

Assurance of Autonomy for Robotic Space Missions

Held on August 14- 15, 2018, at California Institute of Technology

Summary by Martin Feather, Jet Propulsion Laboratory, California Institute of Technology



Outside the meeting room
Photograph by Martin Feather



Inside the room, listening to a comment
Photograph by Martha Wetherholt

Contents

Purpose	2
Participation.....	2
Activities.....	2
Prior and current uses of autonomy on missions & tech demos.....	3
Brian Williams (MIT) – Assuring Autonomous Health Management on Deep Space One	3
Steve Chien (JPL) – the Autonomous Sciencecraft Experiment (ASE)	3
Elyse Fosse (JPL) – Mars2020 Simple Planner – Assurance of Autonomy for Robotic Space Missions 3	
Raymond Francis (JPL) – Integrating AEGIS intelligent targeting into the Mars Science Laboratory Mission.....	3
Emmanuel Lesser (ESA) – Software Product Assurance for Autonomy on-board Spacecraft.....	3
Assurance challenges	4
Michael Aguilar – Automation, TRL, and Trust	4
Gerek Whitman and Kenneth Costello (NASA IV&V) – IV&V of Autonomy.....	4
Keisuke Sugawara (JAXA) – JAXA’s activity of assurance for autonomous spacecraft	4
Three challenging autonomous mission concepts.....	4
Distant Kuiper Belt Object (KBO) fast flyby.....	5
Lava tube cave exploration	5
Robotic Moon base	5
Related studies.....	6
Guillaume Brat (NASA Ames) – Assurance for Autonomy in Aviation.....	6
Gabor Karsai (Vanderbilt University) – Assurance-based Learning-enabled Cyber-Physical Systems .6	
Brian Williams (MIT) – Risk-bounded Architectures for Exploring Europa.....	6
System and software engineering	7
Madeline Diep (Fraunhofer) – Modeling Requirements for Autonomy	7
Leila Meshkat (JPL) – Autonomy in space – is it risky?	7

Relevant assurance methods	7
Rick Kuhn (NIST) – Combinatorial Methods in Software Assurance & Testing	7
Connie Heitmeyer (Naval Research Lab)	7
Mats Heimdahl (University of Minnesota) – Formal Methods: Academic and Industry Perspective ..	7
Klaus Havelund (JPL) – Formal Methods @ JPL (and NASA Ames)	7
Next Steps	8
Post-it® notes (large and small)	8
Impressions	10
Attendees.....	14
Acknowledgements.....	15

Purpose

The original statement of purpose was as follows:

While there have been meetings on assurance of autonomy in other application areas, this meeting seeks to develop a roadmap for assurance of autonomy specifically for robotic space missions. Key characteristics that distinguish this application area include: the lack of detailed prior knowledge of the environments in which those missions are to operate; the challenges of mimicking deep space operating conditions for purposes of testing; the need to be highly assured of failsafe operation; limited and delayed (due to speed of light over solar system distances) communication; and the one/few-of-a-kind, must-work-the-first-time nature of most space missions.

It turned out to be a trifle optimistic to expect to emerge with a roadmap during the first meeting of this kind. The meeting comprised a series of presentations covered successes and challenges of infusing various forms of autonomy into space and underwater missions, and the kinds of approaches to assuring autonomy software and systems.

Participation

The meeting brought together a total of 38 participants (see the Attendees section for complete list) to provide a mix from the following (overlapping) areas:

- Developers of autonomy – primarily for space missions; for (terrestrial) underwater exploration (which shares most of the key characteristics listed in the purpose statement), and aviation.
- “Assurers” (testers, V&Vers, IV&Vers) of autonomy and other critical forms of software.
- Practitioners as well as researchers.
- Space agencies, universities, research facilities and other institutions, and industry.

Activities

The following presentations occurred, with plentiful discussion taking place in the course of them. As a result, the originally planned agenda sessions set aside for “Discussion” generally got squeezed out.

Prior and current uses of autonomy on missions & tech demos

Brian Williams (MIT) – Assuring Autonomous Health Management on Deep Space One

In 1999 The Deep Space 1 mission, a technology demonstration spacecraft, flew and exercised the “Remote Agent” software as an experiment. Brian’s presentation covered the Remote Agent’s goal-based commanding and model-based diagnosis, with an emphasis on the Livingstone technology for model-based diagnosis. Brian’s successors to Livingstone, while they have not seen use on subsequent spacecraft, have been employed on autonomous under water vehicle missions (see his presentation on the second day)

Steve Chien (JPL) – the Autonomous Sciencecraft Experiment (ASE)

Steve’s presentation described the Autonomous Sciencecraft Experiment, which operated the Earth Observing-1 spacecraft from 2004 onwards until the spacecraft was decommissioned over a decade later. *“The ASE software uses onboard continuous planning, robust task and goal-based execution, and onboard machine learning and pattern recognition to radically increase science return by enabling intelligent downlink selection and autonomous retargeting.”* [<https://ase.jpl.nasa.gov/>]. Steve described this very successful long-term operation, and its “value proposition” as indication of its savings in operations costs as well as its capacity to detect and respond to science events (e.g., volcanic eruptions).

Elyse Fosse (JPL) – Mars2020 Simple Planner – Assurance of Autonomy for Robotic Space Missions

Elyse’s presentation described how the “Simple Planner” is designed to help the Mars2020 rover accomplish its scientific objectives. Its purpose is to reduce the “ground-in-the-loop” cycle, and to be “opportunistic” – i.e., take advantage of availability of resources (e.g., time) to do more, enabled by the rover’s awareness of actual conditions. Its introduction is following a phased approach, beginning with “paper prototypes” to inform the science and operations personnel, simulations, prototyping, reviews, end-to-end and other kinds of testing (with more planned). While the rover system provides safeguards (e.g., for low voltage), the simple planner itself avoids inadvertently triggering these.

Raymond Francis (JPL) – Integrating AEGIS intelligent targeting into the Mars Science Laboratory Mission

Raymond described AEGIS, “Automated Exploration for Gathering Increased Science”. This on-board software performs autonomous target selection to gather additional science. On Curiosity, it controls the ChemCam’s camera and laser; as it does so it must keep safe, i.e., never point the camera at the sun, and never fire the laser at the rover. His presentation covered the steps taken to successfully infuse AEGIS onto Curiosity, including a stepwise introduction process, its own safety checks (even though the rover does its own checks), its bounded execution, and robustness to postulated failures, and testing at many levels.

Emmanuel Lesser (ESA) – Software Product Assurance for Autonomy on-board Spacecraft

Emmanuel described an ESA study done *in the early 2000’s* [Blanquart, J-P., S. Fleury, M. Hernek, C. Honvault, F. Ingrand, J-C. Poncet, D. Powell, N. Strady-Lécubin, and P. Thévenod. "Software Product Assurance for Autonomy On-Board Spacecraft." In DASIA 2003-Data Systems In Aerospace, vol. 532. 2003]. This early study foresaw the desirability of autonomous space systems, together with the challenges of their assurance. For AI-based systems key categories of hazards were identified, along with

corresponding approaches to assuring the dependability of AI-based systems against these hazards. For example, formal methods and improvements to testing [see also the “Relevant assurance methods” session on the second day] were called for, and the architectural concept of a “Safety Bag” to protect from commands that would compromise system safety. Emmanuel also mentioned possibilities for augmenting an existing standard (ECSS-Q-ST-80C: “Space product assurance: Software product assurance”) with specific requirements for autonomous systems.

Assurance challenges

Michael Aguilar – Automation, TRL, and Trust

Michael’s presentation showed the NASA Technology Readiness Level (TRL) definitions, couched in hardware terminology, and suggested the steps it would take for a software the effort to go from TRL3 to TRL7, taking advantage of current-day capabilities. He suggested NASA could help by making engineering/operations data readily available (from flight, and from testing) and thereby support *Using actual recorded engineering telemetry to develop and verify autonomous systems could meet a “high fidelity software test”*. [Audience comments indicated the raw data needs to be annotated and correlated, and engineering models provide to understand the data].

Michael went on to identify several failures due to incorrect configuration settings, and more generally the challenge of assuring rule based systems, for which we lack metrics of coverage of the rules themselves (as contrasted with metrics of code coverage).

Gerek Whitman and Kenneth Costello (NASA IV&V) – IV&V of Autonomy

IV&V’s overall approach is to identify the risks/concerns (by looking at the mission’s functional capabilities and architecture), prioritize them, decide the assurance strategy they will follow to address them, and then apply their experience, tools and techniques to fulfil that strategy. When autonomy provides mission- and safety-critical capabilities, it thus falls within scope for IV&V to assure it will execute as intended (and only as intended) and be robust to adverse conditions. Their presentation also outlined some of the challenges they foresee in assuring future mission use of autonomy.

Keisuke Sugawara (JAXA) – JAXA’s activity of assurance for autonomous spacecraft

Keisuke’s presentation considered (1) assurance of an autonomous command function (e.g., for FDIR), and (2) assurance of AI. For the former, the many possible combinations of scenarios and states makes understanding and testing their many possible interactions challenging; applying model checking (e.g., using the model checker SPIN) is identified as a means to check these interactions for safety properties. [This would be an example application of formal methods, a topic also addressed later in the day]. For the latter, use of a (non-AI) component akin to a safety monitor [called a “Safety Bag” in the study Emmanuel mentioned] would protect the system from inappropriate outputs of the AI component.

Three challenging autonomous mission concepts

The intent had been to discuss three examples of highly autonomous mission concepts¹ that would pose assurance challenges. Time allowed for only a brief outline of each of them; more detail than was described during the meeting follows:

¹ Thanks to Brent Sherwood (JPL) for permission to reproduce his list of these

Distant Kuiper Belt Object (KBO) fast flyby

Half a dozen identical spacecraft speed outward into the Kuiper Belt. They were injected together onto a fast trajectory by a heavy-lift rocket rarely used for planetary science. Each spacecraft then precisely targeted a unique Jupiter flyby; now, years later, the six trajectories encounter six widely separated Trans-Neptunian Objects – dwarf planets like Pluto but much farther out. Twenty years past Jupiter, each craft wakes up, learns its state, and tunes its flyby keyhole aim. Years later, over a period of just days, and peaking over just minutes, it speeds past its target, learning whatever it can from its only opportunity. It decides the best combination of instrument priority, cadence, measurement bands, sensitivity, precision, pointing, storage allocation, and transmit priority based on features it detects and understands upon approach: features that get increasingly clear, but at an accelerating rate. After the encounter, the spacecraft transmits data to Earth, in the priority order it decides.

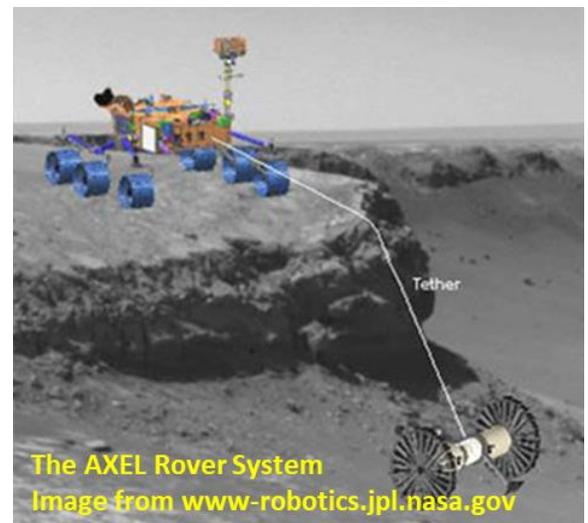
Autonomy challenges: quick analysis of emergent science observations to build, V&V, execute, and report the most important results from a sequence of actions that extracts the best science value from a moment in time.

NOTE: Steve Chien drew my attention to a former study of this kind of mission concept:

<https://ai.jpl.nasa.gov/public/projects/agile-science/>

Lava tube cave exploration

AXEL rappels into a cave skylight from a Mars lander a safe distance away from the pit's lip. It interrogates the pit wall microscopically as it descends and then, dangling on its tether in the void, images the cave system around the collapse pit. Finally setting down on the rubble pile at the bottom, it reconnoiters the site, then climbs to the peak of the pile and releases its little ones: Cave Worms that crawl out and up along the ceiling to explore the dark reaches of the lava tube in both directions. From its vantage point, AXEL receives findings transmitted by all the Worms – multi-sensor readings of location, scene, environment, mineralogy, chemistry, astrobiology – and sends the most important ones up to the lander that then relays them off Mars.



Autonomy challenges: coordinated behavior, awareness of its own health, manage fractionated sensors in an unstructured, emergent environment, mapping it and pursuing intriguing features, from comparative volcanic geology on the Moon to potential biomarkers and extant astrobiology on Mars.

Robotic Moon base

The first Moon base provides several functions: • **outpost** that extends international human space flight operations onto the surface of another planet • **laboratory** for experimenting in the lunar environment, with native materials, and on samples retrieved from all over the Moon • **factory** that demonstrates production and use of resources, starting with water and oxygen • **testbed** for learning how to expand lunar activities and reach deeper into space.

Most of the time, most of the Moonbase is robotic, a result of the relative cost of crewed missions. For years, crews will visit only episodically or periodically. In addition, robots excel for activities that are

heavy, hazardous, tedious, and routine. Therefore, mobile robots play enabling roles in three phases: 1) **building the base**, which includes surveying, grading, excavation, construction, assembly, and expansion; 2) **unoccupied base operations**, which includes productive activities like resource utilization, scientific exploration, and base growth as well as sustainment activities like monitoring, inspection, and equipment changeout; 3) **support to human crews**, which comprises all these types of activities plus novel tasks directed by the crew. One early, high-priority robot task is to restructure the native local environment, for example by grading, paving, and emplacing navigation beacons, to facilitate routine, indefinite robotic operations.

Autonomy challenges: continuous, coordinated action by multiple machines of multiple types and overlapping capabilities, in an evolving but resource-constrained planetary environment, sometimes around humans.

----- END OF FIRST DAY -----

Related studies

Guillaume Brat (NASA Ames) – Assurance for Autonomy in Aviation

Guillaume described the development of tools that do much more up-front checking of requirements, design, and architectures before code is produced. The motivation is to reduce the very expensive recertification process that aviation typically requires when making changes in response to defects discovered late in the lifecycle. [See the NASA Ames webpage for Robust Software Engineering: <https://ti.arc.nasa.gov/tech/rse/> - the Research page it links to lists several of the tools they have developed]. He also discussed the “Assurance for Autonomous Systems for Aviation Workshop” that took place in early 2016 [the report is available at: <https://ntrs.nasa.gov/search.jsp?R=20170000385>] and some of the findings (albeit somewhat high-level) that emerged. He also pointed to the “Reliable Machine Learning in the Wild – ICML 2017 Workshop” [see <https://sites.google.com/site/wildml2017icml/>] as having identified key challenges in dealing with machine learning.

Gabor Karsai (Vanderbilt University) – Assurance-based Learning-enabled Cyber-Physical Systems

Gabor summarized an ongoing DARPA Assured Autonomy program (see <https://www.darpa.mil/program/assured-autonomy>) addressing systems with “Learning-Enabled Components” (LECs). The inclusion of LECs, while desirable for autonomy’s operation in unknown and unstructured environments, considerably exacerbates the challenge of assuring safety. The DARPA program seeks to address these challenges.

Brian Williams (MIT) – Risk-bounded Architectures for Exploring Europa

Brian’s presentation covered his work on risk-bounded activity planning and scheduling involving motion planning, as might be used on a Mars rover, or the Europa Hydrobot Concept. He detailed demonstrations of his work in (terrestrial) ocean campaigns employing autonomous submersibles. The appeal of this demonstration arena is its close analogue to space missions, with similar challenges and a similar value proposition, but at much lower cost and higher frequency.

n.b., some information on this topic is on the Keck Institute for Space Studies website at:

<http://kiss.caltech.edu/techdev/systems/systems.html>

System and software engineering

Madeline Diep (Fraunhofer) – Modeling Requirements for Autonomy

Madeline's presentation considered various approaches to modeling requirements, with the purpose of catching autonomy requirements errors early. She considered several requirements engineering methodologies, with increasing levels of specificity. Starting with high level requirements (using Goal Decomposition to represent and reason about system objectives, relating this to the Vashev & Hinchey "Generic Autonomous Requirements" (GAR) work), then through an agent interaction model considering actions and information exchange, and finally finite state machines at the individual agent level.

Leila Meshkat (JPL) – Autonomy in space – is it risky?

Leila's presentation on risk made reference to the previous day's presentation by Keisuke, to explicitly provide an example of a decision within which autonomy's determination (of the identity of a vessel in an image) might reside. From this, risk, in terms of likelihood and consequence, could then be determined. She also contrasted the risks that can arise in traditional commanding of spacecraft with the (different) risks that can arise if autonomous commanding is done on-board.

Relevant assurance methods

Rick Kuhn (NIST) – Combinatorial Methods in Software Assurance & Testing

Rick described combinatorial testing methods as a way to develop efficient test suites, i.e., test suites covering interactions among the test variables' values. While exhaustive testing (of all possible input combinations) is almost always infeasible, testing to cover "t-way" combinations (e.g., if $t=2$, this would be pairwise) of values is feasible and, based on practical studies, highly effective. Rick listed several testing-related tools that NIST makes available, and encourages would-be users to email him (kuhn@nist.gov) to request them.

Connie Heitmeyer (Naval Research Lab)

[sorry, my notes on Connie's presentation are particularly skimpy]

Connie described work to analyze requirements for consistency and completeness, and for property guarantees. The approach uses a tabular notation for the formal expression of requirements, and is supported by tools to perform the analyses. There have been numerous successful real-world applications of this approach.

Mats Heimdahl (University of Minnesota) – Formal Methods: Academic and Industry Perspective

Mats related his many years of experience applying formal methods to verify a model satisfies its requirements, or at a lower level of detail, an implementation meets its specification. He considered three different ways of expressing requirements – natural language, temporal logic, and statecharts – covering how they are related, their strengths and weaknesses, and the ways their correctness can be assured.

Klaus Havelund (JPL) – Formal Methods @ JPL (and NASA Ames)

Klaus' presentation covered three somewhat complementary methods for assuring complex software:

- Static analysis, in which tools are run to analyze code for its adherence to rigorous rules of a coding standard

- Functional verification of models that drive a planner, by mapping the model into the form it can be analyzed for adherence to safety and progress properties by the model checker SPIN. Model checking was used much earlier on the Lisp code of the Executive component within DS1's Remote Agent [see the opening presentation by Williams] to find failure scenarios in some interleavings of possible behaviors.
- Runtime verification, in which an efficient run-time monitor of code can spot non-conformances as that code executes.

Next Steps

- Presentations from the meeting, if given permission from the authors, will be distributed to the attendees. *Sept. 11th 2018: I have most of these, I will distribute them shortly -- Martin*
- Another such meeting approximately a year after this first one. It can, but need not, be at the same location. *If someone would like to volunteer to host it elsewhere, please do let me know; I would be happy to repeat at this same location – Martin*
- Distribute a summary of the meeting. *Here it is! Comments for improvements/corrections welcomed – Martin*
- Over the course of the next year, we should also look for other related events at which some of us might congregate – *Everyone!*
- Set up some common site to serve as a repository of information and locus for coordination - *TBD – Martin*

The following suggestions were made for content and activities:

- Webinars/seminars to fill in the gaps
- Focus groups on specific topics (e.g., risk)
- Example problems that need to be addressed (there was mention of communities in other areas where this has worked well, and cases where it hasn't)
- Make it a platform to let interaction flow!

Post-it[®] notes (large and small)

-
- **BENEFITS**
 - Fail operational
 - Flight “pull” (when available)
 - Increased science return
 - Ops costs reduced
 - Tasking by end users (without ops)
 - Lower anomaly rate
 - Counter boredom!
 - Reduce ground in the loop cycles
-
- **POSSIBLE CONCERNS**
 - Resource usage (CPU, etc.)
 - Can the Autonomy be updated in flight?
 - Is it robust to poor inputs?
 - Can its risk be managed?
 - How to aggregate probabilities and uncertainties?

- Proof properties over a space – it doesn't matter if the nature is stochastic. Doesn't affect V&V

- CONCERNS

- V&V of rule engines
 - Rule base
 - Rule engine
 - Changing rule base
 - Conflicting rules e.g. ordering
 - Multiple rule “modalities” across large system and the way they combine
- Make it look like current practice
- Graceful degradation
- Optimize situation (biomimic)
- How respond when unsafe condition

- (untitled)

- Does something stop autonomy from an unsafe action?
 - Safety nets
 - Run-time assurance
 - Partitioning: 1. Safety functions 2. Mission functions
 - Monitors
 - Contingencies
- Can autonomy know when it is “uncertain” / know when to call for help?
- Is it OK for autonomy to not act?
- Metrics for coverage analysis for validating model completeness
- Uncertainty characterization during validation
- Can it be turned on gradually?
- Optimize the situation
- How/when can formal methods be used?

- APPLICATION AREAS

- Classification
- Planning & Scheduling
- ISHM
- Mobility
- Sampling

- TECHNOLOGIES

- Run time modeling
 - V&V for variations of may be difficult
- Learning algorithms
- Expert systems
- Neural networks
- Fuzzy logic

- Risk
- Planning/adaptation trade in design for reliability
- Models – self correcting? OODAs

- Discontinuities in state space
 - Could make V&V more difficult; not more risky
 - CFE – Risk: model
 - Local conditions require quick response
-

- [WAYS FORWARD]

- What are the hard problems that existing math can solve
 - Bring the decision makers to the table
 - Create “library” of existing work to help fill gaps
 - Set up group webinars on topics to follow up
 - People will deploy autonomy even without a good reason to trust it. What data could we collect to build that trust over time, for either a given system or a class of systems? (How would we know if part of the class really shouldn't be deployed? What are the variables that matter?)
-

- Where are the adoption challenges
 - Painpoints
 - What are the tools we can use
 - Do we have any autonomous systems out there
 - What is the value proposition of autonomous systems
-

- Get up to speed on the space aspect of autonomy
 - Commonality
 - Learn
 - Building bridges
 - Bridge pockets of expertise
 - Look for commonalities
-

- More validation of models
 - New research areas
 - How to deal with software that can change
 - Can the software system change due to hardware degradation
 - Continued engagement
-

Impressions

Here are my impressions from having had a couple of months to reflect on the meeting. I make no guarantee as to their correctness or sensibility, and certainly not as to their completeness in covering the topic in general. – Martin.

The need for a focus on “assurance of autonomy for robotic space missions”

I think there *is* the need; autonomy in general, and its assurance, is a very broad topic, while robotic space missions have characteristics that tend to distinguish them from most terrestrial applications of

autonomy (e.g., autonomous cars). While this does not preclude looking at those other application areas, I still believe it worthwhile to have our particular focus.

Nevertheless, I agree with the sentiment that (terrestrial) underwater applications have the closest parallel: communication with the vehicle(s) once underwater is very limited; the vehicles are resource constrained (e.g., power); sensing the environment is challenging; the environment itself is only partially known in advance, has influences (e.g., current) on motion, may exhibit surprises, and has worthy science objectives. These applications have the relative advantages of being lower-cost, offer more opportunities, are more risk-tolerant (although perhaps there can be human safety considerations that would not apply to robotic space missions), and with a shorter timeframe of deployment and experimentation. Brian Williams' second presentation covered this well.

Space mission autonomy we did *not* consider

Even given our limited focus, there was not time to consider every aspect of robotic space mission autonomy and its assurance. In particular, we didn't address autonomous space systems that interact with astronauts (as might be the case on the International Space Station, for example), autonomy (or at least advanced automation) of ground facilities, or (on Earth) of processing the data returned by a space mission (some of which uses machine learning capabilities). Systems that learn by themselves, as is the subject of the DARPA initiative that Gabor described to us, seem far from acceptance by space missions. We also paid little or no attention to how multiple autonomous assets might interact – with the increasing interest in smallsats and cubesats, this is not necessarily so far off.

DS1 was a demonstration success; EO-1 was a science success; AEGIS is a science success

DS1's Remote Agent (RAX) was described as a breakthrough demonstration of autonomy in space, and indeed in 1999 it actually flew (and operated, after a hiccup) RAX, an autonomous system that combined planning and scheduling with fault diagnosis with a run-time executive. E.g., "It's one small step in the history of space flight. But it was one giant leap for computer-kind, with a state of the art artificial intelligence system being given primary command of a spacecraft."

[<https://ti.arc.nasa.gov/tech/asr/groups/planning-and-scheduling/remote-agent/>]. Brian Williams described how the diagnosis component went on to be used in autonomous underwater setting.

Since then autonomous planning and scheduling was used to good effect on Earth Observer-1 (EO-1) in the Autonomous Sciencecraft Experiment. Over more than a decade, EO-1's autonomous control performed flawlessly to do on-board detection of interesting science events (e.g., volcanic eruption) in order to both selectively send data down to Earth, and to retarget for further observation. Overall, this made possible rapid response to (unpredicted) science events, and vastly reduced operations costs. As Steve Chien put it, a very strong "value proposition" was evident.

AEGIS, another science success, autonomously chooses science targets for (first) a camera on the MER rover, and (currently) for the ChemCam laser spectrometer on the Curiosity rover.

There must be a well-understood value to the use of autonomy to spur its adoption by missions. Assurance is a necessary, but not sufficient, consideration.

Missions demand high assurance that autonomy will do no harm, and autonomy may be viewed as "probationary"

From Raymond Francis' presentation we heard of factors that allowed the MSL mission to trust AEGIS on Curiosity: past use (on MER); demonstration of what it *would* do on real data (i.e., images of Mars); duplication of the rover's own safety checks; memory-bounded execution; timeout bounded; robust to faults/anomalies; testing, testing, and testing!; incremental checkout of capabilities; simplicity and ease of use by the mission operators.

Elyse Fosse's description of the ongoing work on M2020's "Simple Planner" echoed these factors. The concept was introduced to the science and operations personnel by "simulating" to them what it would do; reviews followed to show achievability and confirm how it would work in the flow of mission activity planning; prototyping to show, for example, it will operate within available resources; tests, tests and more tests planned!; safeguards at both the mission level and the rover level.

Note that having some form of safeguard – a "safety check", or as the ESA study from over a decade ago called it, a "safety bag", is very desirable as a way to prevent an autonomous system's erroneous command from leading to catastrophe. When such a check is possible, it can be easier to be assured of the safety check itself than of the more complex autonomy, thus leading to assurance of the critical overall system behavior. This is not always possible – e.g., the autonomous targeting of the comet Deep Impact *had* to work – there was no "safe mode" to fall back on!

I sense that some autonomy is still in a "probationary" status, where a (non-catastrophic) fault that could be blamed on autonomy (e.g., loss of a day of science data because the system had to go into safe mode in reaction to some autonomous command) would lead to autonomy's rejection – even though there are occasional such instances due to non-autonomous execution.

As use of autonomy becomes more intertwined with mission success, its scrutiny will fall within the scope of software concerns that NASA's IV&V facility considers for prominent missions. IV&V (and missions themselves) will need to address upcoming challenges as the forms of autonomy (think "AI") become utilized.

All this takes considerable effort.

Algorithms, implementations, performance, models, and configuration parameters – they are all important to understand and validate

It is obviously necessary to V&V the autonomous software's algorithms, their implementation, and their performance with respect to computational resources (e.g., V&V a planner; a diagnosis engine). To the extent that these can be reused on different missions, these aspects may have acquired some level of confidence, but what's changed is the data accompanying the algorithms – models of the system and its behavior that the planner or diagnosis engine reasons from; even just configuration parameters that are changed to match the new circumstances. My impression is that it is relatively less well understood how to V&V these forms of data (and maybe less well formalized what it means).

V&V of systems that themselves learn, or are heavily dependent on learning in their construction, remains a research challenge

We have heard of the challenges of V&Ving deep learning systems (the example Gabor pointed to), and their inscrutability (there doesn't seem to be a way of understanding their decision process). If we are unlikely (at least in the near future) to use such technology in critical control of robotic space missions,

perhaps we can defer consideration of its assurance and look to terrestrial communities to achieve progress in this area. Gabor described the relatively new DARPA project on this topic.

We will need to work with and, as appropriate, adapt/leverage systems and software engineering practices

My impression is that traditional approaches to requirements flow-down (decomposition into lower-level requirements) and testing associated with requirements may need some adjustment to deal with requirements for autonomous systems. As Madeline Diep (and others) pointed out, it is easy to state a very high level requirement on an autonomous system – e.g., that it achieves its goal and remains safe as it does so – the decomposition of that requirement becomes challenging because of the combinatorics. This is closely coupled with the challenge of sufficiently testing an autonomous system – since it is intended to operate in a wide variety of circumstances, this would seem to require a lot of testing, much more than typically would be the case. The interplay between testing in simulation, testing with (some) real data (e.g., as was mentioned above, testing AEGIS with real Mars images), testing with (some) actual hardware, testing in conditions that as realistically as possible mimic what might be the operational conditions (e.g., testing a rover in JPL’s “Mars Yard” or even out in the desert) is, I think, not well understood. Techniques such as the combinatorial methods from NIST (Rick Kuhn’s presentation) offer to help decide efficient test suites, once we have decided what factors are important to vary, and what values of those factors are of interest.

Risk management pervades our mission development and assurance practices. I am not sure we know how to assess the risks that autonomous systems may exhibit.

The possible interplay between Model Based Systems Engineering (MBSE) and autonomy was a topic we had no time to explore (in fact, Emmanuel Lesser’s presentation identified it as “Potential future work”). Given that systems engineers will be, in some cases already are, constructing models of the systems as they do their design and development, it seems plausible that autonomy will be able to leverage these models.

Since autonomy is implemented in software, analysis methods for software systems’ V&V hold promise in application to autonomy software, especially the gamut of “formal methods” some instances of which we heard from the presentations by Mats Heimdahl, Connie Heitmeyer, and Klaus Havelund. The allure of some of these methods is their ability to determine properties of system designs without requiring explicit exploration of every possible behavior. There is further overlap here between the models that emerge from MBSE, the models that some forms of autonomy depend upon, and the formal methods that apply well to models.

p.s.,

Autonomy vs Automation

If you were disappointed in the lack of discussion of autonomy vs automation (I was advised to steer clear of this topic as it has the potential of leading to length exchanges), you may be interested in the following article, recently brought to my attention:

David B. Kaber (2017): A conceptual framework of autonomous and automated agents, Theoretical Issues in Ergonomics Science, DOI: 10.1080/1463922X.2017.1363314

Attendees

Please note: The JPL email domain is changing from jpl.nasa.gov to jpl.caltech.edu at an unspecified future date. E.g., Martin.S.Feather@jpl.nasa.gov will change to Martin.S.Feather@jpl.caltech.edu – both forms of address work right now. The list below used the jpl.caltech.edu version.

Allen Nikora	JPL/Caltech	Allen.P.Nikora@jpl.caltech.edu
Ben Smith	JPL	ben.smith@jpl.caltech.edu
Brian Williams	MIT	Williams@mit.edu
Chris Landauer	The Aerospace Corp	chris.landauer@aero.org
Connie Heitmeyer	Naval Research Lab	constance.heimmeyer@nrl.navy.mil
Elyse Fosse	JPL	efosse@jpl.caltech.edu
Emmanuel Lesser	ESA	EMMANUEL.LESSER@ESA.INT
Fernando Figueroa	NASA Stennis	fernando.figueroa@nasa.gov
Gabor Karsai	Vanderbilt University	gabor.karsai@vanderbilt.edu
Gerek Whitman	NASA IVV / Engility	Gerek.A.Whitman@ivv.nasa.gov
George Gorospe	NASA Ames	george.e.gorospe@nasa.gov
Guillaume Brat	NASA Ames	guillaume.p.brat@nasa.gov
Harald Schone	JPL/Caltech	Harald.Schone@jpl.caltech.edu
John Day	JPL	john.c.day@jpl.caltech.edu
John Kelly	Retired NASA & JPL	jkelly7@mac.com
Keisuke Sugawara	JAXA	sugawara.keisuke@jaxa.jp
Kenneth Costello	NASA/IVV	kenneth.a.costello@nasa.gov
Kirstie Bellman	The Aerospace Corp	Kirstie.L.Bellman@aero.org
Klaus Havelund	JPL/Caltech	klaus.havelund@jpl.caltech.edu
Ksenia Kolcio	Okean Solutions	ksenia@okeansolutions.com
Leila Meshkat	JPL	leila@jpl.caltech.edu
Lorraine Fesq	JPL/Caltech	Lorraine.M.Fesq@jpl.caltech.edu
Madeline Diep	Fraunhofer CESE	mdiep@fc-md.umd.edu
Mallory Graydon	NASA	m.s.graydon@nasa.gov
Martha Wetherholt	NASA HQ	martha.wetherholt@nasa.gov
Martin Feather	JPL	martin.s.feather@jpl.caltech.edu
Mats Heimdahl	University of Minnesota	heimdahl@umn.edu
Michael Aguilar	NESC/NASA	michael.aguilar@nasa.gov
Mitch Ingham	JPL	mitch.ingham@jpl.caltech.edu
Nelson Brown	NASA Armstrong	Nelson.Brown@nasa.gov
Paula Ward	MIT Lincoln Laboratory	plward@ll.mit.edu
Priyanka Srivastava	JPL/Caltech	Priyanka.Srivastava@jpl.caltech.edu
Raymond Francis	JPL	Raymond.Francis@jpl.caltech.edu
Rebecca Castano	JPL/Caltech	Rebecca.Castano@jpl.caltech.edu
Rich Doyle	JPL	rdoyle@jpl.caltech.edu
Rick Kuhn	NIST	kuhn@nist.gov
Seung Chung	JPL	seung.h.chung@jpl.caltech.edu
Steve Chien	JPL	steve.a.chien@jpl.caltech.edu

Acknowledgements

Many thanks to the following:

- JPL's Office of Safety and Mission Success (500) for their sponsorship
- Harald Schone (JPL) for the impetus behind the meeting
- The steering committee: Pat Beauchamp (JPL), Guillaume Brat (NASA Ames), Ken Costello/Don Ohi (NASA IV&V), Martha Wetherholt (NASA HQ)
- Advice and pointers from Steve Chien (JPL)
- Advice and guidance throughout from Lorraine Fesq (JPL)
- Allen Nikora for dinner arrangements at the Athenaeum (photos below by Martha Wetherholt):



Left-to-right:
Ksenia Kolcio, Lorraine Fesq, Leila Meshkat



Front-to-back:
Fernando Figueroa, Allen Nikora, Kirstie Bellman