

# **Executive/Decision Making Systems for Autonomous Vehicles: Related Work**

By: Connor Lange

## **2. Related Work:**

There have been a number of systems in the literature that share some of the same functionality as the Generic Decision Making Framework for Autonomous Systems (GDMFAS). Much of the executive system research in the area of autonomous space vehicles has focused on scheduling and planning.

### **2.1 Groundstation Approach:**

The first executive systems designed for spacecraft appeared in the early 1990's. Unlike GDMFAS, these executive systems were in control of the groundstation instead of the spacecraft. While this approach replaces expensive groundstation personnel, it doesn't assist the spacecraft in making real-time decisions. Three such systems are GENIE [1], LOGOS [2], and more recently, ASOF [3].

GENIE, a system created in 1993 by NASA Goddard Space Flight Center, was the original groundstation-based executive and only marginally succeeded at replacing a groundstation operator. The GENIE system had limited intelligence and required a fair amount of human intervention to solve issues that occurred during execution. The system had multiple operation modes, including a shadow mode where GENIE wouldn't execute the commands, but would display them instead, but was closer to a prototype than a full replacement for a groundstation operator.

NASA Goddard's next attempt at groundstation automation wasn't until 1995 with the LOGOS system. LOGOS added additional groundstation capabilities but still utilized GENIE for spacecraft command tasks. LOGOS new features included, among other things, health monitoring and error correction. Although LOGOS was closer to a full human

replacement, command execution still suffered the same defects as GENIE due to GENIE's integration in the system.

The most recent approach is the ASOF system. ASOF was developed in 2010 as a Cal Poly Masters thesis, and takes an agent-based approach to executive system functionality. Like LOGOS, ASOF is a framework that allows agents to be deployed to achieve specific, user-defined functionality. However, ASOF doesn't define its agent interface as rigidly as LOGOS does, which allows ASOF to be more extensible. Further, ASOF doesn't rely on a previous system which increases its flexibility and (in this case), its reliability. Since the author of ASOF, is familiar with what has and hasn't worked in the past, he was able to design a system from the ground up that meets the needs of groundstation ops, but doesn't integrate a flawed system into the architecture, making the end result more effective.

## **2.2 Platform Independent Approaches**

With the faith in autonomous systems increasing, newer systems were developed. ASPEN [4], deployed by JPL in 2001, is a planning/scheduling system that facilitates the execution of tasks by decomposing mission objectives into smaller activities that the system can actually execute. Although ASPEN was originally designed for groundstation deployment, its platform independent implementation allows it to easily be deployed on spacecraft as well; an ASPEN prototype was deployed on the EO-1 spacecraft in late 2000 [5]. Despite the benefits that ASPEN offers, a drawback of the system is its reliance on a modeling language. While a modeling language may be easier for humans, it can lack expressiveness in certain situations and make the execution of tasks more complex.

## **2.2 On-board Approaches**

Executive system approaches that are deployed on-board spacecraft are the systems that are most similar to GDMFAS. There are a variety of such systems in the literature, but three systems capture the ideology behind the GDMFAS design. The three systems, NEAT [6], CASPER [7], and the NMRA [8], aim to accurately model the spacecraft status and make decisions based on sensor data and previous states.

While NEAT is more similar to the Scheduler module in GDMFAS, many design decisions present in NEAT apply to the overall system design of GDMFAS. Like other scheduling/planning systems, the Near-optimal Evolutionary Autonomous Task manager

(NEAT) primarily focuses on scheduling tasks and executing them. However, NEAT also factors in the resource usage of the tasks it schedules. Since resource usage can change depending on what else is occurring in the system, NEAT models resource usage of executing tasks using a neural network. When a task finishes execution, NEAT updates its resource usage data. In addition to providing adaptability in the scheduler, NEAT attempts to calculate an optimal schedule. To this end, NEAT calculates a complete schedule, but only executes the first task; the rest of the schedule is thrown away and the schedule is recalculated. This behavior ensures maximum adaptability and provides a near-optimal scheduling solution.

The next on-board system, CASPER, was developed by JPL as a scheduling/planning module. Unlike NEAT, CASPER was designed to be used as part of a larger system. CASPER has been used in a variety of systems, including the Autonomous Spacecraft Experiment (ASE), aboard the EO-1 spacecraft. Although CASPER doesn't throw away its schedule immediately after executing a task, it is still adaptable due to its utilization of replanning. If the state of the spacecraft, or part of the spacecraft, changes CASPER has the option to correct for the change and then resume a (potentially modified) schedule. In many respects, CASPER is a better system than NEAT. Unlike NEAT, CASPER is modular, it doesn't throw away its schedule (leading to less frequent schedule calculations), and is more abstract.

The final on-board system, the New Millennium Remote Agent architecture (NMRA), is by far the closest to GDMFAS in terms of system design and available functionality. The NMRA was designed by JPL and was deployed on the Deep Space 1 (DS-1) satellite in 1998. Like CASPER and NEAT, the NMRA includes planning and scheduling functionality. The NMRA also includes an executive and a controls module. Like GDMFAS, the executive module present in the NMRA has the primary responsibility of managing tasks and system state at run-time. Although the NMRA considers the current state of system resources when executing a task, it does not model resource usage or attempt to update predicted resource usage based on past behavior. In addition to resource management, the NMRA's executive module is responsible for invoking other modules (such as the planner) and monitoring their state. This behavior is extremely similar to that of the GDMFAS system. Although it is included in the planning module, the NMRA contains a form of the Task Manager module found in the GDMFAS system. Although the task

manager is merely a temporal database, the submodule stores various parameters about tasks in the system. For example, the database keeps track of the deadlines of different tasks to allow for simpler planning phases. While the NMRA is the system that is the most similar to GDMFAS, the NMRA falls short with respect to intelligence and execution monitoring. Even though GDMFAS provides a more accurate implementation of execution monitoring and resource utilization, it isn't without trade-offs. However, the added functionality and improved task resolution make GDMFAS task execution more expensive than the NMRA.

### Sources:

- 1) J. Hartley, E. Luczak, and D. Stump, "Spacecraft control center automation using the generic inferential executor (genie)," in In International Symposium on Space Mission Operations & Ground Data Systems, ser. SpaceOps '96, 4th, 1996, pp. 1007–1014.
- 2) W. Truskowski, H. Hallock, and J. Kurien, "Agent technology from a nasa perspective," in In Proceedings of the Third International Workshop on Cooperative Information Agents III. London, UK: Springer-Verlag, 1999, pp. 1–33.
- 3) J. Anderson, Autonomous Satellite Operations for CubeSat Satellites. Cal Poly State University:SLO (Master's Thesis), 2010.
- 4) S. Chien, G. Rabideau, R. Knight, R. Sherwood, B. Engelhardt, D. Mutz, T. Estlin, B. Smith, F. Fisher, T. Barrett, G. Stebbins, and D. Tran, "Aspen - automated planning and scheduling for space mission operations," in Space Ops, 2000.
- 5) R. Sherwood, A. Govindjee, D. Yan, G. Rabideau, S. Chien, and A. Fukunaga, "Using ASPEN to automate EO-1 activity planning," in Aerospace Conference, 1998 IEEE.
- 6) A. Carrol and P. Palmer, "Adaptive resource modelling for autonomous planning and scheduling," in IWPSS Proceedings, 2006.
- 7) S. Chien, R. Sherwood, D. Tran, B. Cichy, G. Rabideau, R. Castano, A. Davies, R. Lee, D. Mandl, S. Frye, B. Trout, J. Hengemihle, J. D'Agostino, S. Shulman, S. Ungar, T. Brakke, D. Boyer, J. Van Gaasbeck, R. Greeley, T. Doggett, V. Baker, J. Dohm, and F. Ip, "The eo-1 autonomous science agent," in Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems - Volume 1, ser. AAMAS '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 420–427. [Online]. Available: <http://dx.doi.org/10.1109/AAMAS.2004.262>

8) B. Pell, D. E. Bernard, S. A. Chien, E. Gat, N. Muscettola, P. P. Nayak, M. D. Wagner, and B. C. Williams, "An autonomous spacecraft agent prototype," in Proceedings of the first international conference on Autonomous agents, ser. AGENTS '97. New York, NY, USA: ACM, 1997, pp. 253–261. [Online]. Available: <http://doi.acm.org/10.1145/267658.267724>