MEXEC: AN ONBOARD INTEGRATED PLANNING AND EXECUTION APPROACH FOR SPACECRAFT COMMANDING

Virtual Conference 19–23 October 2020

M. Troesch[1], F. Mirza[1], K. Hughes[1], A. Rothstein-Dowden[1], R. Bocchino[1], A. Donner[1], M. Feather[1], B. Smith[1], L. Fesq[1], B. Barker[1], and B. Campuzano[1]

[1]Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA 91109, USA, E-mail: martina.i.troesch@jpl.nasa.gov

Appeared in Troesch, M.; Mirza, F.; Hughes, K.; Rothstein-Dowden, A.; Bocchino, R.; Donner, A.; Feather, M.; Smith, B.; Fesq, L.; Barker, B.; and Campuzano, B. MEXEC: An Onboard Integrated Planning and Execution Approach for Spacecraft Commanding. In *Workshop on Integrated Execution (IntEx) / Goal Reasoning (GR), International Conference on Automated Planning and Scheduling (ICAPS IntEx/GR 2020)*, October 2020.

EXTENDED ABSTRACT

The prevalent form of spacecraft commanding is through the use of sequences. Sequences generally define a sequence of commands to execute at absolute or relative times, but they contain no information about why a specific command is needed or why a command is scheduled in a particular way to achieve a desired effect [1]. Although some sequences allow for control logic, such as if statements or loops, and some even allow for event driven responses [2], there is limited flexibility in how sequences are executed. For example, sequencing languages such as the Virtual Machine Language (VML) can provide higher-level programming capabilities compared to traditional sequences. This was demonstrated successfully on Spitzer [3], where less conservative schedules could be generated by taking advantage of relative timing instead of conservative estimates, which allowed pre-identified observations to be added from a list whenever possible. However, the resulting execution is still based on a sequence and no projection or re-planning is performed. This inhibits the autonomous behavior that can be achieved.

When unexpected states or events happen, the spacecraft response is often to go into safe mode, which may prevent the spacecraft from performing tasks that could have been safe to execute [1]. This means scheduled science gains are potentially lost.

An alternative approach is to use task- or goal-based planning and commanding on board to increase autonomous behavior by maintaining intentions and effects on board the spacecraft.

A move toward more intelligent, autonomous systems through goal-based commanding instead of sequence-based commanding was seen on Remote Agent, which was demonstrated on Deep Space One (DS1) [4]. Remote Agent flew on DS1 for 48 hours and used model-based programming and onboard search with goal-based, closed-loop commanding to achieve more autonomous behavior.

The Autonomous Sciencecraft Experiment (ASE) on Earth Observing One (EO-1) [5] on goals with robust execution by responding to events and anomalies at execution time. ASE used the Continuous Activity Scheduling Planning Execution and Replanning (CASPER) [6] software to perform onboard planning and communicate with an execution system. ASE flew for over 12 years [7], aiming to maximize science return by taking data, processing it to create new goals, and re-planning on board.

The idea of a spacecraft executive has been used in other software on many other missions, such as the Spacecraft Commanding Language (SCL) on ASE [5] and TACSAT-3 [8], Remote Agent Executive on DS1 [9], and VML on Spitzer and Dawn [2].

A goal can be described through tasks, which model spacecraft behavior and constraints, where the behavior is the expected change of state by executing the task and the constraints are the states that are required to successfully carry out the task. A set of tasks makes up a task network. The tasks are activities to be accomplished and the modeled behavior and constraints on each task enforce establishment and protection of required states. An onboard planner can use a task network with the most up-to-date state information to generate conflict-free schedules and effectively use time and resources, while an executive can perform real-time constraint checking on executing tasks. If any unexpected events occur, the planner can re-plan a new schedule, while the executive can con-

tinue to execute any tasks that are safe, based on constraints.

We follow this approach in MEXEC (Multi-mission EXECutive), a multi-mission, goal-based, onboard, integrated planning and execution software that uses task networks [10]. With MEXEC, a goal is specified, maintaining intent through a network of tasks, so it is possible to increase science return and improve robustness compared to sequences, as well as respond to anomalies without safing the spacecraft. MEXEC shares many characteristics with the planner and executive on Remote Agent as well as CASPER on ASE, such as having a separate planner and executive and performing constraint-based, periodic planning in a limited scheduling window. MEXEC also inherits the use of a commit window from CASPER. However, one of the major differences is that MEXEC provides a more consistent representation of behavior modeling at planning and execution time and has a tighter coupling between the planner and executive. Looking ahead to future missions, the M2020 Onboard Planner [11] and MEXEC have many similarities. In fact, they use the same timeline library to search for valid intervals to place tasks during planning. Both use a planner to schedule tasks and an executive to perform real-time constraint checking. However, MEXEC is multi-mission, whereas the Onboard Planner has a specialized planning algorithm with limited choice points or options during scheduling including what states can be represented in the timeline library.

Although increased autonomy is an enabler for future missions where human-in-the-loop commanding is not possible, it is important to methodically prove out autonomous capabilities and allow users to gain confidence in new software. To that end, we have developed MEXEC in a systematic way to be flight-ready, including performing incremental flight and testbed experiments on the Arcsecond Space Telescope Enabling Research in Astrophysics (ASTERIA) CubeSat.

ASTERIA was a 6U CubeSat deveoped as a collaboration between the Jet Propulsion Laboratory and the Massachusetts Institute of Technology. The flight software was built in F' [12] and nominal operations were performed with sequences. It was deployed from the International Space Station in 2017 for a 90-day prime mission to demonstrate precision photometry technology and take opportunistic science. It successfully achieved all of its primary mission goals [13]. ASTERIA had three extended missions, the third of which focused on using ASTERIA as a “testbed in the sky” for technology demonstrations [14]. As part of those demonstrations, MEXEC was flown to demonstrate nominal science opera1tions using task networks. A second in-flight experiment was planned with a more complex goal of managing momentum, but was converted to a testbed-only demonstration when the spacecraft stopped communicating [15].

In this paper we describe MEXEC and the two experiments that were performed, which show that MEXEC can be integrated into existing flight software and be used for operations, as well as enable robustness and fail-operational commanding.

Acknowledgement

The research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004).

References

[1] Gat, E., and Pell, B. 1998. Smart executives for autonomous spacecraft. IEEE Intelligent Systems and their Applications 13(5):56–61.

[2] Grasso, C., and Lock, P. 2008. VML sequencing: Growing capabilities over multiple missions. In SpaceOps 2008 Conference, 3295.

[3] Peer, S., and Grasso, C. A. 2005. Spitzer Space Telescope Use of the Virtual Machine Language. In 2005 IEEE Aerospace Conference, 1–10. IEEE.

[4] Muscettola, N.; Nayak, P. P.; Pell, B.; and Williams, B. C. 1998. Remote agent: To boldly go where no AI system has gone before. Artificial intelligence 103(1-2):5–47.

[5] Chien, S.; Sherwood, R.; Tran, D.; Cichy, B.; Rabideau, G.; Castano, R.; Davis, A.; Mandl, D.; Frye, S.; Trout, B.; et al. 2005. Using autonomy flight software to improve science return on Earth Observing One. Journal of Aerospace Computing, Information, and Communication 2(4):196–216.

[6] Chien, S.; Knight, R.; Stechert, A.; Sherwood, R.; and Rabideau, G. 2000. Using Iterative Repair to Improve Responsiveness of Planning and Scheduling. In International Conference on Artificial Intelligence Planning Systems (AIPS 2000).

[7] Ellis, E. G. 2017. The Journey of NASA’s Smartest Satellite Finally Comes to an End. https://www.wired.com/2017/ 03/say-farewell-eo-1-nasas-smartest-satellite. [Online; accessed February 25, 2020; posted March 24, 2017].

[8] Mackey, R.; Brownston, L.; Castle, J. P.; and

Sweet, A. 2010. Getting diagnostic reasoning off the ground: maturing technology with TacSat-3. IEEE intelligent systems 25(5):27–35.

[9] Pell, B.; Gat, E.; Keesing, R.; Muscettola, N.; and Smith, B. 1997. Robust periodic planning and execution for autonomous spacecraft. In Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI), 1234–1239.

[10] Verma, V.; Gaines, D.; Rabideau, G.; Schaffer, S.; and Joshi, R. 2017. Autonomous Science Restart for the Planned Europa Mission with Lightweight Planning and Execution. In Proceedings of the 10th International Workshop on Planning and Scheduling for Space (IWPSS2017). Pittsburgh, PA.

[11] Rabideau, G., and Benowitz, E. 2017. Prototyping an onboard scheduler for the mars 2020 rover. In International Workshop on Planning and Scheduling for Space.

[12] Bocchino, R.; Canham, T.; Watney, G.; Reder, L.; and Levison, J. 2018. F Prime: an open-source framework for small-scale flight software systems. In Proceedings of the AIAA/USU Conference on Small Satellites. Advanced Technologies II, SSC18-XII-328. https://digitalcommons.usu.edu/smallsat/2018/all2018/328/.

[13] Smith, M.; Donner, A.; Knapp, M.; Pong, C.; Smith, C.; Luu, J.; Pasquale, P.; and Campuzano, B. 2018. On-Orbit Results and Lessons Learned from the ASTERIA Space Telescope Mission. In Proceedings of the AIAA/USU Conference on Small Satellites. The Year in Review, SSC18-I-255. https://digitalcommons.usu.edu/smallsat/2018/all2018/255/.

[14] Fesq, L.; Beauchamp, P.; Donner, A.; Bocchino, R.; Kennedy, B.; Mirza, F.; Mohan, S.; Sternberg, D.; Smith, M. W.; and Troesch, M. 2019. Extended Mission Technology Demonstrations Using the ASTERIA Spacecraft. In 2019 IEEE Aerospace Conference, 1–11. IEEE.

[15] Jet Propulsion Laboratory, California Institute of Technology. 2020. Tiny satellite for studying distant planets goes quiet. https://www.jpl.nasa.gov/news/news.php? feature=7568. [Online; accessed January 3, 2020; posted January 3, 2020].