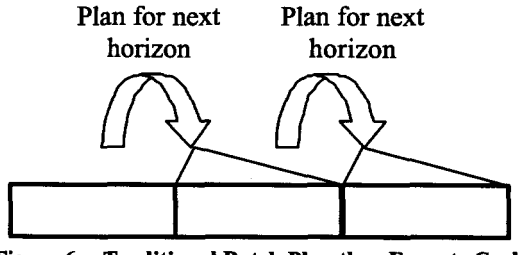


Figure 6 -- Traditional Batch Plan then Execute Cycle

This approach has a number of drawbacks. In this batch oriented mode, typically planning is considered an off-line process, which requires considerable computational effort, and there is a significant delay from the time the planner is invoked to the time that the planner produces a new plan. If a negative event occurs (e.g., a plan failure), the response time until a new plan is generated may be significant. During this period the system being controlled must be operated appropriately without planner guidance. If a positive event occurs (e.g., a fortuitous opportunity, such as activities finishing early), again the response time may be significant. If the opportunity is short lived, the system must be able to take advantage of such opportunities without a new plan (because of the delay in generating a new plan). Finally, because the planning process may need to be initiated significantly before the end of the current planning horizon, it may be difficult to project what the state will be when the current plan execution is complete. If the projection is wrong the plan may not be executable.

To achieve a higher level of responsiveness in a *dynamic planning* situation, we utilize a *continuous planning* approach and have implemented a system called CASPER (Continuous Activity Scheduling Planning Execution and Replanning) [4]. Rather than considering planning a batch process in which a planner is presented with goals and an initial state, the planner has a current goal set, a plan, a current state, and a model of the expected future state. At

any time an incremental update to the goals, current state, or planning horizon (at much smaller time increments than batch planning)[1] may update the current state of the plan and thereby invoke the planner process. This update may be an unexpected event or simply time progressing forward. The planner is then responsible for maintaining a consistent, satisficing plan with the most current information. This current plan and projection is the planner's estimation as to what it expects to happen in the world if things go as expected. However, since things rarely go exactly as expected, the planner stands ready to continually modify the plan. From the point of view of the planner, in each cycle the following occurs:

- Changes to the goals and the initial state first posted to the plan,
- Effects of these changes are propagated through the current plan projections (including conflict identification)
- Plan repair algorithms [5] are invoked to remove conflicts and make the plan appropriate for the current state and goals

This approach is shown in 7. At each step, the plan is created by using incremental replanning from:

- The portion of the old plan for the current planning horizon
- The change (Δ) in the goals relevant for the new planning horizon
- The change (Δ) in the state
- The new (extended) planning horizon

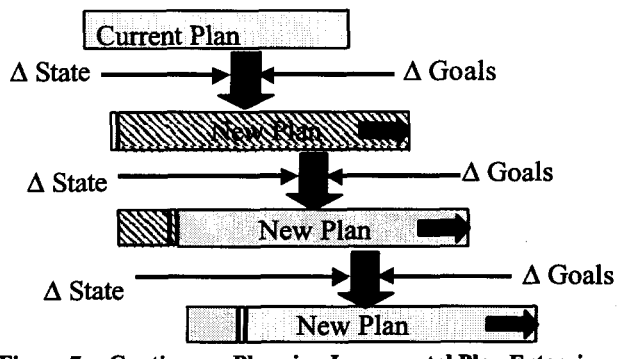


Figure 7 -- Continuous Planning Incremental Plan Extension

In the ASE concept, CASPER is responsible for long-term 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 and constellation operations constraints. For example, when change is detected in an image, CASPER plans a response. If it is appropriate to take a more detailed image of the change area, CASPER will modify the operations plan to include the necessary activities to re-image. If this includes changing the formation of the constellation, the cluster manager will be consulted. Other required activities, such as calibration of the radar, acquisition of the image, and subsequent science processing are all planned by CASPER.

[1] For the spacecraft control domain we are envisioning an update rate on the order of tens of seconds real time.

6. Supporting Software: Observation Planning

The Observation Planner (OP) interfaces to flight software that automatically determines the current spacecraft position and orbit. This software uses GPS signals to very accurately pinpoint the location and velocity of the TS-21 constellation. Onboard, the OP has a potential observation target list. Periodically, it takes the current orbit solution and simulates forward to predict over flights for the next 15 days of each and every potential target. These over flights are then used by CASPER as observation opportunities when planning future science observations.

7. ASE and Multi-Agent Systems

While Techsat-21 is a multi-spacecraft constellation, ASE is not a multi-agent system. In the ASE architecture, the constellation is treated as a single agent, with each of the spacecraft being a redundant subsystem. On one spacecraft, the "master" spacecraft, CASPER is running in order to perform the planning (and replanning) for the entire constellation of three spacecraft. The plans developed on the "master" spacecraft are sent on to the other two "slave" spacecraft. Because of this architecture there is no decentralized coordination problem. While there is significant interest in multi-agent coordinating spacecraft at NASA [17, 18], for the Techsat-21 mission, use of a multi-agent, distributed architecture was viewed as too risky for flight at this time.

8. Related Work and Conclusions

In 1999, the Remote Agent experiment (RAX) [11] executed for a few days onboard the NASA Deep Space One mission. RAX is an example of a classic three-tiered architecture [16], as is ASE. RAX demonstrated a batch onboard planning capability (as opposed to ASE's continuous planning) and RAX did not demonstrate onboard science. RAX also included an earlier version of the Livingstone and Burton mode identification and fault recovery software. PROBA[13] is a European Space Agency (ESA) mission that will be demonstrating onboard autonomy and launches 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 [6]. The 3CS mission is scheduled for launch in late 2003. 3CS will use onboard science data validation,

replanning, robust execution, and multiple model-based anomaly detection. The 3CS mission is considerably less complex than Techsat-21 but still represents an important step in the integration and flight of onboard autonomy software.

ASE will fly on the Techsat-21 mission will demonstrate an integrated autonomous mission using onboard science analysis, replanning, robust execution, model-based estimation and control, and formation flying. ASE will perform intelligent science data selection that will lead to a reduction in data downlink. In addition, the ASE experiment will increase science return through autonomous retargeting. Demonstration of these capabilities in onboard the Techsat-21 constellation mission 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 cost.

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.

9. 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] M.C. Burl and D. Lucchetti, "Autonomous Visual Discovery", *SPIE Aerosense Conference on Data Mining and Knowledge Discovery,* (Orlando, FL), April 2000.

[3] M.C. Burl, W.J. Merline, E.B. Bierhaus, W. Colwell, C.R. Chapman, "Automated Detection of Craters and Other Geological Features *Intl Symp Artificial Intelligence Robotics & Automation in Space*, Montreal, Canada, June 2001

[4] S. Chien, G. Rabideau, R. Knight, R. Sherwood, B. Engelhardt, D. Mutz, T. Estlin, B. Smith, F. Fisher, T. Barrett, G. Stebbins, D. Tran, "ASPEN - Automating Space Mission Operations using Automated Planning and Scheduling," *SpaceOps*, Toulouse, France, June 2000.

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

[6] 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," *Intl Symposium on Artificial Intelligence Robotics and Automation in Space*, Montreal, Canada, June 2001

[7] Interface & Control Systems, Spacecraft Command Language, http://www.sclrules.com.

[8] A. Jonsson, P. Morris, N. Muscettola, K. Rajan, and B. Smith, "Planning in Interplanetary Space: Theory and Practice," *Procs 5th Int Conference on Artificial Intelligence Planning Systems, Breckenridge*, CO, April 2000.

[9] J. Kurien and P.P. Nayak, "Back to the Future for Consistency-based Trajectory Tracking," *Proc. National Conference on Artificial Intelligence*, Austin, TX, 2000.

[10] G. Rabideau, R. Knight, S. Chien, A. Fukunaga, A. Govindjee, "Iterative Repair Planning for Spacecraft Operations in the ASPEN System," *Intl Symp on Artificial Intelligence Robotics and Automation in Space*, Noordwijk, The Netherlands, June 1999.

[11] rax.arc.nasa.gov, Remote Agent Experiment Home Page.

[12] D. Surka, J. Mueller, M. Brito, B. Urea, M Campbell, "Agent-based Control of Multiple Satellite Formation Flying," *Intl Symposium on Artificial Intelligence Robotics and Automation in Space*, Montreal, Canada, June 2001

[13] The PROBA Onboard Autonomy Platform, http://www.estec.esa.nl/proba/

[14] B.C. Williams and P.P. Nayak, "A Model-based Approach to Reactive Self-Configuring Systems," *Proceedings of the National Conference on Artificial Intelligence*, Portland, Oregon, 1996.

[15] B.C. Williams and P.P. Nayak, "A Reactive Planner for a Model-based Executive," *Proc. International Joint Conference on Artificial Intelligence*, Nagoya, Japan, 1997.

[16] R. P. Bonasso, R. J. Firby, E. Gat, David Kortenkamp, D. Miller, and M. Slack, Experiences with Architecture for Intelligent, Reactive Agents, Journal of Experimental and Theoretical Artificial Intelligence , Vol. 9, No. 1, 1997.

[17] S. Chien, A. Barrett, T. Estlin, and G. Rabideau, A Comparison of Coordinate Planning Methods for Cooperating Rovers, Fourth International Conference on Autonomous Agents (Agents 2000), Barcelona, Spain, June 2000.

[18] B. Clement, A. Barrett, E. Durfee, G. Rabideau, "Using Abstraction to Coordinate Multiple Robotic

Spacecraft," Proceedings of the 2001 Intelligent Robots and Systems Conference, Maui, HI, November 2001.