A Demonstration of Robust Planning, Scheduling and Execution for the Techsat-21 Autonomous Sciencecraft Constellation

Steve Chien, Rob Sherwood, Michael Burl, Russell Knight, Gregg Rabideau[1],
Barbara Engelhardt, Ashley Davies, Rebecca Castano, Tim Stough, Joe Roden [1]
Paul Zetocha, Ross Wainwright, Pete Klupar, [2]
Pat Cappelaere, Jim Van Gaasbeck, [3]
Derek Surka, Margarita Brito, [4]
Brian Williams, Mitch Ingham[5]

[1]Jet Propulsion Laboratory, California Institute of Technology
[2]Air Force Research Laboratory
[3]Interface & Control Systems
[4]Princeton Satellite Systems
[5]Space Systems Laboratory, Massachusetts Institute of Technology

Abstract. The Autonomous Sciencecraft Constellation (ASC) will fly onboard the Air Force's TechSat-21 constellation (scheduled for launch in 2004). ASC will use onboard science analysis, replanning, robust execution, model-based estimation and control, and formation flying to radically increase science return by enabling intelligent downlink selection and autonomous retargeting. These capabilities will enable tremendous new science that would be unreachable without this technology. We offer a demonstration of the planning, scheduling, and execution framework used in ASC.

1 Context

Robust planning, scheduling, and execution in a real environment is a challenging task. In general, each of these elements is difficult in their own right, and the fusion of these can be equally challenging. We offer a demonstration of the integration of each of these in the real task of operating the Techsat-21 (TS21) constellation of sciencecraft [4]. Our demonstration includes nominal operations as well as operations with anomalies. Our system is capable of generating its own science goals based on previous information-gathering activities. In general, the system as a whole provides considerable autonomy. The rest of this document describes the specific mission and its background, as well as our approach to addressing the challenges we face in flying such a mission. [·]

TechSat-21 is scheduled for a late 2004 launch and will fly three satellites in a near circular orbit at an altitude of 600 Km. The primary mission is one-year in length with the possibility for an extended mission of one or more additional years. One of the objectives of TechSat-21 is to demonstrate advanced radar systems. 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.

[·] 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. Other portions were performed at the Massachusetts Institute of Technology Space Systems and Artificial Intelligence Laboratories, under contracts from AFOSR and the DARPA Mobies program.

OSE was chosen because it is message passing and thus suitable for distributed applications. Each satellite will have 128 Mbytes of SDRAM as well as considerable (Gigabytes) of disk storage

2 Autonomy Technologies and Scenario

The ASC onboard flight software includes several autonomy software components:

1. *Onboard science algorithms* [1] that will analyze the image data, generate derived science products, and detect trigger conditions such as science events, “interesting” features, and change relative to previous observations
2. *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.
3. *Robust execution management software* using the Spacecraft Command Language (SCL) package to enable event-driven processing and low-level autonomy
4. The Continuous Activity Planning, Scheduling, and Replanning (CASPER) *planner* that will replan activities, including downlink, based on science observations in the previous orbit cycles
5. The ObjectAgent and TeamAgent *cluster management software* will enable the three Techsat-21 spacecraft to autonomously perform maneuvers and high precision formation flying to form a single virtual instrument

We will demonstrate ASC – specifically autonomous recognition of science events and response including planning and execution. For example, ASC will monitor lava flows in Hawaii and respond as follows:

1. Initially, ASC 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 the volcano with its radar.
4. The *Onboard Science Software* compares the new image with previous image and detects that the lava field has changed due to new flow. Based on this change the science software generates a goal to acquire a new high-resolution image of an area centered on the new flow.
5. The addition of this goal to the current goal set triggers the CASPER planner 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 ObjectAgent to plan required slews and/or maneuvers.
6. 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. And ObjectAgent and TeamAgent execute maneuvers planned by CASPER and requested at run-time by SCL.
7. Based on the science priority, imagery of identified “new flow” areas; are downlinked. This science priority could have been determined at the original event detection based on subsequent onboard science analysis of the new image.

As demonstrated by this scenario, onboard science processing and spacecraft autonomy enable the focus of mission resources onto science events so that the most interesting science data is downlinked. In this case, a large number of high priority science targets can be monitored and only the most interesting science data (during times of change and focused on the areas of change) need be downlinked.

3 Planning and Execution Framework

ASC utilizes a hierarchical architecture for planning and execution. CASPER operates at the highest level of abstraction, creating mission plans in response to high level science and engineering goals. CASPER performs traditional planning and scheduling and reasons about high level states and resources. CASPER uses local search, iterative repair, and continuous planning to respond at the 10s of seconds timescale (for further details see [2]). CASPER is being deployed in a wide range of applications including spacecraft operation, rover control, ground communications station automation, and high level control of unpiloted aerial vehicles.

Scheduled CASPER activities correspond to scripts in the Spacecraft Command Language (SCL) [10] that provide robust execution. SCL integrates procedural programming with a real-time, forward-chaining, rule-based system to provide a “smart” executive command and control function. This functionality can be used to implement retries use of alternate execution methods for robust execution as well as fault detection, isolation and recovery (FDIR). SCL is a mature software product, and has successfully flown on Clementine I and ROMPS.

Mode identification (MI) uses a declarative model to interpret sensor information to determine the configuration of the system in the presence of incomplete, noisy data. Mode reconfiguration uses these same models to determine command sequences to achieve desired configurations. 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.

Our model-based execution framework is an enhanced version of the Burton system, described in [9] and under development at the MIT AI and Space Systems laboratories. This framework incorporates the Mode Estimation capabilities of Livingstone 1 and 2, described in [5] 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.

ObjectAgent and TeamAgent [7] provide an autonomous maneuver and formation flying capability for ASC. At plan time, CASPER consults OA and TA on the feasibility and resource requirements to perform formation changes, maneuvers, and slews. At execution time, formation changes, maneuvers, and slews planned by CASPER and requested by SCL are performed by OA and TA. In this execution time function OA and TA perform closed loop control in their use of lower-level attitude and control software to achieve the desired goals.

4 Related Work and Conclusions

In 1999, the Remote Agent experiment (RAX) [6] executed for several days onboard the NASA Deep Space One mission. RAX demonstrated a batch onboard planning capability but did not demonstrate onboard science. RAX also included an earlier version of the Livingstone and Burton mode identification and fault recovery software. PROBA[8] is a European Space Agency (ESA) mission launching in 2001 that will be demonstrating onboard autonomy.

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]. Launching in 2002, 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.

ASC 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. ASC will perform intelligent science data selection that will lead to a reduction in data downlink. In addition, the ASC 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.

5 References

[1] 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

[2] S. Chien, R. Knight, A. Stechert, R. Sherwood, and G. Rabideau, “Using Iterative Repair to Improve Responsiveness of Planning and Scheduling,” *Proc Fifth Intl Conf on Artificial Intelligence Planning and Scheduling*, Breckenridge, CO, April 2000.

[3] 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

[4] S. Chien, R. Sherwood, M. Burl, R. Knight, G. Rabideau, B. Engelhardt, A. Davies, P. Zetocha, R. Wainright, P. Klupar, P. Cappelaere, D. Surka, B. Williams, R. Greeley, V. Baker, J. Doan, "The Techsat-21 Autonomous Sciencecraft Constellation", *ProcIntl Symp Artificial Intelligence Robotics and Automation in Space*, Montreal, Canada, June 2001

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

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

[7] 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

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

[9] 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.

[10] Spacecraft Command Language, www.interfacecontrol.com