



Engineering Resilient Space Systems

Final Report

Study start date: 30 July 2012
Study end date: 28 February 2013
Final Report submission date: 27 September 2013

Team Leads:

Richard M. Murray
California Institute of Technology
murray@caltech.edu

John C. Day, Michel D. Ingham, and Leonard J. Reder
Jet Propulsion Laboratory
john.c.day@jpl.nasa.gov, michel.d.ingham@jpl.nasa.gov, reder@jpl.nasa.gov

Brian C. Williams
Massachusetts Institute of Technology
williams@mit.edu

TABLE OF CONTENTS

1.0	Executive Summary.....	1
2.0	Introduction.....	4
2.1	Scientific Motivation and Opportunities	5
2.2	Technical Motivation and Opportunities	7
3.0	Study Description	9
4.0	Outcomes of the Study	13
4.1	Focus Group Outcomes	13
4.1.1	Reference Mission Focus Group Outcomes	13
4.1.2	Capabilities Focus Group Outcomes	15
4.1.3	Architecture Focus Group Outcomes.....	15
4.2	Insights from Lightning Talks	21
4.2.1	What are the Appropriate Metrics?	21
4.2.2	Borrow Technologies from Other Domains	21
4.2.3	Observations on Technology	21
4.3	Roadmap for Technical Development	23
4.3.1	Venus Lander	24
4.3.2	Mars Sample Return	25
4.3.3	Trojan Tour and Rendezvous	26
4.3.4	Synthesis of Capability Patterns	28
4.4	Continuing and Future Work.....	32
4.4.1	KISS Technical Development Proposal	32
4.4.2	Other Tasks and Proposal Concepts	34
5.0	Conclusions	36
6.0	References	39
	Appendix A: Workshop Participants	45
	Appendix B: Workshop Agendas	46
	Appendix C: Focus Group Membership	52
	Appendix D: Definitions of Resilience.....	53
	Appendix E: Summary of Reference Missions Considered	54
	Appendix F: Architecture for Resilience—A Representative Example	66
	Appendix G: Detailed Report of Capabilities Focus Group Outcomes	68

1.0 EXECUTIVE SUMMARY

Several distinct trends will influence space exploration missions in the next decade. Destinations are becoming more remote and mysterious, science questions more sophisticated, and, as mission experience accumulates, the most accessible targets are visited, advancing the knowledge frontier to more difficult, harsh, and inaccessible environments. This leads to new challenges including: hazardous conditions that limit mission lifetime, such as high radiation levels surrounding interesting destinations like Europa or toxic atmospheres of planetary bodies like Venus; unconstrained environments with navigation hazards, such as free-floating active small bodies; multielement missions required to answer more sophisticated questions, such as Mars Sample Return (MSR); and long-range missions, such as Kuiper belt exploration, that must survive equipment failures over the span of decades. These missions will need to be successful without a priori knowledge of the most efficient data collection techniques for optimum science return. Science objectives will have to be revised ‘on the fly’, with new data collection and navigation decisions on short timescales.

Yet, even as science objectives are becoming more ambitious, several critical resources remain unchanged. Since physics imposes insurmountable light-time delays, anticipated improvements to the Deep Space Network (DSN) will only marginally improve the bandwidth and communications cadence to remote spacecraft. Fiscal resources are increasingly limited, resulting in fewer flagship missions, smaller spacecraft, and less subsystem redundancy. As missions visit more distant and formidable locations, the job of the operations team becomes more challenging, seemingly inconsistent with the trend of shrinking mission budgets for operations support. How can we continue to explore challenging new locations without increasing risk or system complexity?

These challenges are present, to some degree, for the entire Decadal Survey mission portfolio, as documented in *Vision and Voyages for Planetary Science in the Decade 2013–2022* (National Research Council, 2011), but are especially acute for the following mission examples, identified in our recently completed KISS Engineering Resilient Space Systems (ERSS) study:

1. A Venus lander, designed to sample the atmosphere and surface of Venus, would have to perform science operations as components and subsystems degrade and fail;
2. A Trojan asteroid tour spacecraft would spend significant time cruising to its ultimate destination (essentially hibernating to save on operations costs), then upon arrival, would have to act as its own surveyor, finding new objects and targets of opportunity as it approaches each asteroid, requiring response on short notice; and
3. A MSR campaign would not only be required to perform fast reconnaissance over long distances on the surface of Mars, interact with an unknown physical surface, and handle degradations and faults, but would also contain multiple components (launch vehicle, cruise stage, entry and landing vehicle, surface rover, ascent vehicle, orbiting cache, and Earth return vehicle) that dramatically increase the need for resilience to failure across the complex system.

The concept of resilience and its relevance and application in various domains was a focus during the study, with several definitions of resilience proposed and discussed. While there was substantial variation in the specifics, there was a common conceptual core that emerged—**adaptation in the presence of changing circumstances**. These changes were couched in various ways—**anomalies, disruptions, discoveries**—but they all ultimately had to do with changes in underlying assumptions. Invalid assumptions, whether due to unexpected changes in the environment, or an inadequate understanding of interactions within the system, may cause unexpected or unintended system behavior. A system is resilient if it continues to perform the intended functions in the presence of invalid assumptions.

Our study focused on areas of resilience that we felt needed additional exploration and integration, namely system and software architectures and capabilities, and autonomy technologies. (While also an important consideration, resilience in hardware is being addressed in multiple other venues, including

other KISS studies.) The study consisted of two workshops, separated by a seven-month focused study period. The first workshop (Workshop #1) explored the ‘problem space’ as an organizing theme, and the second workshop (Workshop #2) explored the ‘solution space’. In each workshop, focused discussions and exercises were interspersed with presentations from participants and invited speakers.

The study period between the two workshops was organized as part of the synthesis activity during the first workshop. The study participants, after spending the initial days of the first workshop discussing the nature of resilience and its impact on future science missions, decided to split into three focus groups, each with a particular thrust, to explore specific ideas further and develop material needed for the second workshop. The three focus groups and areas of exploration were:

1. Reference missions: address/refine the resilience needs by exploring a set of reference missions
2. Capability survey: collect, document, and assess current efforts to develop capabilities and technology that could be used to address the documented needs, both inside and outside NASA
3. Architecture: analyze the impact of architecture on system resilience, and provide principles and guidance for architecting greater resilience in our future systems

The key product of the second workshop was a set of capability roadmaps pertaining to the three reference missions selected for their representative coverage of the types of space missions envisioned for the future. From these three roadmaps, we have extracted several common capability patterns that would be appropriate targets for near-term technical development: one focused on graceful degradation of system functionality, a second focused on data understanding for science and engineering applications, and a third focused on hazard avoidance and environmental uncertainty. Continuing work is extending these roadmaps to identify candidate enablers of the capabilities from the following three categories: architecture solutions, technology solutions, and process solutions.

The KISS study allowed a collection of diverse and engaged engineers, researchers, and scientists to think deeply about the theory, approaches, and technical issues involved in developing and applying resilience capabilities. The conclusions summarize the varied and disparate discussions that occurred during the study, and include new insights about the nature of the challenge and potential solutions:

1. **There is a clear and definitive need for more resilient space systems.** During our study period, the key scientists/engineers we engaged to understand potential future missions confirmed the scientific and risk reduction value of greater resilience in the systems used to perform these missions.
2. **Resilience can be quantified in measurable terms—project cost, mission risk, and quality of science return.** In order to consider resilience properly in the set of engineering trades performed during the design, integration, and operation of space systems, the benefits and costs of resilience need to be quantified. We believe, based on the work done during the study, that appropriate metrics to measure resilience must relate to risk, cost, and science quality/opportunity. Additional work is required to explicitly tie design decisions to these first-order concerns.
3. **There are many existing basic technologies that can be applied to engineering resilient space systems.** Through the discussions during the study, we found many varied approaches and research that address the various facets of resilience, some within NASA, and many more beyond. Examples from civil architecture, Department of Defense (DoD) / Defense Advanced Research Projects Agency (DARPA) initiatives, ‘smart’ power grid control, cyber-physical systems, software architecture, and application of formal verification methods for software were identified and discussed. The variety and scope of related efforts is encouraging and presents many opportunities for collaboration and development, and we expect many collaborative proposals and joint research as a result of the study.
4. **Use of principled architectural approaches is key to managing complexity and integrating disparate technologies.** The main challenge inherent in considering highly resilient space systems is that the increase in capability can result in an increase in complexity with all of the

risks and costs associated with more complex systems. What is needed is a better way of conceiving space systems that enables incorporation of capabilities without increasing complexity. We believe principled architecting approaches provide the needed means to convey a unified understanding of the system to primary stakeholders, thereby controlling complexity in the conception and development of resilient systems, and enabling the integration of disparate approaches and technologies. A representative architectural example is included in Appendix F.

5. **Developing trusted resilience capabilities will require a diverse yet strategically directed research program.** Despite the interest in, and benefits of, deploying resilience space systems, to date, there has been a notable lack of meaningful demonstrated progress in systems capable of working in hazardous uncertain situations. The roadmaps completed during the study, and documented in this report, provide the basis for a real funded plan that considers the required fundamental work and evolution of needed capabilities.

Exploring space is a challenging and difficult endeavor. Future space missions will require more resilience in order to perform the desired science in new environments under constraints of development and operations cost, acceptable risk, and communications delays. Development of space systems with resilient capabilities has the potential to expand the limits of possibility, revolutionizing space science by enabling as yet unforeseen missions and breakthrough science observations.

Our KISS study provided an essential venue for the consideration of these challenges and goals. Additional work and future steps are needed to realize the potential of resilient systems—this study provided the necessary catalyst to begin this process.



ENGINEERING RESILIENT
S P A C E S Y S T E M S

2.0 INTRODUCTION

The recent Planetary Science Decadal Survey (National Research Council, 2011) describes missions that have tremendously challenging requirements on resilience. The spacecraft that support these missions, and the next generation beyond them, must be capable of reasoning about their own state and the state of the environment in order to predict and avoid hazardous conditions, to recover from internal failures, and to ultimately meet critical science objectives in the presence of substantial uncertainties. Moving beyond the current state of the practice requires a fundamental paradigm shift in the way we conceptualize, design, implement, validate, and operate these systems. The challenge is to figure out a way to effectively develop, integrate, and deploy such reasoning capabilities in order to enable new classes of missions at an acceptable cost, without introducing real or perceived risk of mission failure. Our recently completed KISS study investigated the system capabilities, software architectures, and autonomy technologies that will be needed in order to address the anticipated resilience challenges posed by these future missions. The findings resulting from these investigations have set the stage to demonstrate key resilience concepts for future missions through appropriate prototypes.

Many different concepts are related to idea of resilience, including, but not limited to, robustness, adaptability, flexibility, autonomy, fault tolerance, robustness, and operability. During the study, several definitions of resilience were proposed and discussed.¹ While there was substantial variation in the specifics, there was a common conceptual core that emerged—**adaptation in the presence of changing circumstances**. These changes were couched in various ways—*anomalies, disruptions, and discoveries*—but they all ultimately had to do with changing assumptions. All engineered systems, in order to function properly, are based on assumptions about the construction of the system and the environment in which the system is intended to work. These assumptions can be simple or complex, and are typically in regard to the environment (radiation levels, dust storms, temperatures) or the system (turning on a power switch results in a device being active, a device is or is not healthy, performing action *X* affects action *Y* in a particular way). Invalid assumptions, whether due to unexpected changes in the environment, or an inadequate understanding of interactions within the system, may cause unexpected or intended system behavior. A system is resilient if it continues to perform the intended functions in the presence of invalid assumptions. Resilience is a system characteristic or property that increases the robustness of these systems. If the set of assumptions is constant, complete, and unchanging, then no resilience is needed in a system.

In the face of changed assumptions, adaptation of the system configuration and behavior may be required in order to meet the current set of system objectives. The set of alternatives available to a system is one measure of the resilience of a system. The adaptation must be done in a timely manner—the process of adapting to changes (generating and validating revised plans, application of system configuration changes) must be faster than the propagation of effects in order to maintain a high probability of achieving the intended objectives of the system. The timeliness is akin to ‘time to criticality’ in fault protection and hazard assessments, or control latency in control theory.

The ability to adapt to change can be allocated to the end product (in our domain, the spacecraft or rover) or to some other element of the larger system (e.g., the users or operators). If the adaptation process involves the users or operators, then there is a much longer timeframe for reacting to changes. For the envisioned missions and systems discussed in the study, the dynamic and uncertain nature of the environment, coupled with the long distances and time delays, often requires that the adaptation be performed autonomously. Therefore, much of the discussion at the workshops and during the study period focused on ways and means to develop and operate systems that have autonomous resilience capabilities. Despite this attention to autonomy, significant benefits are also expected when resilience is explicitly

¹ See Appendix D for a listing of some of the definitions proposed and discussed during the study.

treated in the design of a broad range of engineered systems—whether highly autonomous or not, primarily in probability of mission success. For example, the loss of the Galileo high-gain antenna (HGA) during the cruise to Jupiter had a huge impact to the mission (Jansma, 2011). In essence, the mission could not be accomplished without the HGA. However, due to the ingenuity of the operations team and the flexibility inherent in the Galileo design, adaptations were made to deal with this failure, and the mission science goals were eventually accomplished. The resilience exhibited by the Galileo mission system (the Galileo spacecraft, ground system, and operations team) allowed this adaptation to be possible. In practice, engineered systems (including operators and users) are only capable of successfully adapting to a subset of invalid assumptions. Further, autonomous adaptation will be limited to a smaller subset of failed assumptions. The goal of the study and of the follow-on work is to explore theories, methods, and technologies that (i) make the consideration of resilience explicit in the design process, and (ii) deploy autonomous adaptation capabilities to enable envisioned (and not yet imagined) missions, and to perform these missions with reduced cost and risk.

2.1 Scientific Motivation and Opportunities

Several distinct trends will influence space exploration missions in the next decade, including hazardous conditions, unknown or unpredictable conditions, various elements (multielement missions), and long-duration flight. Destinations are becoming more challenging and science questions more sophisticated and, as mission experience accumulates, the most accessible targets are visited and the knowledge frontier advances to more difficult, harsh, and inaccessible environments. This leads to new challenges including: hazardous conditions that limit mission lifetime, such as the high radiation environment of Jupiter or the toxic atmosphere of Venus; unconstrained, navigation hazards, such as free-floating active small bodies; multielement missions to acquire information to answer more sophisticated questions, such as MSR; and long-range missions that must survive equipment failures over the span of decades, such as Kuiper belt exploration. These missions often take place without a priori knowledge of the most efficient data collection for optimum science return. Science objectives may have to be revised ‘on the fly’, with new data collection and navigation decisions on short timescales, rather than relying on Earth-based communications with increasingly longer light-time delays.

Yet, even as science objectives are becoming more aggressive, several critical resources remain unchanged. Anticipated improvements in the DSN will only marginally improve the bandwidth and communications cadence to remote spacecraft. Physics imposes unsurpassable light-time delays. Fiscal resources are increasingly limited, with fewer flagship missions, smaller spacecraft, and less subsystem redundancy. As missions visit more distant and formidable locations, the job of the operations team becomes more challenging. Yet, mission budgets for operations support are limited, encouraging significant reductions in operations team size. How then can we continue to explore new and challenging locations without increasing risk or system complexity?

These challenges are present in some degree for the entire Decadal Survey mission portfolio documented in *Vision and Voyages for Planetary Science in the Decade 2013–2022* (National Research Council, 2011), but are especially acute for the following examples.

A Venus lander designed to sample the atmosphere and surface of Venus, to understand its differences from Earth, would have to perform its entire mission in mere hours; accumulated heat would quickly cause subsystem failure and eventual loss of the spacecraft. This would admit little or no direct control from the ground. Lander concepts such as the Venus In-Situ Explorer (VISE) (National Research Council, 2003; Esposito, n.d.), Surface and Atmosphere Geochemical Explorer (SAGE) (Jones, 2003) or Venus Intrepid Tessera lander (VITaL) (National Aeronautics and Space Administration, 2010a) call for in situ or contact instruments to get mineralogical data from the Venus surface during this short interval. These instruments (such as Raman or laser-induced breakdown spectroscopy [LIBS] spectrometers) have narrow fields of view, and would therefore require careful target selection for both science and engineering reasons. However, the uncertain surface morphology could leave the spacecraft oriented at

suboptimal angles and the restriction on ground interaction combined with a strong desire to understand unweathered bedrock material puts the science at risk. This mission would benefit from the application of autonomous capabilities to improve the probability of mission success, including the ability to select optimal science targets, downlinking the best data first and the ability to continue performing science as subsystems begin to fail.

A Trojan asteroid tour and rendezvous mission presents another case of short timescales. The Trojan objects are thought to be ancient and relatively unprocessed remnants of the primitive solar system; their compositions will tell us a great deal about the distribution and migration of material throughout history. However, these objects are distant with very low albedo that prevents them from being accurately surveyed from Earth. A Trojan tour spacecraft (National Aeronautics and Space Administration, 2010b) would therefore act as its own surveyor, finding new objects as it approaches and possibly discovering targets of opportunity on short notice. Little is known about the Trojans, so science observations and possibly the spacecraft trajectory would be redefined on the fly. Positional uncertainty would require rapid changes to the pointing and exposure parameters for both cameras and spectrometer instruments. The mission might achieve this by some combination of onboard replanning or fast reactions from the ground. Flybys are time-critical events where a retreat to safe mode results in irrecoverable loss of science yield.

An MSR mission would return a sample from Mars to Earth for state-of-the-art analyses in laboratories all over the world, perhaps providing the ‘holy grail’ for understanding the history of Mars (National Aeronautics and Space Administration, 2010c; 2010d; Mattingly and May, 2011). Such a mission includes the need for fast reconnaissance over long distances, interaction with an unknown physical surface, and strong desire to handle any possible degradation or faults while maintaining the mission plan. Sample handling introduces new challenges, as multiple spacecraft are envisioned to collect and transfer high-value samples through a long handling and return chain with many potential points of failure. Developing and operating this multispacecraft system in a cost- and mass-constrained world requires an understanding of risks and the tradeoffs inherent in their mitigation. Applying appropriate resilience techniques to circumvent, withstand, and recover from failure has high payoff in these scenarios.

A common thread throughout all these scenarios is that the environment, science observations, and spacecraft capabilities are not fully known in advance, to a degree much greater than on historical and present missions. These missions will benefit from *rapid mission planning* (ability to reduce the duration of science planning cycle), *onboard data understanding* (ability to assess data relative to science objectives), *graceful degradation* (ability to continue science through faults and unexpected environmental interactions), and *low-cost/low-risk cruise* (ability to reduce workforce cost of cruise without increasing risk, including ability to hibernate). The promise of truly resilient spacecraft that can succeed and thrive in the presence of dynamic and uncertain situations opens up new frontiers of possibility without sacrificing cost or risk. The current development paradigm of determining all the factors that affect risk and mission success at design time, or in a slow ground-based reaction to changes in these assumptions, are coarse and costly tools for addressing the challenges posed by these missions. These missions can be done with current technology and the current development paradigm, but inclusion of proven resilience capabilities allows for otherwise unavailable design options, and cost- and risk-avoidance approaches and methods. Both near term or far term, future missions will greatly benefit from a greater understanding and infusion of efforts made to understand and increase their resilience, whether the resilience is embedded in a better understanding of the design trades space, or is implemented in onboard autonomous capabilities. Incorporation of resilience to future spacecraft will reduce the risk in the collection of high-value science from more difficult, hazardous, and remote locations, enabling a greater science-cost benefit.

2.2 Technical Motivation and Opportunities

The required resilience to implement these missions cannot be achieved by simply incrementally building on and extrapolating from the current state of the practice; it requires a fundamental paradigm shift in the way we conceptualize, design, implement, validate, operate, and evolve our systems. The current paradigm relies on traditional approaches to preserve the spacecraft in known environments and in response to internal faults—it employs hardware redundancy, shielding, hundreds of preprogrammed reflexes and large technical margins. These solutions have significant costs across multiple dimensions (e.g., power, weight, complexity) and have limited effectiveness in addressing environmental uncertainty. Continued reliance solely on these approaches limits the classes of missions we are capable of pursuing, limits the science return, and limits the level of resilience that is achievable for the missions

we fly, hence translating to increased technical risk. There is a need for an addition to the traditional approaches that includes a balance of both reflex-oriented behavior and the ability to reason about the current state of the system and environment in a comprehensive way. The challenge is to figure out a way to effectively develop and integrate such capabilities in order to enable the new class of missions, to deliver an acceptable probability of returning high-value science (Figure 2-1).

Our study program investigated the system capabilities and autonomy technologies that will be needed to achieve the required resilience for such missions, and the challenges of developing these technologies and capabilities. While we expect continued improvement in hardware reliability, robustness, and cost-effectiveness, the exponential leaps envisioned will require significant advances in systems and software capabilities and, in many cases, renders currently used verification and validation (V&V) techniques nonscalable to such magnitudes. Therefore, we focused on (i) the novel systems engineering techniques needed to architect, design, implement, validate, and operate these systems, and manage their associated complexity; and (ii) the software technologies that will be relied upon to provide the requisite intelligence and behavior for these systems.

The problem of achieving sufficient resilience does not lack for interest—researchers and practitioners in various environments are developing the frameworks, technologies, and applications to address this challenge from different perspectives and domains (safety, fault tolerance, systems engineering). What is missing is a venue for the integration of these ideas. Our study provided this essential catalyst, bringing together key participants in sessions structured to foster communication and focus on development of an actionable plan.

The timeliness of this study is based on three key factors:

1. **Recent scientific developments:** as mentioned above, the recently published Planetary Science Decadal Survey (National Research Council, 2011) called out at least two high-priority New Frontiers missions that will require the type of resilience we are proposing to address (Venus In-Situ Explorer [VISE] and Trojan Tour & Rendezvous [TTRV]). Also underscoring the need to address the resilience challenge is the recent identification by the Kepler space telescope of many extrasolar planets in the habitable zone around other stars (Borucki et al., 2011). This discovery prompts the space science and exploration community to begin conceiving missions that have sufficient resilience to enable exploration of these planets.

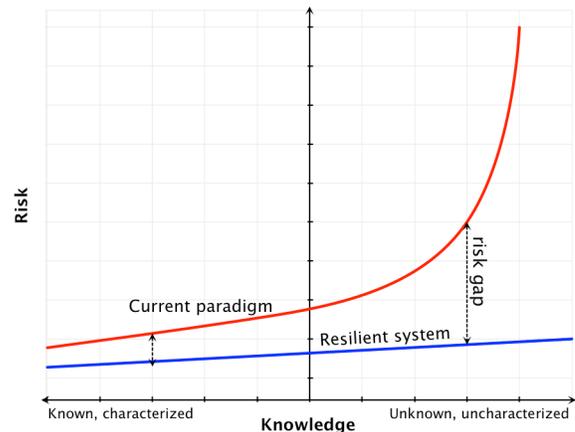


Figure 2-1: A new paradigm is needed to achieve acceptable risk for highly complex missions in highly unknown/uncertain environments.

2. Momentum building in the NASA community: the issues of system complexity and robustness, two key facets of the resilience problem, have been the focus of significant attention recently in the NASA community. For example, the recent NASA Flight Software Complexity study (Dvorak, 2009) identified some of the fundamental software-related challenges to be addressed in order to achieve resilience. The Science Mission Directorate sponsored the first fault management (FM) workshop (Fesq, Fretz, and Newhouse, 2013) to identify and characterize the problems faced by recent missions; a follow-on FM workshop was held on April 9–12, 2012. Finally, NASA’s Office of the Chief Engineer has convened the NASA Integrated Model-centric Architecture (Conroy, Mazzone, and Lin, 2013) activity to leverage the game-changing potential of model-based systems engineering as a means to conquer system complexity. Each of these efforts targets a different aspect of the broader resilience challenge.
3. Maturity of relevant technology: Missions like Earth Observing One (EO-1) (Chien et al., 2005) and Deep Space One (DS-1) (Rayman, Varghese, Lehman, and Livesay, 2000) have flight-demonstrated certain key technologies and architectural concepts that are required for truly resilient space systems, but these applications have not focused on achieving the levels of robustness and adaptability to uncertain environments required to confidently execute scenarios like those described above. State-of-the-art autonomous control architectures (e.g., JPL’s Mission Data System (MDS)) that take a more holistic approach to autonomy and fault management have been successfully demonstrated in prototype system environments, but have not yet found a project customer to fully deploy these capabilities.

Some of the relevant technologies and techniques include model-based systems engineering (e.g., JPL’s Integrated Model-Centric Engineering initiative (Bayer et al., 2010)), model-based reasoning (e.g., Remote Agent’s Livingstone diagnosis and repair system—NASA Software of the Year 1999 (Williams and Nayak, 1996)), deliberative planning and scheduling with plan repair (e.g., JPL’s Continuous Activity Scheduling Planning Execution and Replanning [CASPER] planning and execution system (Chien, Knight, Stechert, Sherwood, and Rabideau, 1999)), smart executives (e.g., Remote Agent EXEC system—NASA Software of the Year 1999 (Gat, 1997)), automated data analysis (e.g., JPL’s Autonomous Exploration for Gathering Increased Science [AEGIS]—NASA Software of the Year 2011 (Estlin et al., 2012)), formal methods for software V&V (e.g., Simple Promela Interpreter [SPIN] model checker (Holzmann, 1997)), sensor fusion, integrated system health management, adaptive control systems, and machine learning techniques. Many of these technologies are fairly mature (Technology Readiness Levels [TRL] 7–8, having been flight validated (Mankins, 1995)), and are ready to be integrated into a system context.

3.0 STUDY DESCRIPTION

The study consisted of two workshops, separated by a seven-month focused study period. This structure was chosen to allow the topic to be explored in multiple ways, and allow for continued opportunity for collaboration between the workshops. The study focused on areas of resilience that we felt needed additional exploration and integration, namely, system and software architectures, approaches, and capabilities. While also an important element, resilience in hardware is being addressed in multiple other venues, including other KISS studies.

"If you want to build a ship, don't drum up people to collect wood, and don't assign them tasks and work; but rather teach them to long for the endless immensity of the sea."

Antoine de Saint-Exupery

The structure and themes for our study are shown in Figure 3-1. The first workshop explored the ‘problem space’ as an organizing theme, and the second workshop explored the ‘solution space’. Summary descriptions of each workshop are included below, and detailed agendas for each workshop are included in Appendix B.

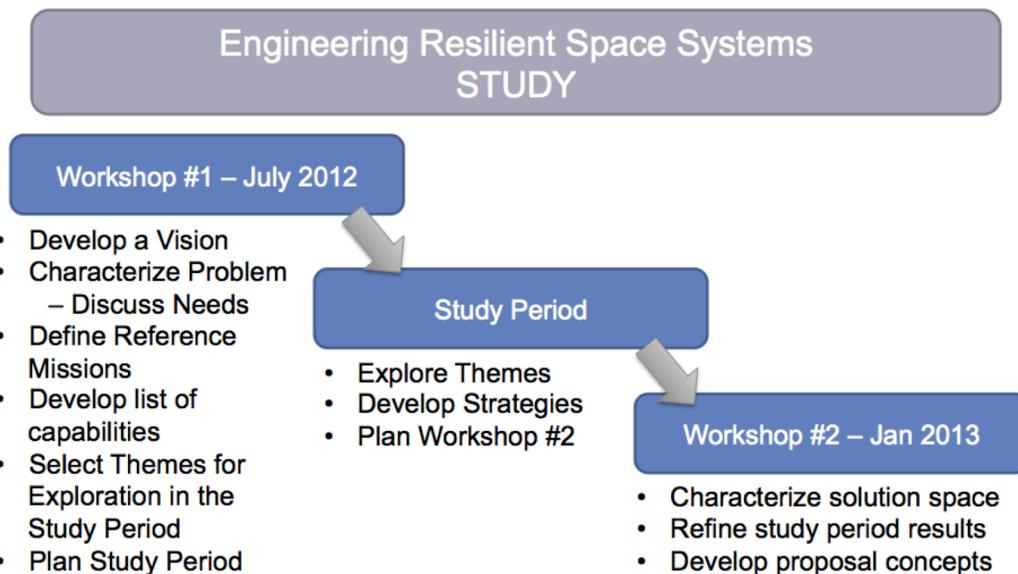


Figure 3-1: Structure of Study

The first workshop was held July 28 through August 3, 2012. The first day of the workshop was marked by a series of short courses to present fundamental ideas related to resilience. The day was moderated by Len Reder (JPL), who introduced the session with some key questions about the nature of resilient systems. The specific short courses presented were (www.kiss.caltech.edu/workshops/systems2012/index.html):

- “Principled System Architecture” (Dr. Robert Rasmussen, JPL)
- “Capturing Flight Software Architecture using Domain-Specific Language” (DSL; Dr. Kim Gostelow, JPL)
- “Control Theory and Methods” (Dr. Richard Murray, Caltech)
- “Autonomy Practices” (Dr. Brian Williams, Massachusetts Institute of Technology [MIT])
- “Ultra-Reliability for Interstellar Missions” (Dr. Henry Garrett, JPL)

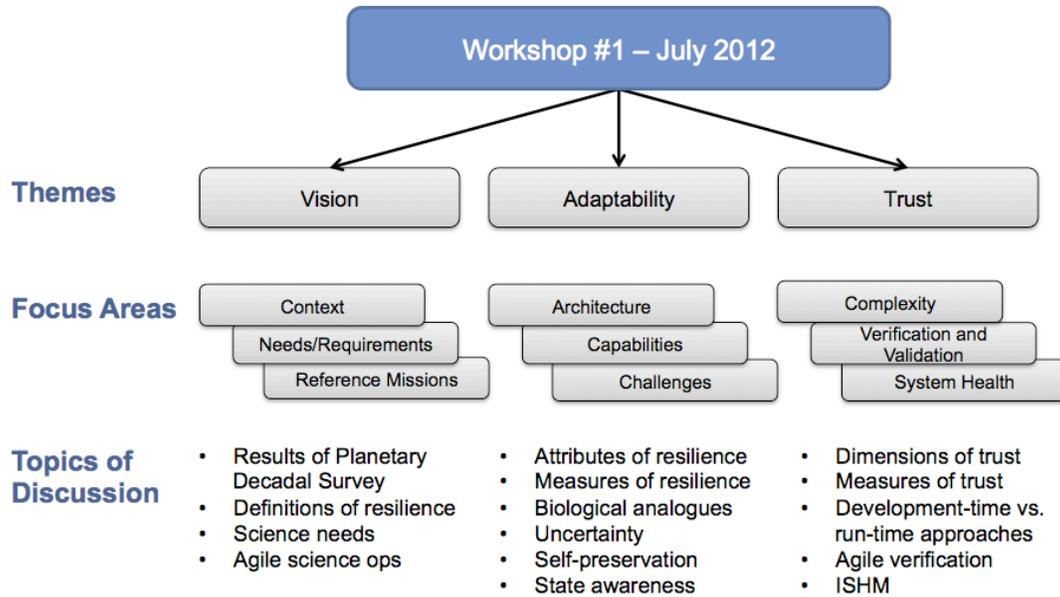


Figure 3-2: Workshop #1 Themes and Topics

In support of the theme of ‘exploring the problem space’, a different topic was designated for each day of the workshop. These topics and areas of focus are shown in Figure 3-2. The first topic was *need*, and focused on the vision behind the study, and rationale and need from the science perspective. This day was moderated by Dr. Michel Ingham (JPL), who led discussions to establish a vision and conceptual basis for resilience in engineered systems, document science motivation for resilience, and capture a set of driving reference missions. The second topic was *adaptability*, moderated by Dr. Richard Murray (Caltech). Discussions on this day focused on establishing key architectural attributes for adaptability, documenting key capabilities for resilient systems, and determining challenges to realizing envisioned systems. The third topic was *trust*, moderated by Dr. Brian Williams (MIT). These discussions addressed key architectural attributes for trustworthy resilience, and the documentation of development-time and run-time approaches for achieving trust in future resilient systems. On the final day of the workshop, moderated by John Day (JPL), the topic was *synthesis*, and discussions focused on weaving together the disparate discussions and thoughts from the prior four days into a focused plan for the study period.

During these four days, there were a series of talks given by study participants and invited speakers. These talks were grouped into two categories: *context talks*, intended to provide additional background for consideration of resilience in space systems, and *provocative talks*, intended to spur discussion and the generation of new ideas. A full listing of these talks is included in Appendix B. On Wednesday afternoon, there was a session for the postdoctoral students and early career hires to present their interests and research. In addition, as a wrap-up activity, on Thursday afternoon, each participant gave a 2-minute *lightning talk*, which was their opportunity to summarize the important aspects of the workshop from their perspective. We found this activity to be a very important tool for collecting input from each participant, and provided a useful summary of the themes of the workshop. Additional information on the lightning talks is described in Section 4.1.

The work to be performed during the study period was defined on the last day of Workshop #1, as part of the synthesis activity on that day. The study participants, after spending the prior days discussing the nature of resilience and its impact on future science missions, decided to split into three focus groups, each with a particular thrust, to explore specific ideas further. The three focus groups and areas of exploration were:

1. Reference missions: address/refine need by exploring reference missions
2. Capability survey: look both inside and outside NASA to collect, document, and assess efforts and technology that could be used to address the science need
3. Architecture: analyze the impact of architecture on system resilience, and provide principles and guidance for architecting greater resilience

In each of these focus areas, a subset of the study participants worked together to plan and execute their efforts. A detailed description of the work performed and resulting products is described more fully in Section 4.1 of this report.

The second workshop was held February 26–28, 2013. In the final workshop, we reviewed the work performed by the focus groups during the study period in a series of outbriefs, and then began the two-step process of exploring the ‘solution space’. The first step consisted of a reference mission exercise to focus on specific/concrete needs, and the second step was a synthesis and planning session. The scope of the workshop is shown in Figure 3-3, and the flow of activities is shown in Figure 3-4.

For the reference mission exercise, the participants were split into three teams, each with the intent of reviewing a specific reference mission. Each of these teams cut across the focus groups in order to encourage cross-fertilization of ideas in these discussions. Each team was tasked with looking at key needed capabilities for a specific reference mission. The purpose of the exercise was to understand the mission science goals for each reference mission, and assess the evolvability and extensibility of capabilities by reviewing their utility to an interstellar rendezvous mission. Each team was asked to:

- Review and assess associated mission resilience needs
- Identify and prioritize potential capabilities, technologies, and architectural characteristics to meet needs
- Develop a roadmap for identified capabilities/technologies/architectures (in particular, ideas for near-term tech development proposals)

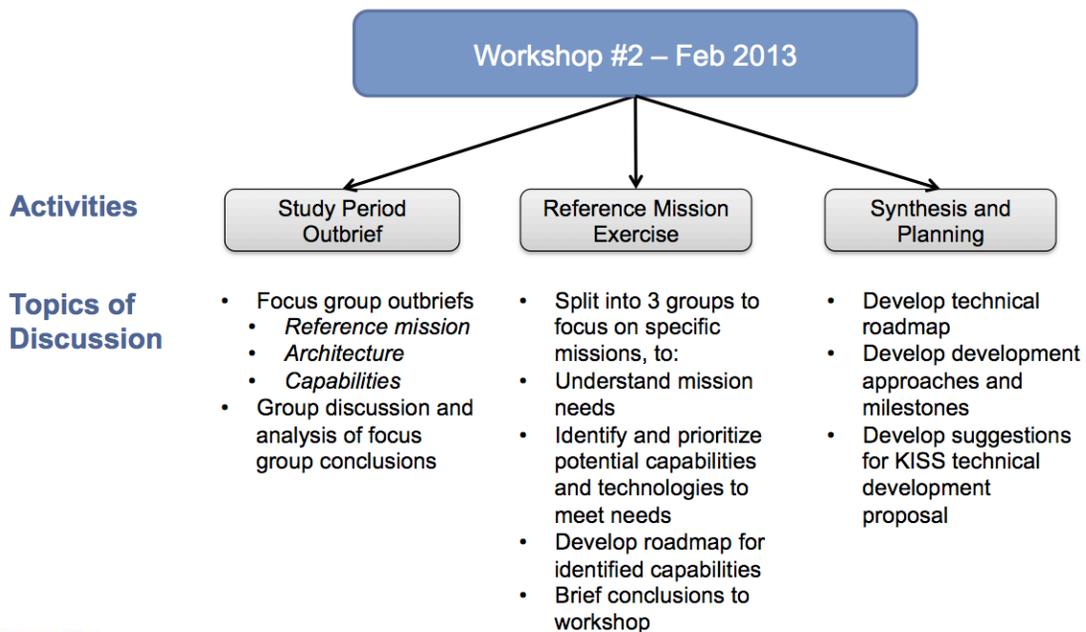


Figure 3-3: Workshop #2 Activities and Topics

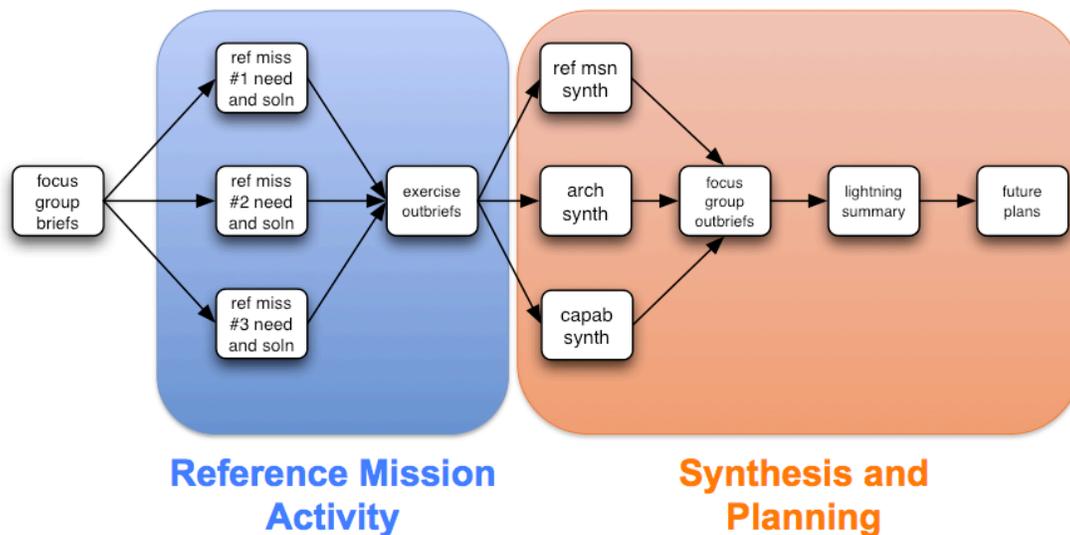


Figure 3-4: Workshop #2 Flow of Activities

The three reference missions used for this exercise, with the rationale for their selection and the moderators for each, were:

- Venus lander: extremely short mission, unknown sampling conditions (L. Tamppari)
- MSR chain: including the 2020 Mars Surface Mission (MSM) in the near term and future stages of sample return, which will require spacecraft with very different capabilities (T. Estlin)
- Trojan tour and rendezvous: long cruise, flyby-type encounters that are known only a few days in advance, unknown characteristics of bodies, perhaps hazardous debris (J. Day)

The synthesis and planning session was divided into three elements: (1) reassessment of results by each of the original focus groups, (2) a round of lightning talks by each participant, and (3) a group discussion of future plans and proposals for pursuing the ideas developed during the two workshops and study period. The results and conclusions of these discussions are described in Section 4.1.

The second workshop also included a set of talks by study participants, but these focused primarily on the research and topics of participants new to the study. In particular, the following talks were given (www.kiss.caltech.edu/workshops/systems2013/index.html):

- “Formalization of Systems Engineering and System Health Management” (Dr. Stephen Johnson, University of Colorado, Colorado Springs)
- “Model-Driven Development of High-Assurance Dynamically Adaptive Systems” (Dr. Betty Cheng, Michigan State University)
- “Autonomous Underwater Explorers” (Rich Camilli, Woods Hole Oceanographic Institute [WHOI])
- “A Framework for Resilient Architecture” (Dr. Kim Gostelow, JPL)
- “Resilience and Cybersecurity” (Dr. Howard Shrobe, DARPA)

4.0 OUTCOMES OF THE STUDY

4.1 Focus Group Outcomes

4.1.1 Reference Mission Focus Group Outcomes

This purpose of the Reference Mission Focus Group was to expound upon the need for resilient systems from the perspective of the science community. The charter of this focus group included the following deliverables:

1. Definition of vision and mission statements to guide our work,
2. Creation of a list of resiliency needs that the current set of reference missions have, and
3. Identification of metrics by which different proposed technologies and/or approaches can be assessed in terms of their efficacy in addressing the reference mission needs

The group accomplished this work primarily through e-mail and biweekly teleconferences.

To better understand the needs and perspectives of scientists working on future missions, we surveyed the set of reference missions from the Decadal Survey, and performed a very productive series of interviews with knowledgeable scientists/engineers for 10 different future missions. These interviews were performed using a set of specific questions intended to elicit capabilities that would drastically improve or enable science from these missions. In doing so, we began relationships with those people that may lend themselves to future endeavors (e.g., future proposals or perhaps for technology infusion into their missions). The specific missions and the individuals that participated in the interviews are captured below:

- Venus Orbiter (Kevin Baines)
- Venus Lander (Sue Smrekar)
- Mars Reconnaissance Orbiter (Sue Smrekar)
- Mars Science Laboratory/Curiosity (Joy Crisp)
- Mars Sample Return (Erik Nilson)
- Trojan Asteroid Flyby and Rendezvous (Julie Castillo-Rogez)
- Europa orbiter mission (Dave Senske, Brian Cooke)
- Titan Balloon mission (Christophe Sotin)
- Interstellar Probe (Gentry Lee)
- Comet Surface Sample Return (Carol Raymond)

These interviews increased our understanding of the needs of these reference missions that could be addressed or provided through capabilities enabled by advanced software/systems architecture approaches. There were many commonalities among the missions, including rapid science planning, ‘smart’ science return, and maintaining science goals even with degrading/failing hardware/software. We developed categories for organizing the mission-specific needs into generic statements of capability:

- Rapid mission planning
 - **Ability to reduce the duration of science planning cycle**, including ability to analyze current data, plan objectives, generate sequences, de-conflict operations, verify operations, and execute. Enable quick and robust reactions to new objectives.
- Data processing
 - **Ability to downlink best data first**, including the onboard ability to recognize relative science value of collected data and to prioritize the downlink order in response. Also includes ability to provide thumbnails or other metadata to enable on-ground downlink decisions.
- Graceful degradation
 - **Ability to continue science through faults and unexpected environmental interactions**. Not fail safe or fail operational but a level between, which uses system resources in a best

effort manner to achieve the most science through faults, single event upsets [SEUs], resets, etc.

- Low-cost cruise
 - **Ability to reduce workforce cost of cruise without increasing risk**, including ability to hibernate and perform continuous thrusting over long mission cruise periods.

We used common needs from the missions to set a rough priority for the capabilities. Figure 4-1 contains a summary of the capabilities and their assessed value to each mission.

ID	Name	Mission Type	Needed Capabilities			
			Rapid Mission Planning	Data Processing	Graceful Degradation	Low Cost Cruise
1	Venus Lander (e.g., INTREPID, SAGE)	Harsh environment lander	X	X	X	X
2	Venus Aerial Mission	In-situ aerial explorer	X	X	X	X
3	Titan Aerial Explorer	In-situ aerial explorer	X	X	X	X
4	Mars Sample Return (set of missions)	Landed rover	X	X	X	X
5	Orbiter "style" (MRO, Cassini)	Multi-tasking orbiter	X		X	X
6	Trojan Tour and Rendezvous	Small body tour	X	X	X	X
7	Europa Clipper	Repeated High-value flyby			X	X
8	Interstellar Probe	Distant, long-duration explorer		X		X
9	Interstellar Rendezvous	Distant, long-duration explorer	X	X	X	X
10	Comet Surface Sample Return	Sample Return	X		X	X

Capability scoring: Strong/enabling Significant Weaker

Figure 4-1: Summary and Prioritization of Mission Resilience Needs

We used the information gained to focus the discussions in the second workshop by selecting resilience categories that were important to consider and by selecting representative (end member) reference missions to examine. The representative missions selected for the workshop activity were:

- Venus Lander: lander, extremely short mission, unknown sampling conditions
- MSR chain: may include Mars 2020, many different spacecraft with different needs, near-term/realistic
- TTRV: long cruise, flyby-type encounters that are known only a few days in advance, unknown characteristics of bodies, perhaps hazardous debris
- Interstellar probe: very far in future, environment almost completely unknown, provides an ambitious long-range target for our vision for resilience; requires us to think about scalability and evolution of proposed technology

For each of the four selected missions, we developed summary presentations that included a summary of the mission, key measurements, capability needs, mission requirements with success metrics identified, and mission requirements matched with resiliency need categories. These details are contained in Appendix E.

4.1.2 Capabilities Focus Group Outcomes

The goal of the capabilities survey was to look at existing capabilities and formulate lists of the following:

- Enabling software and autonomy technologies (e.g., modeling and knowledge representation, machine learning, automated reasoning, planning and scheduling middleware, languages, frameworks, runtime verification)
- Key processes for agile and verifiable development enabling resilient system development and management of complexity

From these lists, we have identified certain key capabilities areas that have potential based on existing work that has already been done and the current state of the technology. To do this, we started with a series of one-hour teleconferences, which led to several presentations being given at JPL. A single all-day meeting was held at Caltech to capture a broad set of capabilities. We surveyed the Capabilities Focus Group and developed the following sets of areas of interest:

- Software (S/W) design patterns, DSLs, models for software synthesis, automated testing, and standards
- Autonomy patterns, artificial intelligence (AI) planners, and self-adaptive software
- Fault detection, isolation, and recovery (FDIR) patterns and their verification
- Technologies taken from other domains (e.g., automotive manned aircraft, unmanned aerial vehicle [UAV], unmanned underwater vehicle [UUV], etc.)

These areas were selected based on the group's expertise and interests. Additional material has been produced and the summary of the group's discussions is included in Appendix G of this report.

4.1.3 Architecture Focus Group Outcomes

This section of the report summarizes the activities and outcomes of the Architecture Focus Group. The objectives of this focus group were to analyze and discuss the impact of architecture on system resilience, and to provide principles and guidance for architecting our systems for greater resilience. To this end, the Architecture Focus Group held five teleconferences between the first workshop and the second workshop, and volunteered to work on various assignments between the teleconferences. The three primary activities of this focus group were:

- Perform a focused literature review to orient and set context for the group's efforts
- Perform an analysis of architectural resilience, looking at examples (and counter-examples) of resilient architectures and the relevant characteristics they exhibit, and extracting a set of key tradeoffs and principles
- Identify fruitful areas for future research in 'architecting for resilience'

4.1.3.1 Literature Review

Given the short period of the study, the Architecture Focus Group performed a limited review of the pertinent literature to provide appropriate context for the analysis. In particular, the group identified a small number of key references, focusing initially on recent writings from senior members of the focus group and the community, Bob Rasmussen and John Doyle. In addition, a few classic architecture references were identified as essential background for further work in this area.

This preliminary literature review conducted by the focus group revealed a wealth of work in the individual areas of system architecture and resilience, but a comparatively small intersection of work in

architecting systems for resilience. Furthermore, almost no prior work was found in specific area of architecting *resilient space systems*.²

The implication of this finding is that, in order to ground the future direction of study in this important area, the space domain will draw from principles developed and lessons learned in other disciplines, including biology (e.g., bacterial biosphere), computer science (e.g., internet), and power systems (e.g., smart grid).

4.1.3.2 Analysis of Architectural Resilience

The following section of the report addresses the Architecture Focus Group’s initial progress on an analysis of architectural resilience. The compilation of a more extensive bibliography of literature in this important area is identified as an important future step for research.

Resilient Architecture Examples

As part of the Architecture Focus Group’s analysis during the study period, a sample of existing known architectures was articulated and analyzed in terms of resiliency. These architectures are generally well known and are in some cases bio-inspired and in others, technology-inspired. A subset of the collected architectures is presented for analysis and discussion in Table 4-1.

Table 4-1: Architecture Samples

Architecture	Description
Starfish Geckos Worms Jellyfish	These biological systems are designed to survive the loss of a limb or appendage. They benefit from an architecture that has appropriately placed <i>possible failure points</i> , that is, places where a limb or appendage can be more easily severed in the case of a predator grasping and removing it. The architecture provides the ability to regenerate tissue from these points.
Bacteria Viruses	Bacteria have the amazing ability to adapt, within a small number of generations, through mutation, easily developing resistance to antibiotics. Viruses can easily infect and cause changes to another organism's cells through transfer of genes from the virus to the host. Bacteria and viruses exhibit both Vertical Gene Transfer (VGT) and Horizontal Gene Transfer (HGT).
Internet	The internet is designed to be resilient to outages, by redirecting flow of information along a different path to get to the ‘customer’, hence the ability to reconfigure on the fly. However, it was not designed for the level of evolvability it has achieved (i.e., ability to reconfigure [e.g., plug-n-play] enabled it to incorporate new hardware and software). The internet adopted standardized protocols for information transfer enabling the above features.
Columbus' Expedition	Columbus’s expedition made use of three ships. Each ship was diverse with respect to others in terms of capabilities and options. The largest ship, Santa Maria, sunk in 1492 but the expedition (mission) still succeeded. The expedition exhibited the ability to replan and adapt on the fly as needed. The ships’ diversity supported this.
DARPA’s F6 Fractionated Spacecraft (Brown and Eremenko, 2006)	A subset of disaggregation approaches wherein the functionality provided by a single large monolithic satellite is delivered by a cluster of wirelessly networked modules and other networked nodes sharing a variety of H/W and S/W resources. F6 demonstrates several resilience heuristics such as functional and physical redundancy, graceful degradation, survivability, and diversity.

While the above examples span a number of mission and system concepts we can already see similarity in them with respect to resilient architecture heuristics. Physical and functional redundancy are common themes in resilient systems.

Resilient Architecture Characteristics

² Much of the published work in architecting for space systems has been in the area of ‘mission architecting’, which is a subset of the broader discipline of space system architecting focused on early mission formulation.

Next we examine some of the characteristics of the example architectures described above. In identifying characteristics, we examine each architecture’s robustness and fragility to external and internal disruptions (Table 4-2).

Table 4-2: Characteristics of the Example Architectures

Architecture	Architectural Characteristics	Robustness	Fragility
Starfish Geckos Worms Jellyfish	Morphologic tissue regeneration. Targeted points of “frailty” lead to robustness.	Self-healing Regenerative Target points of fragility result in robustness	Weakened/degraded state (at least temporarily)
Bacteria Viruses	HGT - between organisms VGT - parent to child provides evolution/mutation at the genetic level	Resistance to external factors (e.g., antibiotics) Survival - mutations within host cells protect virus	VGT mechanism subject to rapid mutation and therefore deselection. HGT subject to resistance (antibiotic).
Internet	Structural redundancy Flexible/robust protocols (cognitive aspect/’smarts’ in the system)	Functional redundancy Common I/O protocols Adoption of standards and standardization	Global physical address space has led to security problems
Columbus’ Expedition	Distributed system Functional & physical redundancy Robust set of options/choices	Functional redundancy Physical redundancy Diversity	Each ship a single point of failure. All ships subjected to environment.
DARPA’s F6 Fractionated Spacecraft	Distributed system Payload, communications, computing Functional & physical redundancy Standardized information protocol	Significantly enhanced Adaptability and survivability	Wireless communications can be a security threat

Some of the general resiliency heuristics immediately evident in these architectures include:

- *Diversity*
 - This heuristic describes the ability of a system to incorporate different functionality or capability into the system but not in a redundant sense such as hardware redundancy. The choice to use different types of ships on Columbus’ expeditions is an example of this. In the case of the DARPA F6 system, the use of dissimilar satellites (hence, dissimilar hardware, software, and platforms) leads to increased system resilience.
- *Functional and physical redundancy*
 - Redundancy in this respect typically refers to replicated hardware and or software within systems in a fault-tolerant manner to overcome faults or disruptions to this system from external and/or internal factors. Several of the above examples provide clear descriptions of such redundancy.
- *Graceful degradation*
 - This heuristic describes a resilient system’s ability to survive disruptions originating from within or without while still carrying out its missions. Examples of self-regenerative tissue in some of the biological examples above provide the species with the ability to go on living. Loss of function, capability, or capacity in technological systems allows the continuation of their mission with the same or reduced goals.
- *Self-regenerative or self-healing*
 - These attributes exhibited in the first biological example have analogs in some of the technology examples. The internet has the ability to cut off nodes and links due to failures while incorporating new nodes and links on the fly so long as the standard protocols are satisfied. The F6 system, while not being able to regenerate new satellites immediately,

certainly has the ability to degrade gracefully to await a new replacement satellite to incorporate into the existing cluster.

- Information transfer mechanisms
 - Bacteria and viruses exhibit HGT and VGT—communication of data across interfaces. Similarly, the internet incorporates protocols to do the same—transfer data/information both horizontally (node to node) and vertically (through the Open Core Protocol (OCP) stack layers from applications to physical layer). Likewise, the F6 architecture incorporates the internet protocol as well as a standardized messaging protocol across all nodes in the distributed system, thus mimicking HGT and VGT in the biological world.

Resilient Architecture Tradeoffs

Another way to understand resilience is in the context of the tradeoffs typically exhibited by resilient systems. Tradeoff analysis, which involves comparing the cost and benefits between two well-understood options, helps bridge the gap between the more abstract concepts of resilience present in nature to those relevant in the space domain. Tradeoff analysis is especially pertinent in the context of our resilience discussion, because the resilience of a mission is framed by its performance in the face of the risks, while tradeoff analysis inherently weighs the risks different options pose to the mission.

- *Passive vs. active resilience*
 - One of the most useful ways to compare designs that provide resilience is to lay them out on a spectrum that runs between the categories of passive and active resilience. Passive resilience refers to properties that are inherently robust against classes of uncertainties, ‘resilient-by-construction’. Examples of *passive resilience* include using a thicker casing for a satellite to mitigate the impact of micrometeorites and the longitudinal static stability of an aircraft in flight. *Active resilience* refers to the ability of a system to circumvent failure, attack, and adapt to changing mission objectives, and implies some form of control. Examples of active resilience include fault adaptive control, automated planning, and controller synthesis.
- *Reflexive vs. deliberative*
 - The manner in which the active resilience is achieved can be further classified as *reflexive* versus *deliberative*. In the most general sense, this tradeoff addresses the question of how much online reasoning is required. More online reasoning demands more resources, such as processing and power storage, potentially requires more V&V, and is less responsive, but has the benefit of being able to address novel or unaddressed situations. In the field of AI, these categories can be compared to weak AI (rule-based or behavior-based robotics to mimic human behavior) vs. strong AI (systems designed to meet or exceed human capabilities).
- *Resilience/robustness vs. efficiency/cost effectiveness*
 - One explanation for the lack of adoption of resilience technologies is a perceived tradeoff between *resilience* and *efficiency*. In the context of space missions, where mission objectives emphasize new science, current practice allocates risk and funds to fly new science instrumentation, and depends on ‘tried-and-trusted’ supporting technologies and development approaches. Therefore, using mission funds to develop technologies for resilience reduces the apparent science return per dollar spent. However, this view does not take in all the considerations. From a probabilistic point of view, the real objective should be to maximize the *expected value of science return*. In other words, we need to take into account the uncertain variable, their interactions, and the associated loss of science value.
 - There is a precedent for flying resilience; historically, limited forms of resilience are present in riskier mission components such as engines (redundant valves) and sensors (voting mechanisms). But, as our missions evolve and become more complex, so will the interactions

between their components. This evolution demands a similar evolution in technologies for resilience.

- *Complexity vs. verifiability*
 - As the *complexity* of a mission increases, so does the difficulty of verifying its behavior. In this tradeoff, we refer to a very general definition of complexity: the quality of having a large number of different parts and interconnections. Complexity can come from both the problem and the solution. Whether the problem is more complex or the engineered solution is more complex, both complicate verification.
 - Another complication to the V&V process is operating in the presence of unknowns. How do we even begin to write specifications for a spacecraft that must survive conditions that we aren't even aware of? In addition, how do we verify such a spacecraft?
 - Mission complexity and operations in unknown environments challenge current V&V practices, and suggest that new V&V strategies will be needed for resilient systems.
- *Hierarchical vs. flat architecture*
 - Hierarchies can be used to manage the complexity of a system by dividing it into layers. The principle is very general, and can be applied to ideas (network protocols such as Open Systems Interconnection (OSI)), structures (the use of walls and floors to partition a building), and processes (build subassemblies before the final assembly). It has been argued as not only commonly occurring in nature (Doyle and Csete, 2011), but as a logical construct (Simon, 1962).
 - In this tradeoff, we express the question of how much hierarchy is needed. When conveying the functionality of a system to another human, hierarchy provides a mechanism by which we can organize and communicate ideas, at the cost of possibly ignoring important cross-hierarchy details. In contrast, a decision-making algorithm might be capable of considering more simultaneous possibilities, effectively flattening the hierarchy.

Resilient Architecture Principles

One of the most important aspects of architecting is conveying a unified understanding of the system to all those who will develop and use it. Towards this end, principles provide the foundational ideas upon which a system is built. In order to distill the key principles of architecting resilient systems, the Architecture Focus Group surveyed papers, books (Maier and Rechtin, 2009), and architectures of both resilient and non-resilient systems. In this section, we specifically address only those principles that we believe to be important for resilient systems. We assume that general principles of architecting, such as controlling complexity to promote understanding, providing a coherent rationale for requirements, and its importance as an overtly distinct and sustained effort (Rasmussen and Muirhead, *A Case for Model-Based Architecting in NASA*), are equally important and continue to apply.

- *Options*
 - Increasing the diversity and/or redundancy of options available to a resilient system provides the resources needed to recover from failures, tackle new challenges, and meet unknown situations. Diversity provides resilience to unknowns by providing a breadth of functionality (e.g., a child who falls from a bike and can't walk can still use his/her hands to call for help). Redundancy provides resilience through failures (e.g., a power surge in one computing core could cause another computing core to come online, or multiple replicated cores to continue the computations).
 - In the particular case of space missions, where resources are expensive to launch and maintain, perhaps the best way to provide options is through overlapping functionality, a blend of diversity and redundancy. For example, a satellite with both a high-bandwidth antenna for transmitting images and a low-bandwidth antenna for telemetry could use its processor and the low-bandwidth antenna to transmit digested information if the high-

bandwidth antenna fails. In this case, while both antennas have different degrees of functionality, they share enough in common to support another purpose.

- *Modularity and interfaces*
 - In order to more easily use our options, they should be modular and support some uniformity in their interfaces. To say it another way, modularity implies some degree of interchangeability, and therefore some uniformity to their interfaces.
 - Biologists Gerhart and Kirschner coined the phrase “constraints that de-constrain” (Kirschner and Gerhart, 2005; Gerhart and Kirschner, 2007). This phrase expresses the naturally occurring phenomenon in which the presence of constraints can actually enhance the ability of an animal to survive, adapt, and evolve. For example, bacteria use ribonucleic acid (RNA) to encode their genes, but can use horizontal gene transfer to swap segments of RNA between individuals, allowing a bacterial colony to quickly adapt, and ultimately evolve through the persistence of those fragments. In the man-made world, the Universal Serial Bus (USB) interface has allowed a variety of peripherals to thrive, the Transmission Control Protocol/Internet Protocol (TCP/IP) supports a wide range of applications, and LEGO® blocks allow children to explore an unimaginable set of combinations.
 - A consistent, uniformly applied interface to different elements of a space mission would allow a single mission to swap between options in order to adapt, and enable the reuse of technologies from one mission to the next.
- *Ability to choose*
 - How a particular resilient system chooses between options can vary greatly and provide different degrees of resilience. Mechanical options such as self-healing materials and overflow pipes could be considered a type of choice driven by physical means. From a cognitive perspective, reflexive behaviors, voting systems, fault-tolerance methods, decision making, planning, and, of course, humans can also provide the ability to choose.
- *Characterization of uncertainties*
 - In order for us to talk about a resilient system we need to be able to answer the question, “What is it resilient to?” For example, a wallet can be considered resilient to its daily abuses of being folded and unfolded, but we would not generally consider a wallet to be resilient to fire. When architecting a resilient system, it is important that the developers, operators, and end-users know the tolerances of the system.

4.1.3.3 *Directions of Future Work*

The work of the Architecture Focus Group identified numerous areas for future research and development in the area of architecting resilient space systems.

- Development of methodologies and systems engineering frameworks for architecting resilient systems; a representative example of such a framework has been proposed in (Rasmussen and Muirhead, *A Case for Model-Based Architecting in NASA*)
- Development of architectures for resilient space systems; two such architectures have emerged from our study:
 - The Resilient Spacecraft Executive proposed in the KISS Technical Development proposal submitted by Murray, Ingham, et al. entitled “Resilient Space System Architectures: Demonstration of Risk-Aware Hibernation Capabilities using Earth-Based Analogues” (described in Section 4.4.1 of this report).
 - The hierarchical flight software architecture concept proposed by Gostelow, Doyle, and Ingham, described in Appendix F of this report, which inherits and extends concepts from another resilient spacecraft architecture, JPL’s Mission Data System (Ingham, Rasmussen, Bennett, and Moncada, 2005).

- Development of architectural analysis tools, e.g., for performing tradeoffs between system attributes like execution time and mission flexibility; such tools will be used to inform system designers about the appropriate allocation of capabilities to different layers in a specified architecture, given a particular system, operational environment, and mission context. This advancement has been proposed as part of the KISS Technical Development proposal submitted by Murray, Ingham, et al.

4.2 Insights from Lightning Talks

Toward the end of both workshops, we had all participants present lightning talks. We labeled these lightning talks because each one was limited to two minutes and a single slide. This had the benefits of allowing all participants a chance to present their thoughts and the time constraint required a measure of focus that encouraged presentation of the most significant ideas. There were two motivations for doing these talks: (1) provide each participant a last word to wrap up the workshop, and (2) compile a set of slides representing a collective view on the subject of resilience from which we could draw themes and possibly reach new conclusions. This section represents a collective view of the lightning talks presented. All the slides were reviewed and several themes picked out based on ideas that were repeatedly expressed in the talks.

From the first workshop lightning talks, we isolated several key ideas that seemed explicitly apparent: **What are the metrics? Technologies borrowed from other domains are useful.** Further, a set of technologies that were called out in discussions during the first workshop also showed up in the lightning talks for this workshop.

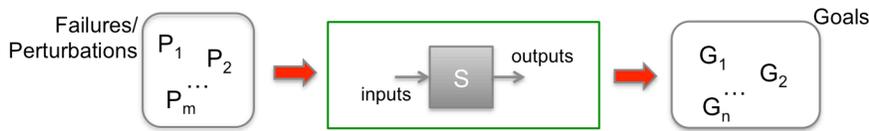
4.2.1 What are the Appropriate Metrics?

Any proposal will have to show quantitative value so this is needed for a technology program to be funded. Kim Gostelow presented a concept for utilizing the concept of an objective function that characterizes the success of a system in reaching defined goals. Then, if the notion of the goals being perturbed is added and measured using our objective function, it might be possible to analytically measure the sensitivities of various systems to perturbations, and thus characterize a metric for resilience of systems (Figure 4-2). John Doyle is also developing various theories for optimizing architecture using analytic means. Perhaps these could be applied to current missions showing varying resilience capabilities.

4.2.2 Borrow Technologies from Other Domains

A strategy that appeared repeatedly in both workshops was to consider applications of techniques for resilience in other domains. Basic technologies to implement resilience in space systems already exist in other domains (e.g., AUTomotive Open System ARchitecture [AUTOSAR] (Heinecke et al., 2004), Robot Operating System [ROS] (Quigley et al., 2009), Integrated Modular Avionics [IMA] (Morgan, 1991), UAVs; see *capabilities results*). How can progress and/or even models and code developed for other domains (e.g., cars, planes, Earth-based robots) be translated toward resilient spacecraft applications?

4.2.3 Observations on Technology



Objective function $F = \sum a_i S(G_i)$ where
 $S(G_i)$ = degree of success in achieving goal G_i and
 a_i = weight for goal G_i

If systems A and B are subjected to the same perturbations and $F_A > F_B$, then system A is *more resilient than* system B.

Parameterize perturbation P_i as $P_i(k)$. Consider $F_A(k)$. The derivative $\frac{dF_A(k)}{dk}$ is the *sensitivity* of system A to perturbation P_i . System A is *more sensitive* to perturbation P_i than system B if $\left| \frac{dF_A(k)}{dk} \right| > \left| \frac{dF_B(k)}{dk} \right|$

Figure 4-2: Towards a Measure of Resilience (Gostelow) Lightning Talk

Fundamental barriers to resilience technology adoption exist due to a lack of common conceptual foundation. Tools are being developed under programs such as Adaptive Vehicle Make (AVM)/META (DARPA Tactical Technology Office, n.d.) that will enhance passive (or latent) resilience; however, more efforts are needed. In particular, more techniques are needed to enable better exploration of the space of uncertainty and its impact, including machine learning, search-based techniques, probabilistic reasoning, and more advanced goal-based modeling. Investment in new methodologies for managing and analyzing the information is needed (e.g., methods for Model-Based Systems Engineering [MBSE]). Architectural analysis and metric tool development are needed. There is a need for formal and analyzable DSLs for specification of (discrete) mission performance, concept of operations (conops), and system/environmental uncertainty. Notations such as set-based languages will be required (i.e., specification of partial orders, lattices, etc.). There is a need for a collection of formal and lighter weight reasoning techniques to be used at different stages of the design process (e.g., architecting, validation, verification), as well as during run-time operation. Given the increasing complexity of these systems, these reasoning techniques should be applicable to models as well as code, and be applicable at design time and at run time. Advanced synthesis tools for ‘correct-by-construction’ designs are needed.

It was noted that spacecraft in general could utilize better time-enabling rapid autonomous reaction, parallel functionality, better onboard power management, smarter downlink, and operation under varying environmental conditions, etc.

Finally, during the first workshop, the bacterial biosphere was identified as being a unique example of resilience. Bacterial resilience after environmental perturbation was discussed. It was explained that such bacterial systems can present many related species, which, over time, may present unique and novel genetic attributes compared to near relatives allowing evolvable characteristics that can adapt to certain environmental changes. New resistant species can become the dominant majority in the face of many species dying off (becoming extinct). In this way, a bacterial biosphere is an illustration of an extremely resilient system. The bacterial biosphere has inspired new self-correcting, evolving, and adaptive software architectures that have new resilience to changes in the environment.

The second workshop lightning talks presented considerably different themes than the first workshop. Discussions continually mentioned architectural versus capabilities trades and V&V was a recurring theme as well. Notions of spacecraft functionality to perform automated self-repairs (i.e., self-healing, self-awareness, collective redundancy) were discussed but only in a few of the lightning talks. Nevertheless, this is an important area as it relates to concepts of redundancy where a system uses extra

parts to fix itself through fine-grain repurposing. Finally, several talks mentioned possible demonstrations of resilience but also discussed the political barriers that stand in the way of realizing true resilience within our flight systems (see *politics and demonstrations*). Each of these themes is discussed in the following subsections.

Verification & validation. Can we capture an interface between learning and aspects of operation and use this to formulate system design policy as a central feature throughout lifecycle used to establish trust early? Can we set up early V&V? V&V is critical and presents a unique challenge to ensure trusted resilience. Enhanced behavioral modeling integrated with the V&V process is needed. Resilience, assurance, and V&V must be inherent parts of model creation/modification to handle uncertainty in systems resulting from environment changes (e.g., resilient by construction). The V&V envisioned is not traditional—we are thinking of new forms of design-time and run-time V&V methods. Design-time V&V is integrated into the design process and enables scalability with system size and complexity as the system grows while guaranteeing decent coverage in the face of uncertainty resulting from intractable system state space. Run-time V&V is embedded into the system for execution during run-time and is a useful approach for verification of evolving systems and in situ validation of learned capabilities. In a learning environment, V&V is a challenge, since it is not obvious how to define specifications for verification of the evolving system. Both these approaches provide rapid V&V to support changing modular software (i.e., agile validation) that is essential for testing new complex systems (to demonstrate technological readiness) and establishing trust in autonomy capabilities needed for resilience in unknown environments.

Politics and demonstrations. Releasing a resilient architecture framework as an open-source project is a compelling demonstration vehicle. We could target possible early adopters to help kick start such an effort and target the robotics community by providing a few useful algorithm implementations. The group struggled to decide on a useful platform that would be large enough to be taken seriously. A possible demonstration using a CubeSat where behavior of planned and randomized failures of components would be implemented was presented. Small doesn't mean lack resilience, perhaps the problem is that most CubeSats are programmed by students without resilience as a primary goal.

Another approach toward a demonstration would be focus on a different nonspace domain, perhaps a 6-month underwater mission with relatively cheap hardware (failures will happen). This would provide a demonstration of what can be done and create a pull from scientists to overcome reluctance of engineers in the spaceflight community.

Many key technologies (capabilities) for engineering resiliency already exist. For example, we know how to build systems that do online planning, making use of onboard models, including structured goals, fault management, and learned behavior. Architectures that support these capabilities already exist (e.g., autonomous vehicles). We have demonstrated driving in environments that are not known in advance and have lots of mechanisms to ensure safety in the presence of failures. Therefore, efforts to demonstrate these capabilities in using alternate funding sponsors are essential.

4.3 Roadmap for Technical Development

As described above, one of the key outcomes from the study was the development of a roadmap of key resilience-related capabilities. This roadmap was developed as a product of the reference missions exercise, which captured the mission resilience needs for three selected reference missions; identified and prioritized the capabilities, technologies, and architectural characteristics to meet these needs; and developed a set of roadmaps for each of the reference missions. Finally, these three roadmaps have been analyzed and synthesized to extract key patterns of capability that are prime targets for investment by technology development programs. In this section of the report, we will first summarize the individual roadmaps for each reference mission, and then discuss the synthesized capability patterns.

4.3.1 Venus Lander

The capability roadmap for the Venus Lander mission is presented in Figure 4-3. The key high-level mission capabilities of safe landing and landed science operations were identified as drivers for resilience. To achieve the safe landing capability, we have identified three resilience-related enabling capabilities: *landing hazard recognition*, *landing hazard avoidance*, and *graceful degradation*. In the context of safe landing, graceful degradation relates to the ability of the lander spacecraft to continue operating through faults and unexpected environmental interactions, using system resources in a best effort manner to get the spacecraft safely to the planetary surface. Rather than gracefully degrading, recent missions have disabled their fault protection capabilities for entry, descent, and landing (EDL), and have relied on a carefully designed and painstakingly developed control sequence to provide some level of robustness through redundancy management. Landing hazard recognition, landing hazard avoidance, and graceful degradation capabilities are discussed in more detail below, under *Synthesis of Capability Patterns*.

To achieve the landed science operations capability, we have identified the following enabling capabilities: detection of interesting phenomena, onboard science analysis, onboard science planning, sampling hazard recognition, sampling hazard avoidance, and graceful degradation. The surface mission will achieve resilience through onboard science autonomy because of the short lifetime of the spacecraft in the Venus environment. In this context, the graceful degradation capability is absolutely critical to enable the mission to squeeze the most science possible out of its limited lifetime. Graceful degradation will be achieved via a multifaceted strategy, including fault-tolerant control to robustly accomplish the current science goals, diagnosis of failures and prognosis of incipient failures, and a range of recovery options including system reconfiguration to accommodate the degradation and task replanning to shed

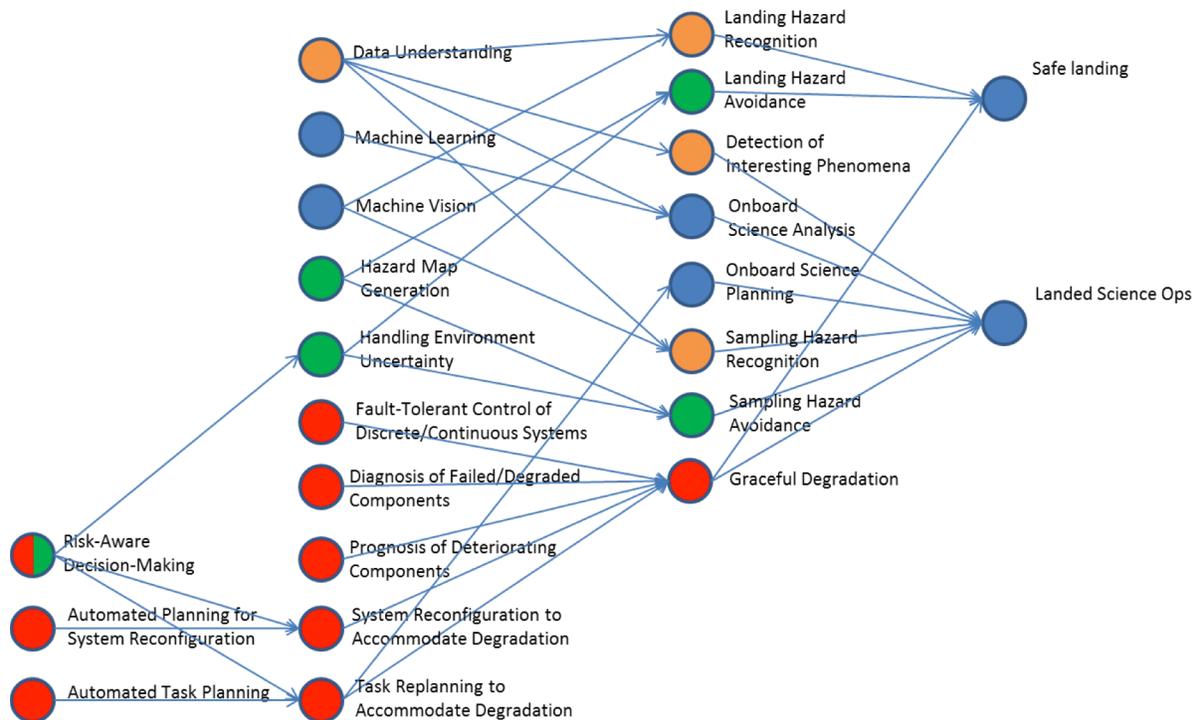


Figure 4-3: Capability Roadmap for Venus Lander Mission. Key mission-level capabilities are identified on the right-hand side of the roadmap. The arrows incoming to each capability node indicate supporting capabilities that enable that capability. The red, orange, and green color coding is used to identify particular capability dependency patterns that are common to the three reference mission roadmaps we developed. Blue nodes indicate other supporting capabilities for the Venus Lander mission that are not common across the three reference missions.

science goals that are no longer achievable due to the degraded capability.

4.3.2 Mars Sample Return

The MSR campaign encompasses a particularly complex set of missions, each requiring multiple system elements and multiple mission-critical activities. The capability roadmap for this mission is presented in Figure 4-4, with particular focus on the first mission of the campaign, the sample collection and caching mission. In this mission, the key high-level capabilities include safe precision landing, robust telecomm link maintenance, long-distance unattended traverse to/from the sampling sites, and sample collection and caching. Additional high-level mission capabilities for the follow-on sample return missions are also identified (cache manipulation, launch from mars surface, and on-orbit rendezvous with orbiting sample), but due to the limited scope of the study, these high-level capabilities are not further elaborated into enabling capabilities. With respect to the safe precision landing capability, the enabling supporting capabilities are the same as those called out for the Venus Lander’s safe landing capability. An additional capability that enables a high-precision landing, terrain-relative navigation, is not currently called out in the roadmap, but is not deemed a critical omission because this capability is the target of ongoing investment at NASA and elsewhere.

Robust telecomm link maintenance refers to the need for a resilient communication link to landed and landing assets on the surface of Mars. The importance of this capability was underscored for the Mars Science Laboratory (MSL) EDL: a few weeks prior to arrival of the MSL at Mars on August 5, 2012, Mars Odyssey, the critical orbiter identified for relay of MSL EDL telemetry to Earth, suffered an onboard fault that caused it to enter safe mode on July 11, 2012. Had the operations team not been able to

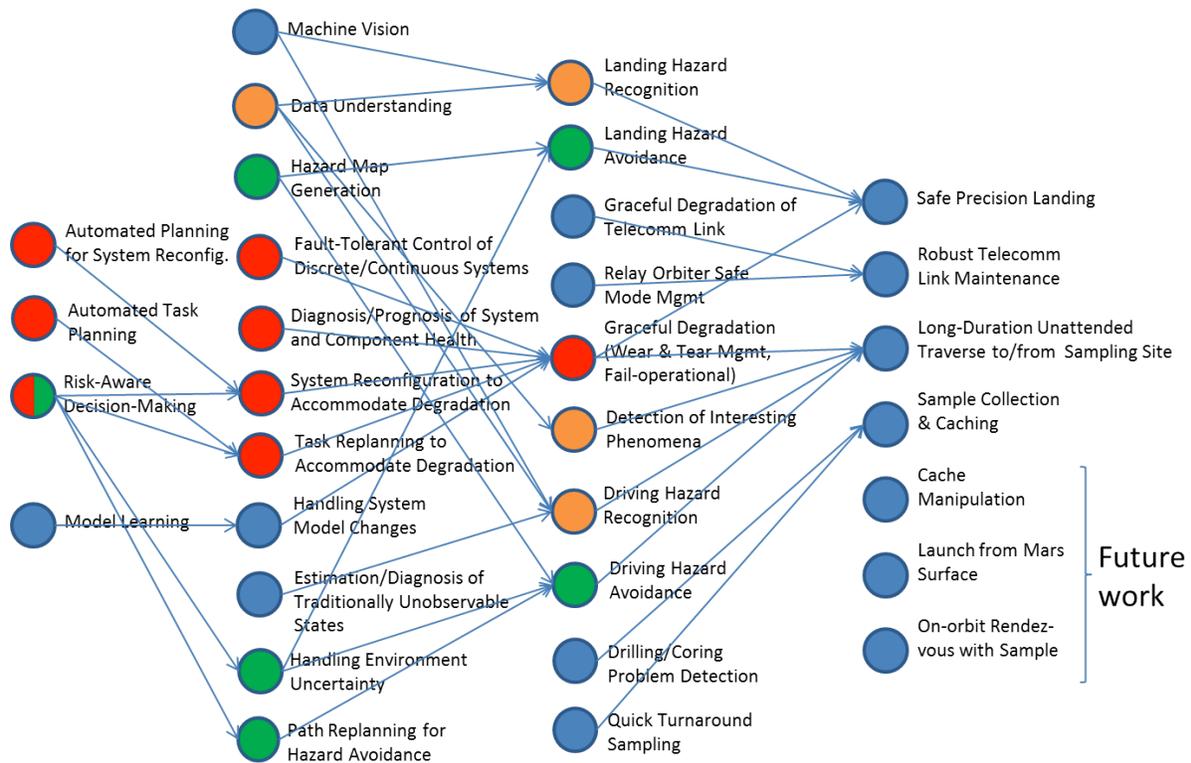


Figure 4-4: Capability Roadmap for MSR Mission. Key mission-level capabilities are identified on the right-hand-side of the roadmap. The arrows incoming to each capability node indicate supporting capabilities that enable that capability. The red, orange, and green color coding is used to identify particular capability dependency patterns that are common to the three reference mission roadmaps we developed. Blue nodes indicate other supporting capabilities for the MSR mission that are not common across the three reference missions.

recover the spacecraft quickly enough (or if the fault had occurred closer to the critical MSL EDL event), the orbiter would have been unable to reposition itself in time to support EDL relay communications, Odyssey would have arrived over the landing area about two minutes after Curiosity landed, and NASA would have likely lost critical engineering telemetry from MSL during its descent and landing on the surface. While MSL's landing success was not at risk, one of the mission requirements for capture and analysis of EDL telemetry for the purposes of state reconstruction, would not have been met. Although not further elaborated in the roadmap, robust telecomm link maintenance is enabled by supporting capabilities such as graceful degradation of the orbiting relay assets (e.g., whereby the relay orbiter might avoid entering safe mode leading up to a critical relay support activity like EDL, especially if the triggering fault was diagnosed to be non-mission-threatening), or of the direct-to-Earth (DTE) communication link.

Long-duration unattended traverse to/from sampling sites is identified as an important capability for surface roving missions like MSR from a resilience standpoint. This mission will likely involve long traverses from the landing site to the sampling sites identified by the scientists as most interesting. The current operational paradigm involves a substantial operations team, which would be unnecessary if the rover could be trusted to largely take care of itself during comparatively routine activities like driving. While completely unattended traverse operations is unrealistic, entrusting the rover to manage many of the off-nominal situations it encounters while driving would enable a much leaner operations team to be at the console. This would enable a reduction in operations costs, and/or enable a significant portion of the operations team to focus on training for the more risky upcoming sample collection and caching operations, resulting in reduced risk for those critical scenarios. As shown in the roadmap, long-duration unattended traverse is enabled by capabilities that include driving hazard detection and avoidance, detection of interesting phenomena (which enables the rover to be alert to its surroundings; from an engineering perspective, it would recognize anomalous situations, such as changes in inclination or wheel slippage, and from a science perspective, it would not drive blindly past particularly interesting science sites), and graceful degradation (which enables the rover to be resilient to the wear and tear that would inevitably occur during long surface missions, as evidenced by the Mars Exploration Rovers Spirit and Opportunity).

Finally, *sample collection and caching* is a driver for important resilience-related capabilities in the areas of drilling/coring problem detection and quick-turnaround sampling. However, due to the limited scope of the study, these capabilities are not further elaborated into enabling capabilities.

4.3.3 Trojan Tour and Rendezvous

The capability roadmap for the TTRV mission is provided in Figure 4-5. In this mission, the key high-level capabilities include hibernation operations during the multiple cruise phases (both cruising to the Trojan belt, and cruising between asteroids in different segments of the belt), and science operations during asteroid encounters. In particular, streamlining hibernation operations is identified as a valuable low-hanging fruit for near-term investment in resilience capabilities (an analogous, but even less challenging problem than the long-duration unattended traverse capability described above). For long-duration missions like TTRV, our current operational paradigm for large missions that employ a large operations team throughout the mission operations subphases will make it very challenging to achieve the significant cost reductions imposed by current NASA planetary science budgets. Our vision for resilience includes as an early accomplishment the development of a highly autonomous hibernation capability that can enable spacecraft to fend for themselves during long periods of routine and largely uninteresting operations, and be operated by a very small crew on the ground. The hibernation operations capability is enabled by a variant of the common graceful degradation capability, whereby the spacecraft would be endowed with the ability to diagnose and recover from many of the faults that cause spacecraft to go into safe mode today.

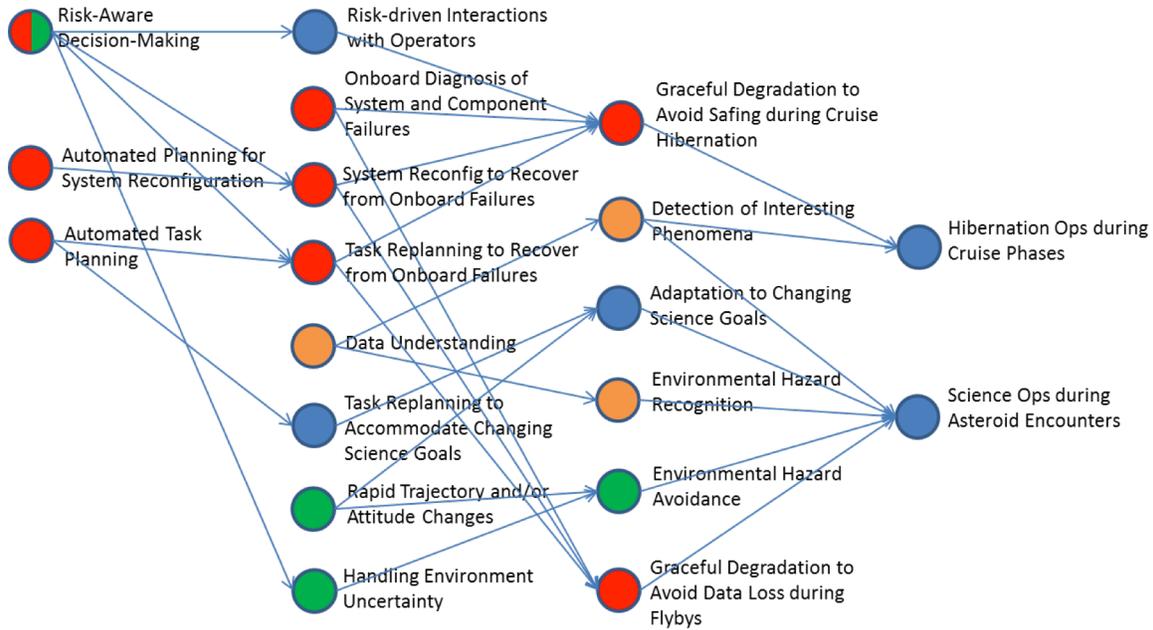


Figure 4-5: Capability Roadmap for TTRV Mission. Key mission-level capabilities are identified on the right-hand-side of the roadmap. The arrows incoming to each capability node indicate supporting capabilities that enable that capability. The red, orange, and green color coding is used to identify particular capability dependency patterns that are common to the three reference mission roadmaps we developed. Blue nodes indicate other supporting capabilities for the TTRV mission that are not common across the three reference missions.

Key to this is providing the spacecraft with the ability to make risk-informed decisions (risk-aware decision-making) and to engage with the operators on Earth when it assesses that a specified threshold of risk to the currently executing, high-level mission sequence has been exceeded. Recovery can take many forms, from simple side-swapping of redundant hardware or device resets, to more involved replanning of the current nominal sequence of tasks, subject to the constraints of the mission (e.g., a SEU-induced reset of an onboard science processor would not require safing of the spacecraft, because the symptoms associated with the reset would be recognized as non-threatening, but could require the spacecraft to reschedule a planned background science activity involving that processor to be performed after the upcoming telecomm link with Earth). Another key capability associated with unattended hibernation operations is the onboard detection of interesting phenomena. This would enable the hibernating spacecraft to “snooze with one eye open” (i.e., be on the lookout at some regular frequency for important science events that would be of likely interest to the scientists on the ground). Depending on the urgency of the type of science observation, the spacecraft could decide to call home at the earliest opportunity, or it could decide to store any interesting but non-urgent results in memory for downlink at some later time.

Because of the highly uncertain and likely dynamic environment encountered by the TTRV mission,³ agile science operations during asteroid encounters is enabled by a set of capabilities that include the onboard detection of interesting phenomena, the ability to adapt to changing science goals, and the ability to gracefully degrade to avoid safing if at all possible during one-shot flybys of interesting asteroids. Similarly, the highly uncertain and dynamic environment implies that the spacecraft operations would be less risky with an onboard capability to detect and avoid environmental hazards (e.g., small asteroid debris, or outgassing plumes, if any are encountered). These capabilities are of broad applicability across the set of reference missions, and thus will be discussed in the TTRV context in more detail in the synthesis section below.

4.3.4 Synthesis of Capability Patterns

We have identified threads or patterns in the capability roadmaps that have wide applicability and are needed across diverse mission types. This set of capability patterns are identified in the capability roadmap figures by the red, orange, and green node colorings, and are summarized below and in Figure 4-6. Further analysis of the roadmaps is ongoing, and any additions or revisions to the identified patterns will be discussed in updates to this report.

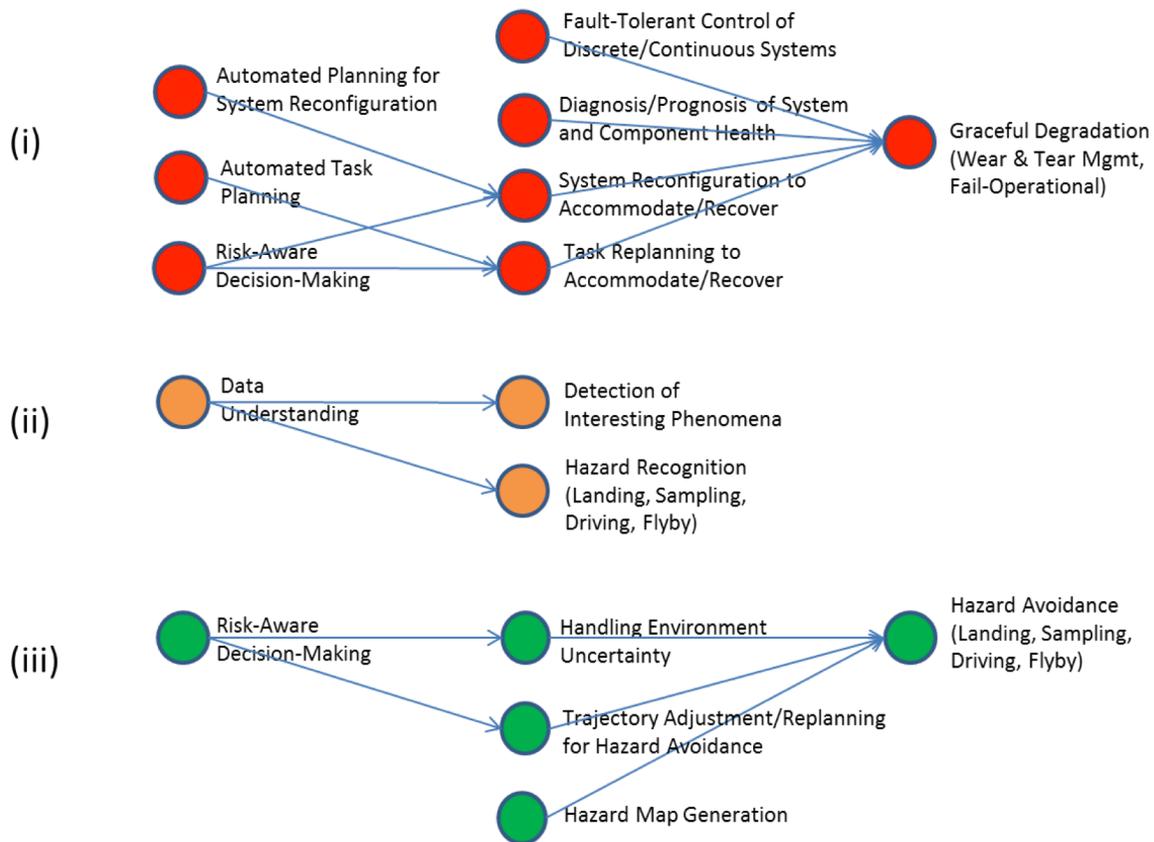


Figure 4-6: Common Capability Patterns for Resilience

³ The Jupiter-orbit Trojans are dark bodies of irregular shape, varied sizes from ~200 km down to ~1 km or less, with unknown rotational periods on the order of hours, sometimes coupled in binary pairs, grouped in clusters that tend to overlap and merge with the overall swarm throughout their orbits (“Jupiter Trojan”, 2013).

4.3.4.1 Graceful Degradation Pattern

This first capability pattern (shown in red) is crucial to realizing the vision of resilient space systems. Graceful degradation, by our definition, is the ability to continue operating through faults and unexpected environmental interactions. Not fail safe or fail fully operational, but a level in between, which uses system resources in a best-effort manner to achieve as many of its goals as possible, despite the occurrence of faults, SEUs, resets, etc. This encompasses general wear and tear management (i.e., the ability of a spacecraft to tend to its own failures and degraded functions, either by performing repair actions or coming up with creative workarounds) and best-effort fail-operational capabilities (i.e., the ability of a spacecraft to complete an important mission activity that would otherwise be lost due to a component failure or system safing response). As shown in Figure 4-6 (i), graceful degradation is enabled by a rich set of supporting capabilities, including fault-tolerant control of discrete/continuous systems, diagnosis and/or prognosis of system and component health, accommodation of or recovery from failure/degradation via automated system reconfiguration or task replanning, and risk-aware decision-making. Many of these capabilities can be implemented either on the ground within software tools used by the operations teams, or as autonomous capability onboard the spacecraft. Our emphasis on the latter should not be construed as dismissive of the value of the former—indeed, wherever it makes sense, these capabilities will be instantiated as ground-based capabilities to validate them, as a precursor to eventual onboard deployment.

Here are some examples of graceful degradation capability, with the elaborated enabling capabilities in *italic*:

- The ability of a Mars surface rover to recognize a stuck front wheel motor (*diagnosis*), and recognize that its driving objective can best be accomplished by turning the rover around and dragging the stuck wheel, instead of pushing it (*automated planning for system reconfiguration*). This scenario actually occurred during sol 778 of the MER Spirit’s mission (Figure 4-7) (“Spirit (rover)”, 2013), except that operators on the ground, and not the rover itself, performed the reasoning required to



Figure 4-7: The MER Spirit’s right-front wheel failed on sol 778.

determine that driving backwards was an appropriate mitigation for the failure.

- The ability for a Venus Lander to determine that one of its computers is failing due to the tremendous heat on the Venusian surface (*diagnosis*), and to reallocate the failing computer’s control functions to other computers in its distributed processing architecture, to maximize the precious lifetime of the spacecraft (*automated planning for system reconfiguration*).
- The ability of the TTRV spacecraft (Figure 4-8) to autonomously: (i) recognize the symptoms of incipient failure for one of its science camera instruments (*diagnosis/prognosis*), (ii) assess that the risk of losing the valuable science opportunity is high if we stick to the current flyby sequence, and make the executive decision to turn off the unreliable science camera during a particularly short-duration asteroid encounter flyby without

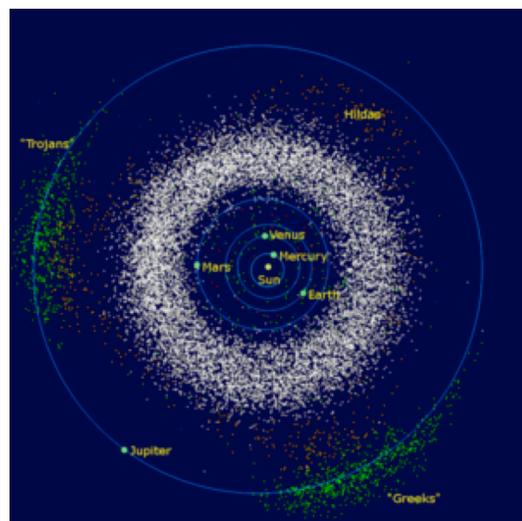


Figure 4-8: A Trojan Tour and Rendezvous mission will visit the Trojan asteroids (shown in green), which are in front of and behind Jupiter’s orbital path.

safing the spacecraft (*risk-aware decision-making*), and (iii) compensate for the loss in that particular set of science data by replanning a more extensive set of observations using the other onboard instruments (*automated task replanning*).

- The ability for a hibernating TTRV spacecraft to recognize that one of its reaction wheels is beginning to show signs of degraded behavior (*diagnosis/prognosis*), and to adjust its onboard control policy to reduce reliance on this reaction wheel, in favor of the spare skew wheel or onboard thrusters, thereby hopefully extending the useful life of the degrading reaction wheel and saving it for the critical encounter flybys (*fault-tolerant control of discrete/continuous systems*).

4.3.4.2 Data Understanding Pattern

This second resilience capability pattern (shown in orange in Figure 4-6 (ii)) is a fundamental enabler for closing decision-making loops onboard the spacecraft, for both science and engineering analysis. Data understanding is a general capability that encompasses the interpretation of data into actionable information. This capability covers a set of capabilities ranging from simple pattern recognition, data classification, and data fusion, to more sophisticated capabilities for abstracting raw data into a model, which can be used to make more significant science decisions. From a science perspective, this includes feature detection in images, which can be used to identify scientifically interesting targets for further investigation (e.g., unusually shaped or colored rocks) or to recognize scientifically significant changes in the observable surface of a planetary body (e.g., lava flows from volcanic eruptions, or new craters). From an engineering perspective, very similar techniques can be used to analyze light detection and ranging (LIDAR) scans or images taken by a spacecraft descending toward a planetary body to identify landing hazards, and to interpret stereo images taken by a mobile rover to recognize driving hazards.

Examples of initial deployments of data understanding capability in recent missions include:

1. The autonomous science performed onboard the EO-1 mission (Chien et al., 2005), which enabled the spacecraft to autonomously detect and respond to dynamic scientifically interesting events observed from low Earth orbit (and in the process save approximately \$1 million per year in operations costs);
2. The AEGIS software deployed on the MER Opportunity (Estlin et al., 2012), which analyzes images onboard, detects and prioritizes science targets in those images, and autonomously obtains novel, high-quality science data of the selected targets within 45 minutes, with no communication back to Earth required (Figure 4-9); and
3. The autonomous navigation performed by the Deep Impact mission (Kubitschek et al., 2007), which closed the loop around images taken by the impactor spacecraft to autonomously track and successfully impact comet Tempel 1.

In addition to these mission deployments, data understanding is a critical aspect of the terrain-relative navigation (TRN) and hazard detection systems that are currently under development as part of NASA's Autonomous Landing Hazard Avoidance Technology (ALHAT) program (Epp, Robertson, and Brady, 2008), which is being funded by the Office of the Chief Technologist to develop sensor hardware and software technology

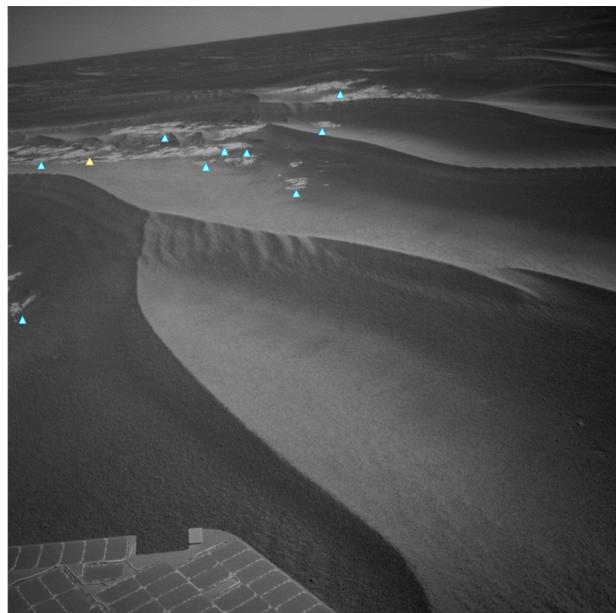


Figure 4-9: Science targets automatically detected by the MER AEGIS software on the MER Opportunity in April 2010.

for landing spacecraft autonomously, safely, and precisely on planetary bodies. These recent deployments and developments have enabled remarkable scientific and engineering accomplishments, but represent just the tip of the iceberg in terms of data understanding capability. The development of more sophisticated data understanding capability is becoming increasingly important as we develop truly intelligent spacecraft whose resilience is dependent on their ability to make more informed decisions onboard.

4.3.4.3 Hazard Avoidance Pattern

The third common resilience capability pattern (shown in green in Figure 4-6 (iii)) represents a critical capability for missions that a spacecraft must interact with a highly uncertain environment, such as those involving in situ exploration of planets and moons, or close interactive rendezvous with primitive solar system bodies. This capability represents the counterpart to the hazard detection/recognition capability discussed under the *Data Understanding Pattern* section above. The most common application of hazard avoidance capability in current space missions is in the context of surface rovers, which are required to avoid running into large rocks that could impede progress, or worse, damage the rover. Similarly but closer to home, the growing fleets of fully autonomous automobiles (inspired by DARPA's recent Grand Challenges) rely on sophisticated hazard avoidance capability to enable them to safely navigate on and off roads with obstacles of many shapes and sizes, including moving pedestrians, cyclists, and other vehicles.

Another important application of hazard avoidance capability is for spacecraft landing. Depending on the planetary body being targeted, the physics of the descent and landing problem can have particularly challenging requirements on timing (urgency), computation and resources (e.g., propellant), and thus represents a particularly prominent source of mission risk. As mentioned above, NASA's ALHAT program (Figure 4-10) is making significant investments in algorithms for efficiently computing hazard maps and quickly interpreting them to select a safe landing site.

Yet another application variant of hazard avoidance capability arises in the area of scientific sampling (e.g., using a robotic arm to dig into soil or drill into rocks to uncover interesting features for scientific analysis). Missions like the Mars Phoenix Lander and MSL have gone to great lengths to carefully consider and select appropriate sites/targets for sampling, not only to maximize the science return, but also to avoid the potentially mission-ending situation where the scoop or drill instrument gets inextricably caught in the ground or the rock. In current missions, the sampling hazard recognition and avoidance loop is primarily closed by human operators on the ground. Technology has been developed and demonstrated on the MER mission that enables automatic arm placement based on onboard hazard analysis (Hayati et al., 2007). Future missions like the short-lived Venus Lander or Lunar South Pole Sample Return mission will need to build on this class of technology to make arm placement risk assessments and decisions in situ.

As shown in Figure 4-6 (iii), hazard avoidance is enabled by several supporting capabilities that include handling of environmental uncertainty and risk-aware decision-making. This is not to say that these are the only supporting capabilities required to implement a hazard avoidance capability, but rather, these are critical capabilities that are not already the focus of significant research and development investment (unlike hazard map generation, or path replanning for example), and which will enable significant advancements in the sophistication of hazard avoidance capabilities we will need on future missions. For example, a future application of risk-aware decision

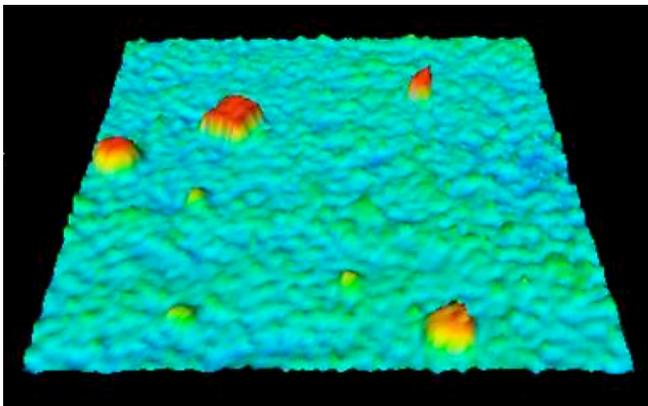


Figure 4-10: LIDAR elevation map with detected hazards, generated by the ALHAT technology.

making, handling of environmental uncertainty, and hazard avoidance capabilities is envisioned for missions being conceived to explore distant primitive bodies, at destinations like the Trojan asteroid field, which share the orbit of the planet Jupiter around the Sun. A mission like TTRV will encounter an unknown but undoubtedly highly dynamic environment. This mission will benefit greatly from a spacecraft with the onboard flexibility to cope with high uncertainty, assess the risk associated with operations in its environment, make local decisions that are informed by risk, and avoid low-albedo debris (small asteroids or outgassed material, if present) in the asteroid cloud. Flying such capabilities will likely reduce the risk associated with the mission, which, in turn, will likely enable a more scientifically interesting but challenging trajectory (e.g., closer flybys).

We have developed a set of capability roadmaps pertaining to the three reference missions, selected for their representative coverage of the types of space missions envisioned for the future. From these three roadmaps, we have extracted a few common capability patterns that would be appropriate targets for near-term technical development, one focused on graceful degradation of system functionality, a second focused on data understanding for science and engineering applications, and a third focused on hazard avoidance and environmental uncertainty. Continuing work is extending these roadmaps to identify candidate enablers of the capabilities, from the following three categories: *architecture solutions*, *technology solutions*, and *process solutions*. Following the study, future technology planning efforts will develop a technical development plan with specific milestones, for specific capability-enabling solutions.

4.4 Continuing and Future Work

The relationships and connections made during the study have enabled the study participants to move forward with collaborative ideas and proposals. Several participants have already teamed together to develop various proposals related to the study, and the co-leads have submitted a KISS Technical Development proposal to demonstrate some of the key technologies. These actions are the first of many expected steps in creating missions that use truly resilient spacecraft to generate breakthrough science.

A set of ongoing research and possible future collaborations is described in the following sections. The first section describes the follow-on work being proposed through the KISS Technical Development program, and the current work to be leveraged by this proposal. The second section describes other related research proposals and tasks being developed through other mechanisms.

4.4.1 KISS Technical Development Proposal

One promising thrust for future work is discussed in our KISS Technical Development proposal, titled “Resilient Space System Architectures: Demonstration of Risk-Aware Hibernation Capabilities using Earth-Based Analogues.” The proposal leverages areas of ongoing research at Caltech, MIT, WHOI, and JPL. The premise of the proposal and descriptions of areas of research are presented below.

Spacecraft to support missions described in the Planetary Science Decadal Survey (National Research Council, 2011) will need to satisfy a tremendously large number of requirements on resilience. To adapt to changes in spacecraft state and changes of the environment, a new class of spacecraft architecture, capable of reasoning to predict and avoid hazardous conditions, and recover from internal failures to ultimately meet critical science objectives in the presence of substantial uncertainties, is needed.

A two-year effort will take first steps to develop a highly innovative prototype software architecture, deployed as a Resilient Spacecraft Executive, that will endow a spacecraft with unprecedented levels of resilience, and then demonstrate how it can reduce risk and cost through capabilities such as highly autonomous spacecraft hibernation. This effort builds on and supports research efforts at Caltech, JPL, MIT, and WHOI. To achieve these objectives, we will mature and integrate key technologies in (i) goal-directed and risk-aware execution/decision-making, (ii) correct-by-construction control policy synthesis, and (iii) model-based systems engineering approaches that facilitate development and trade studies of the underlying architecture.

To illustrate the breadth of applicability and versatility of our Resilient Spacecraft Executive architecture, we will deploy and demonstrate it on two very different platforms representing compelling Earth-based analogues of space systems:

- An autonomous underwater vehicle (AUV) testbed, as an analogue for deep-space missions with long hibernation phases (e.g., TTRV);
- A surface rover testbed, as an analogue to surface missions with long-distance traverses, (e.g., the 2020 MSM or other future MSR missions).

The development of the Resilient Spacecraft Executive will leverage substantial ongoing work as described below:

Caltech: Dr. Richard Murray's group at Caltech has several ongoing, industry-funded projects that were well-aligned with the study. They are developing theory and algorithms for model-based verification and synthesis of control protocols (Braman, Murray, and Ingham, 2007; Ozay, Topcu, Murray, and Wongpiromsarn, 2011; Xu, Topcu, and Murray, 2012). This includes development of specifications for desired system behavior and methods for synthesizing control laws to achieve the specified performance. The work being done for those other projects will use the proposed architecture and testbeds for demonstrations. This will allow the transfer of technologies being developed under external funds to be made available to the current project. There are currently four graduate students and two postdocs whose work is directly relevant, including two researchers who participated in the KISS study. All ongoing projects are nonproprietary and open source.

JPL: JPL has developed strategic institutional capabilities and expertise in Integrated Model-Centric Engineering (IMCE) (Bayer et al., 2010; 2011), which the JPL Engineering and Science Directorate has invested over \$2M in over the past 3 years, and which builds upon an even greater investment (over \$40M) over the past decade in MBSE capabilities, including the patented Mission Data System technology and the associated State Analysis methodology (Ingham, Rasmussen, Bennett, and Moncada, 2005). Furthermore, JPL has developed expertise and capability from current autonomy efforts. Most significantly, this includes the AEGIS software, which earned the NASA Software of the Year award in 2011 (Estlin et al., 2012). Finally, JPL has several rover assets, which have been developed over the past two decades that post-study efforts can use for proof of concept demonstrations. These include the Field Integrated Design and Operations (FIDO) and Athena research rovers, and various simulation and hardware-in-the-loop testbeds from the MER project.

MIT: Dr. Brian Williams's Model-based Embedded and Robotic Systems (MERS) group at MIT has a range of industry and government-funded projects centered on the problem of developing a common autonomous control architecture that is resilient, goal-directed, and model-based (Williams, Ingham, Chung, and Elliott, 2003; Williams and Ingham, 2002; Williams, Ingham, Chung, Elliott, and Hofbaur, 2004; Ono and Williams, 2008; Hofbaur and Williams, 2004). These independently supported projects are well-aligned to the goals of our study and the follow-on proposal, both with respect to developing a risk-aware executive, and a resilient architecture. There are currently four MERS graduate students whose work is directly relevant to extending model-based executives to be risk aware, and three additional MERS graduate students who are developing model-based execution elements that we will leverage for post-study activities. This includes two graduate students who participated in the KISS study.

WHOI: Dr. Rich Camilli's research group in WHOI's Deep Submergence Laboratory (DSL) specializes in developing in situ robotic technologies for oceanographic science. DSL is home to the National Deep Submergence Facility (NDSF), and develops/maintains/operates a fleet of over a dozen autonomous, remotely operated, and human-occupied submersibles (www.whoi.edu/main/ndsf). Camilli's ongoing NASA Astrobiology Science and Technology for Exploring Planets (ASTEP) research program is developing autonomous sensing techniques for characterizing environmental chemical indicators of life (Kunz et al., 2009), and DSL-MERS research collaborations (via Camilli and Williams) focused on advancing autonomy for NDSF submersibles. Camilli's lab also developed the Sentinel AUV glider

platform and mass spectrometer, which are potential platforms for resilience demonstrations. The extensive collaboration between oceanographic and space science robotics will facilitate a bidirectional knowledge transfer benefiting both research domains.

4.4.2 Other Tasks and Proposal Concepts

The study provided motivation and inspiration for various other tasks and proposal concepts. The following is a listing of activities identified to date.

JPL Agile Science Strategic Initiative: David R Thompson will lead a 2014 research task under the Agile Science Strategic Initiative (Thompson et al., 2012), a JPL effort to prototype onboard autonomy for primitive bodies missions. Capabilities for maturation include dynamic science feature recognition and excision of pathological data collection errors, which were both capabilities discussed extensively during the workshops. Additionally, he will advise Melissa Bunte, a participant of the second workshop, on a new NASA postdoctoral fellowship to take place next year. This research will investigate new methods of measuring transient processes during small bodies and outer planets missions.

JPL NASA Institute for Advanced Concepts (NIAC) Proposal: Principal Investigator (PI) Adrian Stoica collaborated with KISS Co-Investigators (Co-Is) Michel Ingham and Leslie Tamppari to submit a NIAC proposal titled “TransFormers for Extreme Environments.” The proposal was selected for a NIAC Phase 1 award. They propose an enabling capability for operation in extreme environments such as permanently shadowed craters or planetary caves, a solution applicable to all types of in situ missions, which is to project and control a favorable microenvironment (around a rover) in the local area where exploration, exploitation, or human visits will take place. The systems that could provide such a capability are called TransFormers. The name suggests their two key properties: they transform the environment, and they adapt to needs by shape change/transformation. TransFormers are gossamer-thin (~100 microns) and fold compactly (~1 cubic meter unfolding to a ~10,000-square-meter area). Their body surface would embed reflectors and solar cells; they would also include antenna elements for communication, and actuation and control elements for shape change. TransFormers present a novel way of improving survivability in extreme environments, and enable new classes of missions, such as operations that involve long periods of time without direct solar input or radioisotope thermoelectric generators (RTGs), at massively reduced cost; missions to polar craters on the Moon and Mercury, or caves on Mars and the Moon would particularly benefit, with remote TransFormers providing illumination, energy, and communications.

JPL Topical Research and Technology Development (R&TD) Proposal Concept: Leonard Reder, PI, with Co-Is Cin-Young Lee (JPL) and Professor Tihamér Levendovszky (Institute for Software Integrated Systems [ISIS], Vanderbilt University) submitted an R&TD proposal titled “Aspect Oriented Programming for Flight Software.” Flight software (FSW) complexity is an ongoing issue for flight projects. JPL has partially addressed this complexity by modularizing FSW code along functional lines (e.g., device drivers, device managers, etc.). Modularization has resulted in limited success; FSW starts out as very clear and easy to understand, however, in the course of development, all kinds of ancillary services must be added such as commanding, telemetry, fault protection, and remote diagnostic capabilities to log run-time activities. These services are pervasive and impact virtually every part of the FSW often resulting in code entanglement and obfuscation in which these pervasive cross-cutting concerns cannot be cleanly separated from the basic FSW functionality that use them. Modification of any of these services necessitates additional changes across the FSW, reducing maintainability and reusability. Aspect-oriented Programming (AOP) is a new programming paradigm that provides abstractions for addressing cross cutting concerns. Our goal is to investigate refining FSW using AOP. The objective of this R&TD task is to demonstrate that AOP can dramatically simplify FSW by cleanly separating cross cutting concerns such as commanding, telemetry processing, fault detection, etc. from the functional code.

State Outcome Space-Based Spacecraft Fault Protection—JPL Topic Area R&TD Proposal Concept: Michael Sievers, PI, with Co-Is Michel Ingham, John Day, and David Bayard (JPL), and Professor Azad

Madni (University of Southern California [USC]), submitted a JPL R&TD proposal titled “State Outcome Space Based Spacecraft Fault Protection.” Many of JPL’s flight missions must operate robustly in harsh and uncertain environments, and these missions will need enhanced capabilities to reconfigure themselves in the presence of faults, whether human-induced or the result of environmental stresses. Several past systems have encountered unanticipated fault conditions that required extensive ground intervention to overcome and in some cases contributed to the system failure. This effort intends to develop a modified hidden Markov Model (MHMM) for spacecraft guidance, navigation, and control (GNC) although the approach is intended to be equally applicable to all spacecraft and instrument functions. JPL’s state of the art depends on up-front identification of faults for devising fault protection measures. In contrast, our proposed approach depends on modeling expected behavior and looking for variances from the expectation. Specifically, our effort will create a model of states and transitions from known GNC architectures (the observable part of the MHMM) while also including hidden states that represent uncontrollable and unobservable conditions. This effort will also develop a means for training the model (determining the state transition probabilities and the emission probabilities) along with prototype software that estimates system state and enables fault diagnosis. In addition, this work will evaluate the probability of detecting and diagnosing faults as the primary quantitative metric.

12th Annual Conference on Systems Engineering Research—“Engineered Resilient Systems: Challenges and Opportunities in the 21st Century” (March 21–22, 2014): The University of Southern California in collaboration with the Stevens Institute of Technology presents the 12th Annual Conference on Systems Engineering Research. The primary conference objective is to provide practitioners and researchers in academia, industry, and government a common platform to present, discuss, and influence systems engineering research with the intent to enhance systems engineering practice and education. Two of the co-leads of the KISS ERSS study (John Day and Mitch Ingham) are leading a session on Autonomous Resiliency Research and Applications, and anticipate participation from other study participants.

5.0 CONCLUSIONS

Exploring space is a challenging and difficult endeavor. As the destinations become more challenging and science questions more sophisticated, and as mission experience accumulates, the most accessible targets are visited and the knowledge frontier advances to more difficult, harsh, and inaccessible environments. These space missions will require more resilience in order to perform the desired science in new environments, under constraints of development and operations cost, acceptable risk, and communications delays. Development of space systems with these capabilities has the potential to revolutionize space science, enabling as yet unforeseen missions and breakthrough science observations.

Our KISS ERSS study provided an essential venue for the consideration of these challenges and goals. The study allowed a collection of diverse and engaged engineers, researchers, and scientists to think deeply about the theory, approaches, and technical issues involved in developing and applying resilience capabilities. The conclusions summarize the varied and disparate discussions that occurred during the study, and include new insights about the nature of the challenge and potential solutions:

1. **There is a clear and definitive need for more resilient space systems.** During our study period, the key scientists/engineers we engaged to understand potential future missions confirmed the scientific and risk reduction value of greater resilience in the systems used to perform these missions. Through this process, we also refined our understanding of the challenges and needs of these potential missions. Several distinct trends will influence space exploration missions in the next decade—new destinations with unknown or poorly characterized conditions and hazards, multielement missions and multimission campaigns, and long-duration flight. This leads to new challenges including: hazardous conditions that limit mission lifetime, such as high radiation levels surrounding interesting destinations like Europa or toxic atmospheres of planetary bodies like Venus; unconstrained environments with navigation hazards, such as free-floating active small bodies; multielement missions required to answer more sophisticated questions, such as MSR; and long-range missions that must survive equipment failures over the span of decades, such as Kuiper belt exploration. Further, we believe new and unprecedented missions will be enabled when we have proven resilience capabilities available for incorporation into system designs. Envision the possibilities if we could send vehicles into very hazardous environments with the same confidence afforded by present-day missions, or if mission planners and scientists could count on good science return, even when operators cannot be in the loop.
2. **Resilience can be quantified in measurable terms—project cost, mission risk, and quality of science return.** In order to consider resilience properly in the set of engineering trades performed during the design, integration, and operation of space systems, the benefits and costs of resilience need to be quantified. We believe, based on the work done during the study, that appropriate metrics to measure resilience must relate to risk, cost, and science quality/opportunity. As these become more specific and concrete, the risk and opportunity cost can be weighed in design trades. Just as current spacecraft and rovers use fault protection to reduce the risk of component failures, future vehicles will apply characteristics to increase resilience in the system design. Many possible metrics for quantifying resilience were discussed during the study, but additional work is necessary to evolve and mature these preliminary ideas. Future work in developing metrics will build upon the definition of resilience established in the study and experience of projects beginning to apply related metrics in current design trades (e.g., NASA’s Space Launch System). In the end, these metrics need to clearly convey the value proposition of resilience; once the costs and benefits can be articulated, the considerations associated with resilience will become as familiar as current trades of mass and power.
3. **There are many existing basic technologies that can be applied to engineering resilient space systems.** The many benefits of bringing together a diverse set of professionals include the wide range of knowledge that is exposed in the process. Through the discussions during the study, we

found many different approaches and multiple thrusts of research that are addressing various facets of resilience, some within NASA, and many more beyond. Examples from civil architecture, DoD/DARPA initiatives (e.g., Adaptive Vehicle Make, Engineered Resilient Systems), ‘smart’ power grid control, cyber-physical systems, software architecture, and application of formal verification methods for software broadened and enriched the discussion. The variety and scope of related efforts is encouraging and presents many opportunities for collaboration and development, and we expect many collaborative proposals and joint research as a result of the study. The problem is difficult enough that there is a need to cooperate with other researchers in the same and related domains to bootstrap efforts. However, a lack of a common conceptual foundation hinders technology integration and adoption. As an example, certain capabilities that are prerequisites for autonomous resilience exist in different fields (machine learning, data analytics, controls, fault detection and isolation, planning and artificial intelligence, etc.), but we currently lack the understanding of how to fit technology pieces together into an integrated system. An approach is needed to figure out how to effectively integrate resilience-related capabilities and technologies in coherent system architectures.

4. **Use of principled architectural approaches are key for managing complexity and integrating disparate technologies.** The main challenge inherent in considering highly resilient space systems is that the increase in capability can result in an increase in complexity, with all of the risks and costs associated with more complex systems. What is needed is a better way of conceiving space systems that enables incorporation of capabilities without increasing complexity. We believe principled architecting approaches provide the needed means to convey a unified understanding of the system to primary stakeholders, thereby controlling complexity in the conception and development of resilient systems, and enabling the integration of disparate approaches and technologies. There is a strong need for the development of a formal framework for making architectural tradeoff decisions between passive (inherent in design) and active resilience, reflexive, habitual and deliberative behavior, and appropriate allocation of function to architectural layers and system elements. We believe the successful application of principled architecture to future NASA missions will enable the development of robust and efficient missions with dramatic reductions in cost and risk. More capable spacecraft can be conceived and deployed, expanding the frontiers of exploration and autonomy. Significant research into the theory behind such a framework is needed, as are means for articulating the resulting principles and application to system designs, but this work will lead to a fundamental change in how complex space systems are developed and operated, with similar benefits for systems beyond the spacecraft domain. A representative architectural example is included in Appendix F.
5. **Developing trusted resilience capabilities will require a diverse yet strategically directed research program.** Despite the interest in, and benefits of, deploying resilience space systems, to date, there has been a notable lack of meaningful demonstrated progress in systems capable of working in hazardous uncertain situations. The roadmaps completed during the study, and documented in this report, provide the basis for a real funded plan that considers the required fundamental work and evolution of needed capabilities. The study allowed exploration of needs and capabilities, and possible technologies and solutions. This exploration has elicited key ideas, including architecture, robust control, planning and reasoning, and V&V approaches, that will be explored in the proposals and collaborations resulting from the study. We need better models and better integration of our models to accomplish our goals; with our current techniques and tools, achieving even the lowest-hanging fruits of resilient systems has been VERY labor intensive—it is clear that our methodologies have to change in order to move forward. Research investment is needed to advance resilience engineering theory and fund the needed development of capabilities, methods, and tools. Furthermore, a path for infusion of these capabilities, methods, and tools is critical; we believe challenging yet credible technology demonstrations are a key step for infusion—not just simulations and laboratory work, but also field testing and use of existing

NASA spacecraft that are beyond their original and extended mission (e.g., the Kepler spacecraft).



ENGINEERING RESILIENT
S P A C E S Y S T E M S

6.0 REFERENCES

- 5th International Workshop on Software Engineering for Resilient Systems (SERENE 2013). Retrieved November 25, 2013, from http://serene.uni.lu/Workshops/SERENE_2013/Call_For_Papers.
- Alderson, D. L., and Doyle, J. C. (2010). Contrasting Views of Complexity and Their Implications for Network-Centric Infrastructures. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, 40(4), 839–852.
- ARINC. (2013). *Avionics Application Software Standard Interface*. ARINC 653.
- Bayer, T. J., et al. (2010). An Operations Concept for Integrated Model-Centric Engineering at JPL. *2010 IEEE Aerospace Conference*, Big Sky, MT, March 6–13, 2010.
- Bayer, T. J., et al. (2011). Update – Concept of Operations for Integrated Model-Centric Engineering at JPL. *2011 IEEE Aerospace Conference*, Big Sky, MT, March 5–12, 2011.
- Benton, J., Coles, A. J., and Coles A. I. (2012). Temporal Planning with Preferences and Time-Dependent Continuous Costs. *ICAPS 2012*.
- Bernard, D. E., et al. (1998). Design of the Remote Agent Experiment for Spacecraft Autonomy. *1998 IEEE Aerospace Conference*, Aspen, CO, March 21–28, 1998.
- Blum, A. L., and Furst, M. L. (1997). Fast Planning Through Planning Graph Analysis. *Artificial Intelligence*, 90(1–2), 281–300.
- Borucki, W. J., et al. (2011). Characteristics of Planetary Candidates Observed by Kepler, II: Analysis of the First Four Months of Data. *The Astrophysical Journal*, 736(1).
- Bradski, G., and Kaehler, A. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. Sebastopol, CA: O’Reilly Media.
- Braman, J. M. B., Murray, R. M., and Ingham, M. D. (2007). Verification Procedure for Generalized Goal-Based Control Programs. *2007 AIAA Infotech@Aerospace Conference and Exhibit*, Rohnert Park, CA, May 7–10, 2007.
- Brown, O. and Eremenko, P. (2006). Fractionated Space Architectures: A Vision for Responsive Space. *Proceedings of the AIAA 4th Responsive Space Conference*, Los Angeles, CA, April 24–27, 2006. AIAA-RS4-2006-1002.
- Buckley, B. and Vangaasbeck, J. (1994). SCL: An Off-The-Shelf System for Spacecraft Control. *Space Mission Operations and Ground Data Systems*, 1, 559–568.
- Byers, C. M., Cheng, B. H., and McKinley, P. K. (2011). Digital Enzymes: Agents of Reaction Inside Robotic Controllers for the Foraging Problem. *Proceedings of the 2011 ACM Genetic and Evolutionary Computation Conference*, Dublin, Ireland, 243–250.
- Castano, R., et al. (2006). Opportunistic Rover Science: Finding and Reacting to Rocks, Clouds, and Dust Devils. *2006 IEEE Aerospace Conference*, Big Sky, MT, March 4–11, 2006.
- Cheng, Y., Johnson, A., and Matthies, L. (2005). MER-DIMES: A Planetary Landing Application of Computer Vision. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CPRV 2005)*, 1, 806–813.
- Chien, S., Knight, R., Stechert, A., Sherwood, R., and Rabideau, G. (1999). Using Iterative Repair to Increase the Responsiveness of Planning and Scheduling for Autonomous Spacecraft. *International Joint Conference on Artificial Intelligence (IJCAI 99)*, Stockholm, Sweden.

- Chien, S., et al. (2005). Using Autonomy Flight Software to Improve Science Return on Earth Observing One. *Journal of Aerospace Computing, Information, and Communication*, 2, 196–216.
- Christopher, M., and Peck, H. (2004). Building the Resilient Supply Chain. *International Journal of Logistics Management*, 15(2), 1–14.
- Coles, A. I., Fox, M., Halsey, K., Long, D., and Smith, A. (2009). Managing Concurrency in Temporal Planning Using Planner-Scheduler Interaction. *Artificial Intelligence*, 173(1), 1–44.
- Coles, A. J., Coles, A. I., Fox, M., and Long, D. (2011). LPRPG: A Planner for Metric Resources. *The 2011 International Planning Competition*, 58.
- Coles, A. J., Coles, A. I., Fox, M., and Long, D. (2012). COLIN: Planning with Continuous Linear Numeric Change. *Journal of Artificial Intelligence Research*, 44, 1–96.
- Conroy, M., Mazzone, R., and Lin, W. (2013). NASA Integrated Model-Centric Architecture (NIMA) Model Use and Re-Use. *2013 IEEE Aerospace Conference*, Big Sky, MT, March 2–9, 2013.
- DARPA Tactical Technology Office. (n.d.). META. Retrieved November 25, 2013, from [http://www.darpa.mil/Our_Work/TTO/Programs/AVM/AVM_Design_Tools_\(META\).aspx](http://www.darpa.mil/Our_Work/TTO/Programs/AVM/AVM_Design_Tools_(META).aspx).
- Doyle, J. C., and Csete, M. (2011). Architecture, Constraints, and Behavior. *Proceedings of the National Academy of Science*, 108(Supplement 3), 15624–15630.
- DSN 1st Workshop on Systems Resilience (WSR 2013). Retrieved November 25, 2013, from <http://systemsresilience.org/wsr2013/wsr2013.html>.
- Dvorak, D. (Ed.). (2009). *NASA Study on Flight Software Complexity*. Final Report submitted to NASA Office of Chief Engineer.
- Epp, C. D., Robertson, E. A., and Brady, T. (2008). Autonomous Landing and Hazard Avoidance Technology (ALHAT). *2008 IEEE Aerospace Conference*, Big Sky, MT, March 1–8, 2008.
- Esposito, L. W. (n.d.). Mission Concept: Venus In Situ Explorer (VISE). White paper to the 2013 Decadal Survey Inner Planets Panel.
- Estlin, T., et al. (2012). AEGIS Automated Targeting for the MER Opportunity Rover. *ACM Transactions on Intelligent Systems and Technology*, 3(3), 50.
- Feiler, P. H., and Gluch, D. P. (2012). *Model-Based Engineering with AADL: A Introduction to the SAE Architecture Analysis and Design Language*. Boston, MA: Addison-Wesley.
- Fesq., L., Fretz, K., and Newhouse, M. (2013). *Report on the 2012 NASA Spacecraft Fault Management Workshop*. Final Report submitted to the NASA Science Mission Directorate.
- Frank, J., Jónsson, A., Morris, R., and Smith, D. E. (2001). Planning and Scheduling for Fleets of Earth Observing Satellites. *Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics, Automation and Space*.
- Garlan, D., Cheng, S. W., Huang, A. C., Schmerl, B., and Steenkiste, P. (2004). Rainbow: Architecture-Based Self-Adaptation with Reusable Infrastructure. *IEEE Computer*, 37(10), 46–54.
- Gat, E. (1997). ESL: A Language for Supporting Robust Plan Execution in Embedded Autonomous Agents. *Proceedings of the 1997 IEEE Aerospace Conference*, 1, 319–324.
- Gehrels, N., et al. (2004). The Swift Gamma-Ray Burst Mission. *The Astrophysical Journal*, 611(2), 1005–1020.
- Gerhart, J., and Kirschner, M. (2007). The Theory of Facilitated Variation. *Proceedings of the National Academy of Sciences*, 104(Suppl. 1), 8582–8589.

- Goldberg, S. B., Maimone, M. W., and Matthies, L. (2002). Stereo Vision and Rover Navigation Software for Planetary Exploration. *Proceedings of the 2002 IEEE Aerospace Conference*, 5, 2025–2036.
- Goldsby, H. J., Cheng, B. H., McKinley, P. K., Knoester, D. B., and Ofria, C. A. (2008). Digital Evolution of Behavioral Models for Autonomic Systems. *2008 International Conference on Autonomic Computing*, 87–96.
- Grasso, C. A. (2002). The Fully Programmable Spacecraft: Procedural Sequencing for JPL Deep Space Mission Using VML (Virtual Machine Language). *Proceedings of the 2002 IEEE Aerospace Conference*, 1, 75–81.
- Harel, D. (1987). Statecharts: A Visual Formalism for Complex Systems. *Science of Computer Programming*, 8(3), 231–274.
- Hayati, S., et al. (2007). Advanced Robotics Technology Infusion to the NASA Mars Exploration Rover (MER) Project. *6th IFAC Symposium on Intelligent Autonomous Vehicles*, Toulouse, France.
- Heinecke, H., et al. (2004). AUTomotive Open System ARchitecture – An Industry-Wide Initiative to Manage the Complexity of Emerging Automotive E/E-Architectures. *Convergence*, 325–332.
- Helmert, M. (2006). The Fast Downward Planning System. *Journal of Artificial Intelligence Research*, 26(1), 191–246.
- Hofbaur, M. W., and Williams, B. C. (2004). Hybrid Estimation of Complex Systems. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, Special Issue on Diagnosis in Complex Systems: Bridging the Methodologies of the FDI and DX Communities, 34(5), 2178–2191.
- Hoffmann, J. (2001). FF: The Fast-Forward Planning System. *AI Magazine*, 22(3), 57.
- Holzmann, G. J. (1997). The Model Checker SPIN. *IEEE Transactions on Software Engineering*, 23(5), 279–295.
- Hsu, C.W., Wah, B.W., Huang, R., and Chen, Y. (2006). Handling Soft Constraints and Goals Preferences in SGPlan. *Proceedings of the ICAPS Workshop on Preferences and Soft Constraints in Planning*, June 2006.
- Ingham, M. D., Rasmussen, R. D., Bennett, M. B, and Moncada, A. C. (2005). Engineering Complex Embedded Systems with State Analysis and the Mission Data System. *Journal of Aerospace Computing, Information, and Communication*, 2, 507–536.
- Jansma, P.A. (2011). Open! Open! Open! Galileo High Gain Antenna Anomaly Workarounds. *2011 IEEE Aerospace Conference*, Big Sky, MT, March 5–12, 2011.
- Jones, R. M. (2003). Surface and Atmosphere Geochemical Explorer (SAGE) baseline design from March 2003 Team X studies. *38th Vernadsky/Brown Microsymposium on Comparative Planetology*, Moscow, Russia, October 27–29, 2003.
- Kautz, H., and Selman, B. (1999). Unifying SAT-based Graph-based Planning. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-99)*, Stockholm, Sweden.
- Kirschner, M. and Gerhart, J. (2005). *The Plausibility of Life: Resolving Darwin's Dilemma*. New Haven, CT: Yale University Press.
- Kubitschek, D. G., et al. (2007). The Challenges of Deep Impact Autonomous Navigation. *Journal of Field Robotics*, 24(4), 339–354.
- Kunz, C., et al. (2009). Toward Extraterrestrial Under-Ice Exploration: Robotic Steps in the Arctic. *Journal of Field Robotics*, 26(4), 411–429.

- Lyke, J. (2007). Space-Plug-and-Play Avionics (SPA): A Three-Year Progress Report. *AIAA Infotech@Aerospace 2007 Conference and Exhibit*, Rohnert Park, CA, May 7–10, 2007.
- Maier, M. W., and Rehtin, E. (2009), *The Art of Systems Architecting*, 3rd ed. Boca Raton, FL: CRC Press.
- Malone, M. (2009). OPERA RHBD Multi-Core. *Presentation at Military and Aerospace Programmable Logic Devices Workshop (MAPLD)*. August 31, 2009.
- Mankins, J. C. (1995). Technology Readiness Levels: A White Paper. NASA Office of Space Access and Technology.
- Mattingly, R., and May, L. (2011). Mars Sample Return as a Campaign. *2011 IEEE Aerospace Conference*, Big Sky, MT, March 5–12, 2011.
- Model Checking. (n.d.) In *Wikipedia*, Retrieved November 25, 2013, from http://en.wikipedia.org/wiki/Model_checking
- Morgan, M. J. (1991). Integrated Modular Avionics for Next-Generation Commercial Airplanes. *Proceedings of the IEEE 1991 National Aerospace and Electronic Conference (NEACON 1991)*, 1, 43–49.
- Murray, R. M., Ingham, M. D., Day, J. C., Williams, B. C., and Reder, L. J. (2012). Retrieved on December 3, 2013 from <http://www.kiss.caltech.edu/study/systems>.
- 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.
- National Aeronautics and Space Administration. (2010a). Venus Intrepid TESSERA Lander: Mission Concept Study Report to the NRC Decadal Survey Inner Planets Panel.
- National Aeronautics and Space Administration. (2010b). Mission Concept Study: Planetary Science Decadal Survey—Trojan Tour Decadal Study. NASA SDO-12348.
- National Aeronautics and Space Administration. (2010c). Mission Concept Study: Planetary Science Decadal Survey—MSR Orbiter Mission (Including Mars Returned Sample Handling).
- National Aeronautics and Space Administration. (2010d). Mission Concept Study: Planetary Science Decadal Survey—MSR Lander Mission.
- National Aeronautics and Space Administration. (2012). *Fault Management Handbook, Draft 2*. NASA-HDBK-1002.
- National Research Council. (2003). *New Frontiers in the Solar System: An Integrated Exploration Strategy*, Washington, D.C.: The National Academies Press.
- National Research Council. (2011). *Vision and Voyages for Planetary Science in the Decade 2013–2022*, Washington, D.C.: The National Academies Press.
- Neches, R. and Madni, A. M. (2012). Towards Affordably Adaptable and Effective Systems. *Systems Engineering*, 16(2), 224–234.
- Nemeth, C., Wears, R., Woods, D., Hollnagel, E., and Cook, R. (2008). Minding the Gaps: Creating Resilience in Health Care. *Advances in Patient Safety: New Directions and Alternative Approaches*, 3, 259–271.
- Object Management Group. (OMG). (2007). *Data Distribution Service for Real-time Systems, Version 1.2*. January 2007.
- Object Management Group. (OMG). (2011). *OMG Unified Modeling Language (OMG UML), Infrastructure, Version 2.4.1*. August 2011.

- Ofria, C., and Wilke, C. O. (2004). Avida: A Software Platform for Research in Computational Evolutionary Biology. *Artificial Life*, 10(2), 191–229.
- Ono, M., and Williams, B. C. (2008). Iterative Risk Allocation: A New Approach to Robust Model Predictive Control with a Joint Chance Constraint. *Proceedings of 47th IEEE Conference on Decision and Control (CDC)*, 3427–3432.
- Ozay, N., Topcu, U., Murray, R. M., and Wongpiromsarn, T. (2011). Distributed Synthesis of Control Protocols for Smart Camera Networks. *ACM/IEEE 2nd International Conference on Cyber-Physical Systems*, 45–54.
- Penberthy, J. S., and Weld, D. S. (1992). UCPOP: A Sound, Complete, Partial-Order Planner for ADL. *Third International Conference on Knowledge Representation and Reasoning (KR-92)*, 103–114.
- Quigley, M., et al. (2009). ROS: An Open-Source Robot Operating System. *ICRA Workshop on Open Source Software*, 3(2).
- Rabideau, G., Knight, R., Chien, S., Fukunaga, A., and Govindjee, A. (1999). Iterative Repair Planning for Spacecraft Operations in the ASPEN System. *International Symposium on Artificial Intelligence Robotics and Automation in Space*, Noodwijk, The Netherlands, June 1999.
- Rayman, M. D., and Varghese, P. (2001). The Deep Space 1 Extended Mission. *Acta Astronautica*, 48(5–12), 693–705.
- Rayman, M. D., Varghese, P., Lehman, D. H., and Livesay, L. L. (2000). Results from the Deep Space 1 Technology Validation Mission. *Acta Astronautica*, 47(2–9), 475–487.
- Rouquette, N. F., Neilson, T., and Chen, G. (1999). The 13th Technology of Deep Space One. *Proceedings of the 1999 IEEE Aerospace Conference*, 1, 477–487.
- Runtime Verification. (n.d.) Retrieved November 25, 2013, from <http://runtime-verification.org>.
- Saranli, U., Buehler, M. and Koditschek, D. E. (2001). RHex: A Simple and Highly Mobile Hexapod Robot. *The International Journal of Robotic Research*, 20(7), 616–631.
- Simon, H. A. (1962). The Architecture of Complexity. *Proceedings of the American Philosophical Society*, 106(6), 467–482.
- Spirit (rover). (n.d.) In *Wikipedia*, Retrieved November 25, 2013, from http://en.wikipedia.org/wiki/Spirit_rover.
- Thompson, D. R., et al. (2012). Agile Science Operations: A New Approach for Primitive Bodies Explorations. *Proceedings of the SpaceOps 2012 Conference*, Stockholm, Sweden, June 11–15, 2012.
- Volpe, R., et al. (2001). The CLARAty Architecture for Robotic Autonomy. *Proceedings of the 2001 IEEE Aerospace Conference*, 1, 121–132.
- Wang, D., and Williams, B. C. (2014). Using Timed Automata to Model Machine Coordination. Submitted to ICAPS 2014.
- Whelan, D. A., Adler, E. A., Wilson, S. B., and Roesler, G. (2000). DARPA Orbital Express Program: Effecting a Revolution in Space-Based Systems. *Small Payloads in Space*, 136, 48–56.
- Williams, B. C. and Ingham, M. D. (2002). Model-Based Programming: Controlling Embedded Systems by Reasoning About Hidden State. *8th International Conference on Principles and Practice of Constraint Programming (CP-02)*, Ithaca, NY, September 2002.
- Williams, B. C., Ingham, M. D., Chung, S. H., and Elliott, P. H. (2003). Model-Based Programming of Intelligent Embedded Systems and Robotic Space Explorers. *Proceedings of the IEEE, Special Issue on Modeling and Design of Embedded Software*, 91(1), 212–237.

- Williams, B. C., Ingham, M. D., Chung, S. H., Elliott, P. H., and Hofbaur, M. (2004). Model-based Programming of Fault-Aware Systems. *AI Magazine*, 24(4), 61–75.
- Williams, B. C., and Nayak, P. P. (1996). A Model-Based Approach to Reactive Self-Configuring Systems. *Proceedings of the National Conference on Artificial Intelligence*, 971–978.
- Williams, B. C., and Nayak, P. P. (1997). A Reactive Planner for Model-based Executive. *Proceedings of the 15th International Joint Conference on Artificial Intelligence (IJCAI 97)*, 1178–1185.
- Wongpiromsarn, T., Topcu, U., and Murray, R. M. (2010). Automatic Synthesis of Robust Embedded Control Software. *AIAA Spring Symposium on Embedded Reasoning: Intelligence in Embedded Systems*.
- Woods, M., et al. (2008). Crest Autonomous Robotic Scientist: Developing a Closed-Loop Science Exploration Capability for European Mars Missions. *i-SAIRAS: International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Hollywood, USA.
- Xu, H., Topcu, U., and Murray, R. M. (2012). A Case Study on Reactive Protocols for Aircraft Electric Power Distribution. *2012 IEEE 51st Annual Conference on Decision and Control (CDC)*, 1124–1129.

APPENDIX A: WORKSHOP PARTICIPANTS

NAME	ORGANIZATION	EMAIL
Ella M. Atkins	University of Michigan Aerospace Engineering	ematkins@umich.edu
Erwin W. Baumann	Northrop Grumman Aerospace Systems Advanced Programs & Technologies	erwin.baumann@ngc.com
Melissa K. Bunte	Arizona State University School of Earth and Space Exploration	mbunte@asu.edu
Richard Camilli	Woods Hole Oceanographic Inst. Applied Ocean Physics & Engineering	rcamilli@whoi.edu
George J. Cancro	JHU/APL Space Department	George.Cancro@jhuapl.edu
Betty HC Cheng	Michigan State University Computer Science & Engineering	chengb@cse.msu.edu
John C. Day	JPL/Caltech Autonomy and Fault Protection	john.c.day@jpl.nasa.gov
Kenneth M. Donahue	JPL - System Architecture and Behavior	kdonahue@jpl.nasa.gov
John C. Doyle	Caltech Control and Dynamical Systems	doyle@caltech.edu
Tara A. Estlin	JPL Automation Group	tara.estlin@jpl.nasa.gov
Lorraine Fesq	JPL Engineering Development Office	lorraine.m.fesq@nasa.gov
Kai Goebel	NASA Intelligent Systems	kai.goebel@nasa.gov
Kim P. Gostelow	JPL Flight Computer and Software Technology	kim.p.gostelow@jpl.nasa.gov
Gerard J. Holzmann	JPL Laboratory for Reliable Software	gholzmann@acm.org
Andrew P. Ingersoll	Caltech Geological and Planetary Sciences	api@gps.caltech.edu
Michel D. Ingham	JPL Systems and Software Division	michel.d.ingham@jpl.nasa.gov
Stephen B. Johnson	University of Colorado, Colorado Springs Center for Space Studies	sjohns22@uccs.edu
Joseph A. Kochocki	Charles Stark Draper Laboratory Avionics Architectures	jkochocki@draper.com
Azad M. Madni	University of Southern California Industrial and Systems Engineering/SAE	azad.madni@usc.edu
Richard M. Murray	Caltech Control and Dynamical Systems	murray@cds.caltech.edu
Necmiye Ozay	Caltech Computing & Mathematical Sciences	necmiye@caltech.edu
Robert D. Rasmussen	JPL - Systems and Software Division	robert.d.rasmussen@jpl.nasa.gov
Leonard J. Reder	JPL Flight Software Architecture & Applications	redr@jpl.nasa.gov
Abhinav Saxena	NASA Ames Research Center/SGT Inc Prognostics Center of Excellence, Intelligent Systems Division	abhinav.saxena@nasa.gov
Howard E Shrobe	DARPA	howard.shrobe@darpa.mil
Janos Sztipanovits	Vanderbilt University Institute for Software Integrated Systems (ISIS)	janos.sztipanovits@vanderbilt.edu
Leslie K. Tamppari	JPL	leslie.tamppari@jpl.nasa.gov
David R. Thompson	JPL Machine Learning/Instrument Autonomy	david.r.thompson@jpl.nasa.gov
Eric M. Timmons	MIT Aeronautics and Astronautics	etimmons@mit.edu
David C. Wang	MIT Aeronautics/Astronautics	davidcw@mit.edu
Brian C. Williams	MIT Aeronautics/Astronautics	williams@mit.edu
Huan Xu	Caltech Mechanical Engineering	mumu@caltech.edu

APPENDIX B: WORKSHOP AGENDAS

Workshop #1

 		
Engineering Resilient Space Systems July 30 - August 3, 2012 Overview Schedule		
Monday, July 30, 2012 - Hameetman Auditorium - Cahill Building		
Time	Short Course - Open to All Interested Parties	Speaker
8:00 - 8:30	Coffee and refreshments	
8:30 - 8:45	Introduction to Short Courses - What is a resilient system?	Team Leads; Short Courses Moderated By: Len Reder
8:45 - 10:00	Principled System Architecture (includes 15 minutes for Q+A)	Robert Rasmussen
10:00-10:30	Break (Coffee, Discussion)	
10:30-11:45	Capturing Flight Software Architecture using DSLs (includes 15 minute for Q+A)	Kim Gostelow
11:45 - 12:45	On site, informal lunch provided by KISS for all short course attendees	
12:45 - 2:00	Control Theory and Methods (includes 15 minutes for Q+A)	Richard Murray
2:00 - 2:30	Break (Coffee, Discussion)	
2:30-3:45	Autonomy Practices (includes 15 minutes for Q+A)	Brian Williams
3:45-5:00	Ultra-Reliability for Interstellar Missions (includes 15 minutes for Q+A)	Henry Garrett
5:00	SHORT COURSE CONCLUDES	

Monday, July 30, 2012 - Third Floor - Keith Spalding Building		
Time	Invitation-Only Workshop	Speaker
5:00 - 5:15	Invitation-only workshop participants walk to Keck Institute and check in at Keith Spalding Bldg. 3rd Floor Rm 376 and enjoy refreshments	
5:15 - 5:45	Introduction to the Institute and to KISS	Michele Judd
5:45 - 6:15	Participant Introductions	
6:15 - 6:30	Walk to Athenaeum	
6:30 - 8:30	KISS Dinner on the Athenaeum Lawn	



Engineering Resilient Space Systems
July 30 - August 3, 2012
Overview Schedule



Tuesday, July 31, 2012 - Keith Spalding Building - Third Floor
Theme: Vision For Future Missions (Requirements & Ref. Mission Ideas)

Time	Workshop	Speaker
8:00 - 8:30	Coffee and refreshments	
8:30 - 9:00	Goals & Products for Workshop	Mitch Ingham
9:00 - 9:30	Provocative Talk: Planetary Decadal Survey technology needs and their scientific rationale	Andy Ingersoll
9:30 - 10:00	GROUP DISCUSSION: The Science Need a) What are the driving science needs? b) How far will current paradigm take us?	Moderated by: Mitch Ingham
10:00 - 10:30	Break	
10:30 - 11:30	Provocative Talk: Vision For Future Missions (Engineering Resilient Systems)	Gentry Lee
11:30 - 12:00	GROUP DISCUSSION: The Vision For Future Missions a) What are useful definitions of resilience? b) What are key provocative questions? c) What is relationship to dependability and robustness?	Moderated by: Mitch Ingham
12:00 - 1:30	KISS Lunch at the Athenaeum	
1:30 - 2:30	Provocative Talk: Agile Science Operations (includes 15 minutes for Q+A)	David Thompson
2:30 - 3:00	GROUP DISCUSSION: The Vision For Future Missions (Continued) a) What are the reference missions? b) What are key mission requirements?	Moderated by: Mitch Ingham
3:00 - 3:30	Break	
3:30 - 3:45	Context Talk: Science Missions	Leslie Tamppari
3:45 - 5:00	Group discussion: The Vision For Future Missions (Continued) a) Revisit science needs? b) Refine list of reference missions?	Moderated by: Mitch Ingham
5:00 - 5:30	Wrap up list of reference missions & provocative questions	Moderated by: Mitch Ingham
6:00 - 9:00	No-Host Dinner in Pasadena (KISS to pay for all postdocs and graduate students)	



Engineering Resilient Space Systems
July 30 - August 3, 2012
Overview Schedule



Wednesday, August 1, 2012 - Keith Spalding Building - Third Floor
Theme: Adaptability (concepts & capabilities)

Time	Workshop	Speaker
8:00 - 8:30	Coffee and refreshments	
8:30 - 9:00	Goals & Products for Workshop	Richard Murray
9:00 - 9:45	Provocative Talk: Architecture, Constraints, and Behavior	John Doyle
9:45 - 10:45	GROUP DISCUSSION: Resilience and Adaptability a) What are attributes of a resilient system? Of an adaptable system? b) How is resilience similar to, and different from, adaptability? c) What can we learn from neuroscience and biological systems? d) How can resilience or adaptability be measured? How much is enough?	Moderated by: Richard Murray
10:45 - 11:15	Break	
11:15 - 12:00	GROUP DISCUSSION: Architectural Quality Attributes For Adaptability a) How to evaluate suitability for adaptable implementations? b) What are the architectural attributes? c) Role of Hardware, Software & Systems?	Moderated by: Richard Murray
12:00 - 1:30	KISS Lunch at the Athenaeum	
1:30 - 2:15	Modeling for Structural Adaptation: Lessons Learned from Model-based Design	Janos Sztipanovits
2:15 - 3:00	GROUP DISCUSSION: Adaptability Concepts and Challenges a) Operational Capabilities (role of flight vs. ground; state awareness) b) Dealing with uncertainty and self-preservation c) Revisit reference mission scenarios	Moderated by: Richard Murray
3:00 - 3:30	Break	
3:30- 3:45	Context Talk: Autonomy Validation	Richard Doyle
3:45 - 4:00	GROUP DISCUSSION: Autonomy Validation	Moderated by: Richard Murray
4:00 - 5:30	Post-docs and JPL ECH short talks (Nominally five 10 to 15 min. talks) + workshop day wrap up.	Moderated by: Richard Murray
6:00 - 8:00	No-Host Dinner in Pasadena (KISS to pay for all postdocs and graduate students)	



Engineering Resilient Space Systems
July 30 - August 3, 2012
Overview Schedule



Thursday, Aug 2, 2012 - Keith Spalding Building - Third Floor
Theme: Trust (Principles, Risks, Patterns, Verification)

Time	Workshop	Speaker
8:00 - 8:30	Coffee and refreshments	
8:30 - 9:00	Goals & Products for Workshop	Brian Williams
9:00 - 9:45	Affordable, Adaptable and Effective: The Case for Engineered Resilient Systems	Azad Madni
9:45 - 10:30	GROUP DISCUSSION: Resilient Systems, possible topics include: a) What are the road blocks to implementation? b) What are road blocks to acceptance/trust in behavior? c) Architecting resilient systems	Moderated by: Brian Williams
10:30 - 11:00	Break	
11:00 - 11:30	GROUP DISCUSSION: Quality Architectural Attributes For Trust a) What is resilience in software? Is self-adaptability a requirement? b) How can we come to trust a self-adaptive system? c) How is the amount of complexity related to trustability? d) How can trust be measured? What level of trust is required?	Moderated by: Brian Williams
11:30 - 1:00	KISS Lunch at the Athenaeum	
1:00 - 1:15	Context Talk: Agile Verification	Gerard Holzmann
1:15 - 2:00	GROUP DISCUSSION: Development-time approaches a) Formal methods for verification and validation b) Tradeoff between flight and ground capabilities in dynamic environments driven by validation considerations	Moderated by: Brian Williams Gerard Holzmann
2:00 - 2:15	Context Talk: ISHM (integrated system health management)	Erv Baumann
2:15 - 3:00	GROUP DISCUSSION: Run-time approaches a) How do you define health in a resilient system? b) How do you determine system health in a resilient system? c) What are the theoretical limits of diagnosis and prognosis?	Moderated by: Brian Williams
3:00 - 3:30	Break	
3:30 - 5:30	Wrapping up workshop with lightning talks given by all participants	All
6:00 - 8:00	Dinner at The Athenaeum (spouses/guest invited)	



Engineering Resilient Space Systems
July 30 - August 3, 2012
Overview Schedule



Friday, August 3, 2012 - Keith Spalding Building - Third Floor
Theme: Planning for Workshop Two! (Optional JPL Tour)

Time	Workshop	Speaker
8:00 - 8:30	Coffee and refreshments	
8:30 - 9:00	Summing up workshop (summary from each day, by moderator).	Team Leads
9:00 - 10:30	GROUP DISCUSSION: Synthesis of discussions: a) What are important themes and ideas from lightning talks? b) Review and assess provocative questions? c) What are the interesting ideas that deserve further exploration?	Moderated by John Day
10:30 - 11:00	Break	
11:00 - 11:45	Develop plan for study period and 2nd workshop: a) Discuss topics, goals and products b) Who would be interested to champion a theme or idea? c) Logistics of telecons between workshops d) What worked well in the workshop and what went wrong?	Team Leads to moderate
11:45 - 12:00	Logistics of checking out, and workshop slideshow	Michele Judd
12:00	Workshop concludes (Meet for pay your own way lunch and optional JPL tour)	
1:30	JPL Tour (must start promptly at 1:30)	Mitch Ingham

Workshop #2

 	
Engineering Resilient Space Systems II February 26 - 28, 2013 Overview Schedule	
Tuesday, February 26, 2013 - Keith Spalding Building - 3rd Floor - Room 376	
Time	Workshop
8:00 - 8:30	Coffee and refreshments
8:30 - 9:00	Goals & Products for Workshop
9:00 - 9:45	Focus group outbrief - Reference Missions (30 minute brief, 15 minute discussion)
9:45 - 10:45	Focus group outbrief - Capability Survey (45 minute brief, 15 minute discussion)
10:45 - 11:15	Break (Coffee & Discussion)
11:15 - 12:00	Focus group outbrief - Architecture (30 minute brief, 15 minute discussion)
12:00 - 1:30	Buffet lunch at Athenaeum provided by KISS
1:30 - 2:00	Setup reference mission breakout activity
2:00 - 3:00	Initial Breakout Session
3:00 - 3:30	Break (Coffee & Discussion)
3:30 - 4:00	Unstructured individual/small group discussions
4:00 - 5:30	Continued breakout sessions
5:30	Wrap up for the day
5:45 - 6:00	Walk to the Athenaeum
6:00 - 8:00	Dinner provided by KISS at the Athenaeum
Wednesday, February 27, 2013 - Keith Spalding Building - 3rd Floor - Room 376	
Time	Workshop
8:00 - 8:30	Coffee and refreshments
8:30 - 9:00	Goal setting, check on progress/questions
9:00 - 10:45	Continued breakout sessions
10:45 - 11:15	Break (Coffee & Discussion)
11:15 - 12:00	Unstructured individual/small group discussions
12:00 - 1:30	Buffet lunch at Athenaeum provided by KISS
1:30 - 2:00	Breakout group #1 report
2:00 - 2:30	Breakout group #2 report
2:30 - 3:00	Breakout group #3 report
3:00 - 3:30	Break (Coffee & Discussion)
3:30 - 4:00	Opportunistic individual/small group discussions
4:00 - 5:30	Split into focus groups, perform synthesis on input received to date
5:30	Wrap up for the day
5:45 - 6:00	Walk to dinner
6:00 - 8:00	Dinner provided by KISS at the Athenaeum

APPENDIX C: FOCUS GROUP MEMBERSHIP

Architecture Focus Group Membership (in alphabetical order):

Kenneth Donahue, JPL

John Doyle, Caltech

Tara Estlin, JPL

Kim Gostelow, JPL

Mitch Ingham, JPL

Joseph Kochocki, C.S. Draper Lab

Robert Rasmussen, JPL

Janos Sztipanovits, Vanderbilt U.

David Wang, MIT

Huan Xu, Caltech

In addition to these core members, participants Richard Camilli (WHOI) and Howie Shrobe (MIT/DARPA) joined the discussions at the second workshop.

Reference Mission Focus Group Membership (in alphabetical order):

George Cancro, JHU APL

John Day, JPL

Tara Estlin, JPL

Lorraine Fesq, JPL

Necmiye Ozay, Caltech

Leslie Tamppari, JPL

David Thompson, JPL

Brian Williams, MIT

Capabilities Focus Group Membership (in alphabetical order):

Ella Atkins, University of Michigan

Erv Baumann, NGAS

Betty Cheng, Michigan State

Azad Madni, USC

Richard Murray, Caltech

Len Reder, JPL

Abhinav Saxena, Ames

Eric Timmons, MIT

APPENDIX D: DEFINITIONS OF RESILIENCE

This appendix includes many of the definitions of resilience referenced, proposed, or discussed during the study.

Ability of a system to adjust its functioning prior to, during, or following changes and disturbances, so that it can sustain required operations, even after a major mishap or in the presence of continuous stress (Nemeth, Wears, Woods, Hollnagel, and Cook, 2008)

Ability of a system to adapt affordably and perform effectively across a wide range of operational contexts, where context is defined by mission, environment, threat, and force disposition (Neches and Madni, 2012)

Ability of a system to return to its original state or move to a new, more desirable state after being disturbed (Christopher and Peck, 2004)

Ability of a of a system to achieve envisioned (science) objectives even if the system (spacecraft) performance and/or environment are not as expected (Murray, Ingham, Day, Williams, and Reder, 2012)

Ability of a system to offer broad utility in a wide range of operations across many potential alternative futures despite experiencing disruptions (A. Madni)

Resilience: a [property] of a [system] is resilient if it can [recover] with respect to a set of [perturbations]. (David Wang, using a variation on David L Alderson and John C Doyle (2010) definition of robustness. Where system—a grouping of components, i.e. the entire space mission, just the spacecraft, the communications subsystem; property—a function or objective of a system; perturbations—disturbances/failures/noise)

Resilience is defined as “establishing the goals of anticipate, withstand, recover, and evolve.” (Aerospace, via Fesq)

“A resilient system is a system that can, in the face of unknown, large-scale events, recover from the failures and maintain its functions.” (DSN 1st Workshop on System Resilience, 2013)

A resilient system can figure out what to do in the presence of anomalies and discoveries (Gentry Lee)

Resilience is adaptation when assumptions do not pan out. (D. Thompson)

Resilience: goal-directed behavior in an uncertain environment. (R. Rasmussen)

Resilient systems provide a high likelihood of mission success and a high rate of return, despite environmental conditions and mission goals that are highly uncertain and that change dynamically. (Brian Williams)

Resilience: “an ability of the system to persistently deliver its services in a trustworthy way even when facing changes, unforeseen failures and intrusions” (5th Int. Workshop on Software Engineering for Resilient Systems SERENE 2013, 2013)

APPENDIX E: SUMMARY OF REFERENCE MISSIONS CONSIDERED

The Reference Mission Focus Group desired to gain insight into the specific resilience needs of potential future missions. We interviewed scientists and/or engineers (see *Scientists/Engineers Interviewed* below) who are working closely with each of the mission concepts that we chose out of the Decadal Survey. In preparation for the interview, we developed a set of potential questions designed to elicit the resilience needs of the mission (see *Interview/Questions* below).

After the interviews, we downselected to four reference missions to be further examined and discussed during Workshop #2. The four reference missions chosen were: Venus Lander, MSR, TTRV, and the Interstellar Probe. The first three represent varied needs. The Venus Lander highlights the importance of rapid and onboard mission replanning, onboard assessment and prioritized downlink of data, graceful degradation on components, and the ability to continue to achieve science goals during component or system degradation and failure. The MSR mission represents rapid mission planning and replanning, and highlights the resilience needs of a set of spacecraft that all must work to achieve the end goal. The TTRV mission highlights the challenges associated with long duration cruise, rapid science analysis and observation replanning, and the need to continue to take science observations even if components or systems are suboptimal. The last mission, Interstellar Probe, represents a mission that is far in the future, and encompassing all of the challenges represented by the former three missions, such that it was a useful test case for understanding if the technology capabilities and architecture concepts discussed would be applicable to future missions.

Scientists/Engineers Interviewed

Mission	Person(s) Interviewed	Date
Venus Lander/MRO	Dr. Sue Smrekar	11/20/12
Titan Balloon	Dr. Christophe Sotin	11/27/12
MSL	Dr. Joy Crisp	11/27/12
Venus Balloon	Dr. Kevin Baines	12/4/12
Europa Mission	Dr. Dave Senske and Brian Cooke	12/18/12
Trojan Asteroid Rendezvous	Dr. Julie Castillo-Rogez	12/19/12
Mars Sample Return	Erik Nilsen	2/13/13
Interstellar Probe	Gentry Lee	1/2/13
Comet Surface Sample Return	Dr. Carol Raymond	1/27/13

Interview Questions

General:

1. What capability would reduce the risk to your mission (science)?
2. What capability would reduce the cost of your mission (science)?
3. What would your concerns be with a system that has more autonomous capability (or X capability)?
4. What do you think are the largest obstacles in performing your mission and getting science return?
5. “If I only had ‘X’ everything would be better.” What is “X?”
6. What would be enabled in your mission if you had X capability? How would your mission change if you had this?

Specific:

1. What would be the value to your mission (or science return or health and safety) if you had *a system that could react to time-varying phenomena autonomously*?
2. What would be the value to your mission (or science return or health and safety) if *you could recognize and avoid environmental hazards*?

3. What would be the value to your mission (or science return or health and safety) if *you had next-generation onboard processing*? And what specifically would you like to see done with that?
4. What would be the value to your mission (or science return or health and safety) if *you had highly efficient operations*? And what specifically does that mean to you?
5. In your opinion, what kinds of decisions should be kept in the realm of the scientists and what kinds should be in the realm of the robotic system?

Final Questions:

1. Of the things we've talked about, which technology would you like to see pursued?
2. Is there anything else you think we should be considering or anything else you'd like us to know about your mission?

Needed Capabilities for Reference Missions

Capability scoring Strong/enabling Significant Weaker

ID	Name	Mission Type	Interview	Workshop Use	Needed Capabilities			
					Rapid Mission Planning	Data Processing	Graceful Degradation	Low Cost Cruise
1	Venus Lander (e.g., INTREPID, SAGE)	Harsh environment lander	S. Smrekar	Workshop exercise	X	X	X	X
2	Venus Aerial Mission	In-situ aerial explorer	K. Baines	Context	X	X	X	X
3	Titan Aerial Explorer	In-situ aerial explorer	C. Sotin	Context	X	X	X	X
4	Mars Sample Return (set of missions)	Landed rover	J. Crisp (MSL)	Workshop exercise	X	X	X	X
5	Orbiter "style" (MRO, Cassini)	Multi-tasking orbiter	S. Smrekar	Context	X		X	X
6	Trojan Tour and Rendezvous	Small body tour	J. Castillo-Rogez	Workshop exercise	X	X	X	X
7	Europa Clipper	Repeated High-value flyby	B. Cooke, D. Senke	Context			X	X
8	Interstellar Probe	Distant, long-duration explorer	G. Lee	Context		X		X
9	Interstellar Rendezvous	Distant, long-duration explorer	G. Lee	Technology evolution check	X	X	X	X
10	Comet Surface Sample Return	Sample Return	n/a	Context	X		X	X

Selected Mission Summaries

Venus Lander Reference Mission Summary



Reference Mission: Venus Lander (SAGE as example)

- Goal: Study the history of the Venusian atmosphere, climate, and surface to compare Venus to Earth and to extrasolar planets

Mission Concept:

- Launch on flyby trajectory
- Separate lander; descend to surface (~1 hour)
- Surface mission (~3 hours)
 - Land, deploy instrument
 - Survey site
 - Take and analyze samples (including drilling)
- Relay data to flyby carrier



References: http://lasp.colorado.edu/sage/archives/PDF/SAGE%20Fact%20Sheet-p6_for%20print_April-21-2010.pdf
And <http://sagemission.jpl.nasa.gov/>

Page 1



Reference Mission: Venus Lander

Key measurements

1. **Ultraviolet and near-infrared imaging** for entry context and cloud dynamics
2. **Entry temperature, pressure, dynamics, and wind speed**
3. **Atmospheric Composition** via Tunable gas and Neutral mass spectrometers
4. **Surface Geology and Weathering** via descent, panoramic, and microscopic imagers
5. **Surface composition and mineralogy** via Neutron-Activated Gamma-Ray Spectrometer and Raman and Laser-Induced Breakdown Spectroscopy

Capability needs

- **Rapid Mission Planning** - Surface mission lifetime is only 3 hours. No ground in the loop. Must get adequate compositional data. Would like to spend precious resources on the "right" rock/soil. Smart selection of samples.
- **Increase value of returned science** – (1) Identify most interesting descent images to reduce data return and possibly help select sample locations. (2) Select between geochemical measurements to prioritize/send back most interesting ones. (3) Rapidly assess drilling efficacy and select another site if needed.
- **Increase measurement frequency** – (1) Increase number of distinct atmospheric measurements taken during descent, by rapidly analyzing them. (2) Quickly expose subsurface for measurements.
- **Identify landing event rapidly** – To enable earlier start of science
- **Landing** - Hazard avoidance. Identification of contact/land at contact/land in particular .



Page 2



Mission requirements and metrics: Venus lander, e.g. INTREPID and SAGE concepts

Unconstrained physical environment

Avoid landing on or specifically land on Tessera/ Hazard Avoidance

- % chance of mission failure
- Landing on terrain of interest

Recognize, recover from landing on slope

- % chance of mission failure

Select and analyze the right samples

- Number of unweathered and distinct samples

Spacecraft cost and complexity

Reduce mass of entry system (large factor in total mission cost)

- % mass savings

Quickly expose the subsurface, somehow

- Depth of fresh material exposed
- Probability of exposing fresh material
- Time to completion, avoiding mission failure due to spacecraft loss

Unexpected science phenomena

Downlink best descent imagery first (e.g. favor diversity, avoid haze)

- Number of terrain features appearing, 800Mbits nominal downlink

Recognize bad data to re-acquire or deprioritize during downlink

- % good data in 800Mbits nominal downlink

Acquire data from distinct targets

- Spectral and chemical diversity of surface and atmospheric downlinked data

Operations efficiency and cost

Rapidly assess drilling efficacy and select another drill site if needed

- Number of unweathered and distinct samples

Rapidly identify landing event

- Amount of unique science returned



Page 3



Mission requirements and capabilities: Venus lander, e.g. INTREPID and SAGE concepts

Unconstrained physical environment

Avoid landing on or specifically land on Tessera/Hazard Avoidance

- Rapid mission planning – change landing target
- Data processing – evaluate surface data acquired
- Fail opportunistically – continue if landing sub-optimal

Select and analyze the right samples

- Rapid mission planning – take different samples
- Data processing – evaluate quality of sample site/sample

Spacecraft cost and complexity

Reduce mass of entry system (large factor in total mission cost)

- No link to capabilities matrix

Quickly expose the subsurface, somehow

- Fail opportunistically – choose another location if drilling not working
- Rapid mission planning – replan strategy for sample acquisition

Unexpected science phenomena

Downlink best descent imagery first (e.g. favor diversity, avoid haze)

- Data processing – smart selection

Recognize bad data to re-acquire or deprioritize during downlink

- Data processing – smart selection
- Rapid mission planning – re-acquire data

Acquire data from distinct targets

- Data processing – on-board analysis
- Rapid mission planning - change sample strategy to get unique samples/data

Operations efficiency and cost

Rapidly assess drilling efficacy and select another drill site if needed

- Fail opportunistically – choose another drill site
- Data processing – assess drill efficacy
- Rapid mission planning – replan drill/sampling strategy

Rapidly identify landing event

- Data processing
- Fail opportunistically – if landing sub-optimal, rapidly identify and continue on



Page 4

Mars Sample Return Reference Mission Summary



Reference Mission: Mars Sample Return (sample collection through Mars ascent)

Mars Sample Return goals:

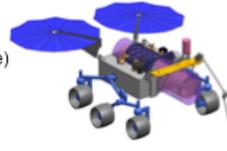
- Determine whether life ever arose on Mars, assess past and present habitability
- Collect and return set of carefully selected samples from geologically diverse sites (especially those that had aqueous processes)

At least three separate missions:

- 1) Acquire and cache samples, 2) Launch samples from Mars, 3) Return samples to Earth
- This reference mission overview focuses on elements of first two

Mission Concept for 2020 Mission:

- Launch MSL-class long-range mobility system (8-11 month cruise)
- Use MSL EDL approach (incl skycrane)
- Surface Mission (one Martian year)
 - Collect 25-35 samples (>10g, likely from diverse areas)
 - Drive 5-20km
 - Cache samples (in 1-2 caches)



Mission Concept for Fetch Rover/MAV (>2025):

- Two concepts:
 - Fetch rover + landed MAV (8 month cache retrieval, up to 14km)
 - Fetch rover + mobile MAV
- Critical steps include potential traverse, retrieving cache, transfer to MAV and leaving Mars surface



References: <http://mars.jpl.nasa.gov/m2020/mission/overview/>,
http://solarsystem.nasa.gov/multimedia/downloads/Vision_and_Voyages-FINAL1.pdf (planetary decadal study),
 Mars 2018 MAX-C Caching Rover Planetary Science Decadal Survey – Mission Concept Study,
 MSR Campaign Studies Overview (April 2012 presentation to Doug McCuiston)



MSR Resilience Needs (Sample collection rover, Fetch rover, MAV)

- Manage telecommunications uncertainty (integrity of relay/DTE path)
 - Metrics: data volume returned, # of science targets captured
 - Timeframe of need: Near-term (5-10 years)
- Automated assessment of data quality and/or science value plus automated response for handling data (e.g., reacquire, prioritize, delete, store, compress, downlink)
 - Metrics: # of science targets captured, accuracy/precision of target identification
 - Timeframe of need: Far-term (> 10 years), nice-to-have for near-term
- Simplify/prioritize telemetry to reduce operator cognitive load and streamline downlink process (ops efficiency and cost)
 - Metrics: operations cost, speed of downlink for critical data
 - Timeframe of need: Far-term (> 10 years), nice-to-have for near-term
- Handle terrain uncertainty during driving
 - Metrics: distance driven, duration of vehicle stops due to hazards
 - Timeframe of need: Near-term (5-10 years)
- Manage vehicle hardware wear and tear
 - Metrics: % of time hardware in use, labor effort to resolve problem
 - Timeframe of need: Far-term (> 10 years), nice-to-have for near-term





MSR Candidate Solutions - Technology

- **Data processing**
 - Feature detection (e.g., layering, texture, edges/shape, albedo).
 - *Smart compression, compressed sensing/sampling*
 - *Architecture support for modular algorithms (e.g., new detectors)*
 - Architecture support for sandboxing (containment) of new software (e.g., ARINC 653)
 - Machine learning to detect anomalies, novelty, patterns
 - Hazard detection, terrain assessment, mapping
 - Multi-core processing and parallelization of algorithms (and enhanced memory)
 - In-situ validation of learned capabilities
 - Signal processing / noise reduction (?)
 - Sensor fusion (?)
- **Rapid mission planning, decision making, execution**
 - Multi-core processing for algorithm speed-up
 - *V&V of complex algorithms. Also enhanced behavioral modeling and integration with V&V process.*
 - Architecture that integrates planning and control layers
 - *State estimation (both system and environment) of traditionally unobservable states. Address performance/scalability and modeling costs.*
 - Search optimization (for faster navigation, reconfiguration for degrading components, etc.)
 - Ground-based planning system that coordinates communication assets (more sophisticated MAROS)
- **Graceful degradation**
 - State estimation (including diagnosis). Address performance/scalability and modeling costs.
 - Architecture support for evolving plant model (to address hardware degradation – e.g., joint failure)
 - Rapid V&V to support the changing of modular software components. Also enhanced behavioral modeling and integration with V&V process.
 - *Design for observability and health management. Integration of diagnostic software with system design, architecture, and operations.*
 - Quick diagnosis (observability).
 - Active probing to enhance observability
 - Continuously assess/project health (prognosis)

High Priority



Page 4



MSR Candidate Solutions - Capabilities

- **Data processing**
 - Telecomm: prioritize data for downlink in case of degraded relay capability, preprocess data to reduce overall downlink
 - Terrain navigation: fuse sensor information to make quick assessment of terrain navigability
 - Science understanding: Assess data quality and/or science value to reacquire, prioritize delete, store, compress, or downlink
 - Ops cost/efficiency: Data processing to recognize/prioritize key data elements (onboard and ground capabilities)
- **Rapid mission planning**
 - Terrain navigation: make a quick decision on the right response to navigate terrain, avoid hazards and drive long distances
 - Telecomm: Manage orbiter modes, coordinate communication assets
 - Wear and tear: Determine potential solutions (ground or onboard)
- **Graceful degradation**
 - Wear and tear: quickly diagnose (i.e., observability), determine strategies to address, perform s/w upgrade, etc.
 - Continuously assess/monitor health (prognosis).
 - Telecomm: gracefully degrade comm capabilities (e.g., no single point of failure)



Page 3



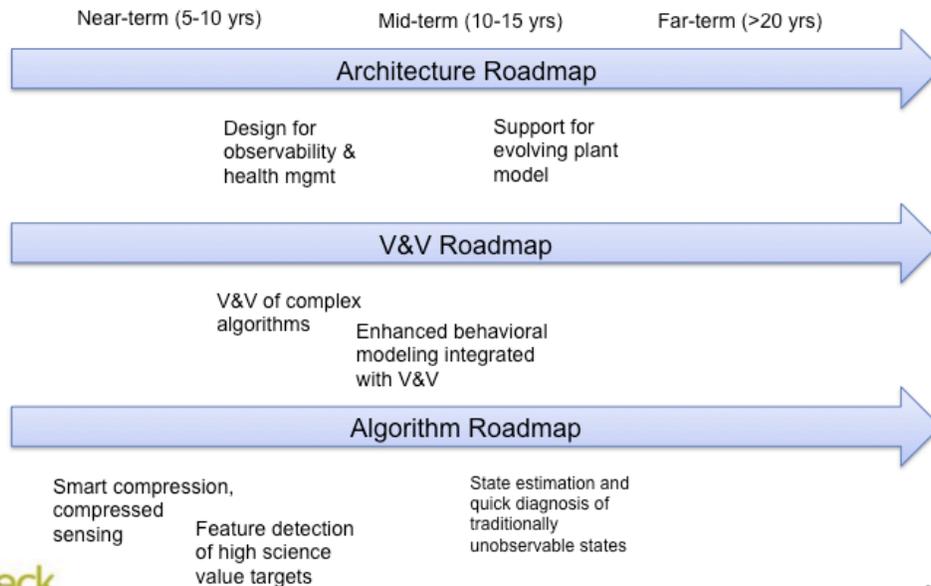
MSR Solution Roadmap

- **Architecture Roadmap**
 - Architecture support for sandboxing (containment) of new software (e.g., ARINC 653)
 - Architecture support for modular algorithms (e.g., new detectors)
 - Architecture that integrates planning and control layers
 - Design for observability and health management. Integration of diagnostic software with system design, architecture, and operations.
 - Architecture support for evolving plant model (to address hardware degradation – e.g., joint failure)
- **V&V Roadmap**
 - V&V of complex algorithms. Also enhanced behavioral modeling and integration with V&V process.
 - Rapid V&V to support the changing of modular software components. Also enhanced behavioral modeling and integration with V&V process.
 - In-situ validation of learned capabilities
- **Algorithm Roadmap**
 - Feature detection (e.g., layering, texture, edges/shape, albedo).
 - Smart compression, compressed sensing/sampling
 - Machine learning to detect anomalies, novelty, patterns
 - Hazard detection, terrain assessment, mapping
 - Multi-core processing and parallelization of algorithms (and enhanced memory)
 - Multi-core processing for planning algorithm speed-up
 - State estimation and quick diagnosis (both system and environment) of traditionally unobservable states. Address performance/scalability and modeling costs.
 - Search optimization (for faster navigation, reconfiguration for degrading components, etc.)
 - Ground-based planning system that coordinates communication assets (more sophisticated MAROS)
 - Active probing to enhance observability
 - Continuously assess/project health (prognosis)
 - Signal processing / noise reduction (?)
 - Sensor fusion (?)

Already targeted for investment
High priority for this study



MSR Solution Roadmap





Mission requirements and Resiliency Need Categories: Mars Sample Return

System State and Environment

Land within landing ellipse
 Graceful degradation – continue landing even if suboptimal
 Rapid mission planning – if landing outside ellipse need more time

Avoiding landing on hazards
 Data processing – evaluate imagery for hazards
 Graceful degradation – continue landing even if hazards

2 Deal with terrain uncertainty during driving (metric: distance, # times stuck)
 Data processing – fuse sensor information to make quick assessment
 Rapid mission planning – make a quick decision on the right response

1a Telecommunications uncertainty (metric: data bandwidth)
 Graceful degradation – single point of failure from orbiter [arch]
 Rapid mission planning – orbiter manage its safe modes, etc.
 Data processing – prioritize for downlink in case of degraded relay capability ,
 preprocess data to reduce overall downlink

Dealing with dust storms
Drilling/coring failures and uncertainty
Collect required number of samples
 Rapid mission planning and data processing, could enable faster operations and more data on ground to speed sample selection

Solar panels
Leveling for launch / attitude constraints

3 Wear and tear – dealing with hardware failures and degradation. (metric: how long h/w down, labor effort to resolve problem, science return)
 Graceful degradation – quickly diagnose (observability), determine strategies to address, perform s/w upgrade, etc. Also continuously assess/monitor health (prognosis). [arch part of solution]
 Rapid mission planning – determine potential solutions (ground or onboard)

Onboard target selection for remote and close-contact imaging and spectroscopy
 Data processing to help quickly select targets
 Rapid mission planning to collect targeted data whenever possible

1b Assess data quality and/or science value to reacquire, prioritize delete, store, compress, or downlink (close the loop). (metric: data quality – amount of data containing key science target, data accuracy)
 Data processing – smart selection

Don't miss dinosaur bone

Spacecraft cost and complexity

Manage software and operations complexity
 Rapid mission planning and data processing enable more time for operators to spend on high complexity tasks
 Software architecture methods that help manage and reduce systems complexity (both onboard software and ground).
 - easier uploads, testing, etc.
 Streamlined V&V process

Operations efficiency and cost

1c Simplify/prioritize telemetry to reduce operator cognitive load and streamline downlink
 Data processing to recognize/prioritize key data elements (onboard and ground)

Simplify sequencing and validation for ops efficiency
 Same needs as "Spacecraft cost and complexity" category

Detecting science phenomena and features

Page 7

Trojan Tour and Rendezvous Reference Mission Summary

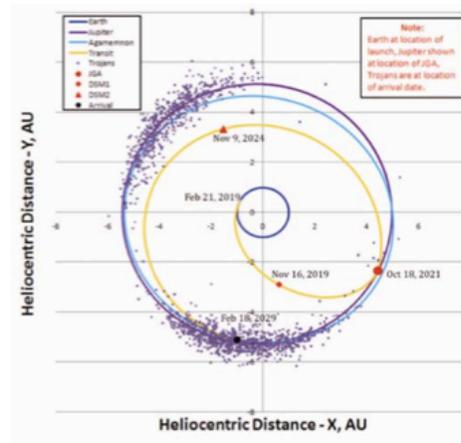
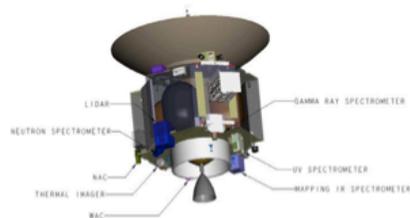


Reference Mission: Trojan Tour and Rendezvous

Goal: visit a cloud of Trojan Belt objects (possibly displaced ancient TNOs) to determine composition, evolution and presence of organics

Multi-phase mission profile:

- Decade-long cruise phase,
- Punctuated by three-week flyby opportunities,
- With a single extended orbital rendezvous



Reference: http://www.nap.edu/reports/13117/App%20G%2016_Trojan_Tour_and_Rendezvous.pdf

Page 1



Reference Mission: Trojan Tour and Rendezvous

Key measurements

1. **Chemical composition** via Gamma Ray, Neutron, UV and IR mapping spectrometers
2. **Geologic state and surface evolution** via wide and narrow angle cameras, LIDAR, thermal imager
3. **Bulk physical properties** via cameras, radio science
4. **Outgassing**, monitored with UV spectroscopy and high-phase imaging

Capability needs

- **Revise science goals.** Very low albedo mean objects are difficult to study from Earth. The Trojan population is poorly characterized, and science goals may change
- **Satisfy difficult New Frontiers cost cap** – cutting ops cost could buy more RTGs, permitting more flybys or rendezvous options and a better population study
- **Execute rapid trajectory and/or slew changes** to exploit newly discovered objects and gravity field measurements
- **Autonomously target surface features and/or outgassing** to improve flyby science return with 1+ hour light time delay
- **Avoid safe mode during a flyby**, since it results in irretrievable data loss



Page 2



Mission requirements: Trojan Tour and Rendezvous

Unconstrained physical environment

Recognize and respond to navigation hazards
% chance mission failure from collision

Spacecraft cost and complexity

Meet New Frontiers cost cap
Total mission cost accounting for propellant mass

Unexpected science phenomena

Identify outgassing for targeted data collection
Probability of capturing an outgassing event

Detect and target outgassing, volatiles
% probability of unambiguous detection

Detect and track new objects during cruise
Number of new objects discovered at nominal Trojan cloud density

Operations efficiency and cost

Replan rapidly to newly-discovered or binary objects
Planning turnaround time, days

Plans should be robust to unknown environments
Science instrument coverage preserved in worst case navigation scenarios



Interstellar Probe Reference Mission Summary

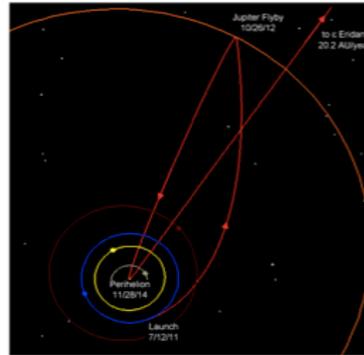
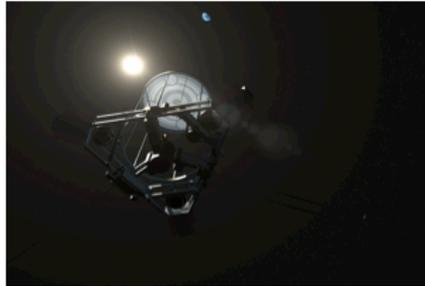


Reference Mission: Interstellar Probe

Interstellar Probe would have a mission duration spanning human lifetimes and light time delays of months or years. This would require an unprecedented level of resilience. It would formulate and perform experiments autonomously and summarize the results for downlink. It would have to navigate completely unknown environments, enduring for decades to complete its mission.

Goal: Travel distances greater than 1000AU

- Understand ISM and its implications for the origin/evolution of matter in Galaxy
- the structure of the heliosphere and its interaction with IS environment
- fundamental astrophysical processes that can be sampled in situ
- Rendezvous with extrasolar planet



Trajectory to Jupiter, Sun and ε Eridani (10.7ly from Earth)

Notional spacecraft design: Probe post-perihelion with optics deployed. Three RTGs and three spherical tanks for the cold-gas attitude control system. View is from the rear with the optical communications system pointed toward the Earth for data transmission. The diffraction patterns in the main optic for the fine-control star tracker and communications system are clearly visible. The magnetometers and plasma wave antennas are also shown deployed for flight.



Reference: NIAC 7600-039 Phase II Final Report: A Realistic Interstellar Explorer, 14 October, 2003

Page 1



Reference Mission: Interstellar Probe

Key measurements by evolutionary mission approach

1. 1.0: fields and particle detectors; demonstrate navigation and comm
 - Magnetometer and plasma wave sensor
 - Energetic particle spectrometer – solar and galactic cosmic ray isotopes
2. 2.0: telescope to detect planets transiting stars, and atmosphere in dis-equilibrium; repair, rebuild, reconfigure; 3D print for repair
 - Lyman-alpha imager
 - X-ray photometer
3. 3.0: s/c figures out the most interesting thing to do when it reaches its destination

Capability needs

- Reach significant penetration into ISM (~1000 Astronomical Units) within the working lifetime of initiators (<50 yrs)
- Efficiently hibernate during long-term cruise in deep space; hands-off ops; severely bandwidth limited so only highly reduced data can be transmitted
- high speed (>200AU/yr speed); Long-duration propulsion
- Long-lived (>50 yrs), self-healing architectures and redundancies; robust, reliable and adaptable
- Survive multiple radiation environments – solar, Jovian, ISM
- Stringent pointing requirements to communicate with Earth - tight control/comm coupling
- Self-characterization and characterization of unknown environments
- Develop and execute appropriate science observations
- Avoid safe mode during encounters, since it results in irretrievable data loss - duh



Page 2



Mission Capabilities and Metrics: Interstellar Probe

Unconstrained physical environment

Recognize and respond to navigation hazards
% chance mission failure from collision

Characterization of unknown environments
Response to potential hazards in ISM;
surviving tolerable rate and energy of cosmic rays

Spacecraft cost and complexity

High speed
AU/year

Long-duration propulsion
of years/duration of propulsion; MTBF and lifetime issues

Self-characterization/self-healing
Mean time to recover from or adapt to faults

Stringent pointing
Maximum pointing error

Long-lived (>50 yrs)
Tenacity factor

Unexpected science phenomena

Science Data Compression
Data compression ratio

Autonomous Data Prioritization/Target selection
of distinct science features in downlink

Operations efficiency and cost

Efficient hibernation
Size of staff required; # of cmd cycles req/yr



Mission Capabilities Aligned with Resilience Capabilities: Interstellar Probe

Unconstrained physical environment

Recognize and respond to navigation hazards
Rapid Mission Planning; Data Processing

Characterization of unknown environments
Data Processing; Rapid Mission Planning;
Tenacity (a.k.a. Fail Opportunistically;
Graceful Degradation)

Spacecraft cost and complexity

High speed
?

Long-duration propulsion
?

Self-characterization/self-healing
Tenacity

Stringent pointing
?

Long-lived (>50 yrs)
Tenacity

Unexpected science phenomena

Science Data Compression
Data Processing

Autonomous Data Prioritization/Target selection
Data Processing; Rapid Mission Planning

Operations efficiency and cost

Efficient hibernation
Low Cost Cruise



APPENDIX F: ARCHITECTURE FOR RESILIENCE—A REPRESENTATIVE EXAMPLE

Figure F-1 shows a spacecraft software system organized in a hierarchy of processes. The top process is the overall master process (sometimes referred to as a ‘sequencer’ or an ‘executive’) that tells the systems what to do, what to accomplish, or how to behave, for the current phase of the mission. This kind of architecture is common. The domain of discourse for all the estimators and controllers in this software system is also the usual things: sensor readings, attitude estimate, commanded pointing direction, telemetry buffer space available, temperatures, etc. In addition, there are models of some of the onboard spacecraft devices. The attitude control system (ACS) will have gyro models, thruster models, etc. In addition, the telemetry system will have a model of its buffers to determine when a buffer is full, a model of its switches and coax connections to enable it to move from one transmitter to another, etc. Some of these models may be considerably more sophisticated than the others, but they are models nonetheless. The domain of discourse of these models is that of the behavior of the system under control and the environment. We will call this *spacecraft software system* and its associated domains of discourse, the *base control system*.

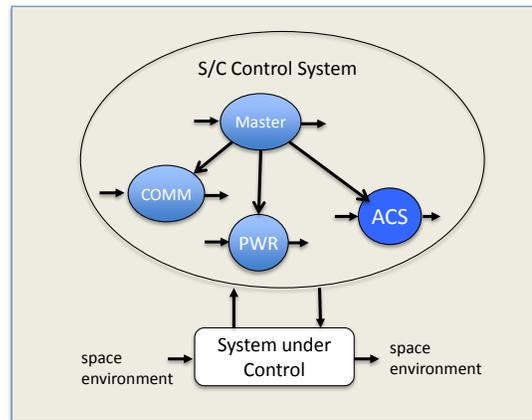


Figure F-1: A Spacecraft Software System

A resilient spacecraft will have a larger domain of discourse, and while some items are the same as in the base control system, there will be higher level entities in its vocabulary. We try and separate these domains, that is, the domain of the base system, and the domain of the additional resilience meta-control component. Looked at as a control system, the resilience meta-control component will have estimators and controllers whose domain of discourse is not a sensor reading, current attitude, and the like, as that is the domain of the base control system. Rather, the domain of discourse of the resilience meta-control component is ACS performance, a measure of how well the ACS system, given the current sensors, actuators, models, and estimators, etc., is doing the required job. Performance may include the power being used, say, to run the sensors and actuators. The job of the resilience unit is to estimate this performance and to control the ACS subsystem to achieve the performance desired. Consider a case where power expended is too large and is determined to be the cause of performance falling below the control point. The resilience controller may decide to change the sensors, estimators, and controllers, and actuators of the ACS system to an arrangement that improves performance, perhaps by using less accurate (but good enough) gyros that use less power to do the job.

The resilience meta-control component controls the ACS system not by using terms from the base system domain but by manipulating terms from the resilience domain. Figure F-2 illustrates the idea. In Figure F-2, the base control system exists and does its usual job running the spacecraft. But there is another control system, the *resilience meta-control system*, responsible for adjusting the base control system by working with, modifying, changing, and indeed controlling how the base control system is put together. The resilience system will need models also,

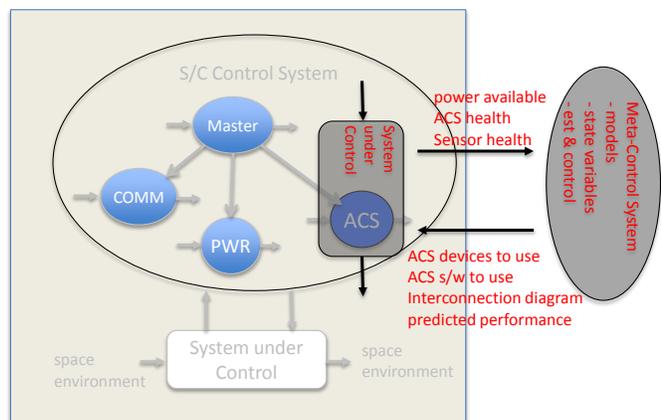


Figure F-2: The Spacecraft Control System with Meta-control for Resilience

and its models will be of the ACS system—the devices used, how they are wired, the switches, etc., all the things necessary to build an ACS system. The resilience meta-control component also controls the software and how it is put together. Its models in this case are of the functions the software performs, how fast it needs to run, and how to hook things up. But if we design a good resilience meta-control system, the rules for how to hook things up are not developed by following rote lists of components to be put together, but rather by running models and synthesizers that find their way through arbitrary components, hard and soft. In fact, an idealized resilience meta-control system would start with nothing but the components and put the spacecraft together from scratch, given a description of what they do and what they need.

Finally, the resilience meta-control component shown in Figure F-2 is shown as a standalone element, but just as the base system has a hierarchy of controllers, so will the resilience meta-control system. However, the two systems are distinct, each with a different hierarchy.

It should be noted that some of those things estimated and controlled by the base system and the resilience meta-control system have overlap, and in some cases it may be hard to decide which goes where. For example, some current base control systems decide when to replace a failed gyro with another. Our argument would be that such behavior belongs to the meta-system, not to the base system. The two systems work with different things, and it is a good idea to separate them when possible.

Of course, there are many more elements needed to build a resilient system, but *domain of discourse* might be a useful principle in a finished architecture. Also, note that the ideas presented here are in the extreme. For example, the base control system and resilience meta-control system represented orthogonally in Figure F-2, would likely cooperate in some way, but the less cooperation needed, the better.

APPENDIX G: DETAILED REPORT OF CAPABILITIES FOCUS GROUP OUTCOMES

The goal of the capabilities survey was to look at existing capabilities and formulate lists of the following:

- Enabling software and autonomy technologies (e.g., middleware, languages, frameworks, etc.)
- Key processes for agile and verifiable development enabling resilient system development and management of complexity

To achieve these goals, we started with a series of one-hour teleconferences, which led to several presentations being given at JPL. A single all-day meeting was held at Caltech to capture a broad set of capabilities. Since capabilities are such a broad subject, there is no way we could deal with everything in this limited time. We surveyed the Capabilities Focus Group and developed the following set areas of interest:

- SW design patterns, DSLs, models for software synthesis, and automated testing, standards
- Autonomy patterns, AI planners, self-adaptive software
- FDIR patterns
- Technologies taken from other domains (e.g., automotive manned air craft, UAV, UUV, etc.)

These areas were selected based on the group's expertise and interests. Based on these areas of interest, the following sections were derived and we report a summary of our discussions here. Details of the capabilities are captured in the "Capabilities Subgroup: Capabilities (and Architectures) Outbrief," 26 Feb. 2013 (www.kiss.caltech.edu/workshops/systems2013/index.html) as presented by co-lead Leonard Reder. In the following, we summarize the key discussions that came out of this effort.

Capabilities ⇔ Architectures: Latent vs. Active

Initially, we discussed the relationship between capabilities and architecture (Figure G-1). The initial question is: "Do capabilities drive architecture or does architecture drive capabilities?" Although there was no consensus, some insights concerning the notions of latent or passive resilience versus active resilience were noted.

Active resilience is analogous to reflexes triggered by specific disturbances. In the power grid, for example, the feature of 'on-the-fly routing' is providing active resilience. In the human body, most of what we're good at are automatic and unconscious reflex reactions that we classify as active resilient capability. For example, when we are young, we learn to rapidly pull away from touching something hot—active resilience toward avoiding a burn.

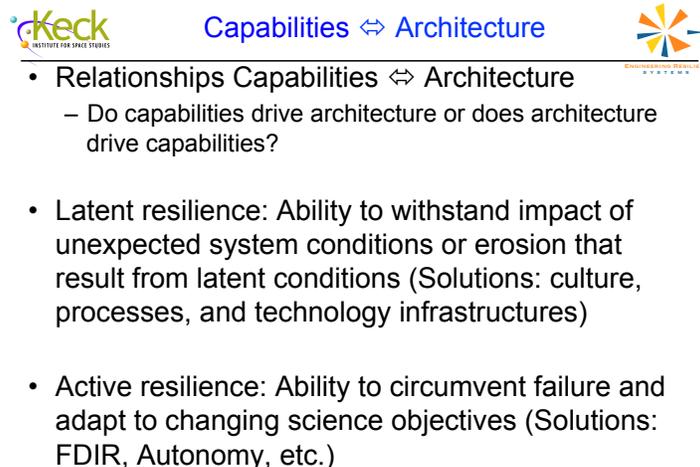


Figure G-1: Capabilities versus Architecture

Latent resilience responds to longer-term goals. Our brain's architecture supports moving things around flexibly. Latent resilience enables an ability to change what one is resilient to, both in terms of sensing and reaction (e.g., machine learning in robots). The difference between capabilities of human pilots and robots, for example, is that we do things a lot better at a higher (latent) level. However, robots are ideal for most real-time (active) resilience in the bottom layers. Humans are not good at active resilience because we're slow. In piloted vehicles, we're heavily augmented with respect to active resilience because we're so slow. But we're extremely good at latent resilience and unable to build robots that have anything near our latent resilience. The handoff is not handled well. The adaptive nature of the brain is not easily reproduced.

The research question turns out to be: "What's the relation between active and latent resilience?" Active resilience is preprogrammed rapid responses while latent resilience is tuning these conditions as challenges change via learned responses. That has not been the AI picture of things, which has been not to think of most resilience coming from automated, fast, real-time, unconscious processes that are tuned by a latent resilient system. Consider the human thought process—we move from application-level thoughts (latent) to middleware reactions (active) using our brains. Today's technology attempts to mimic this. However, we embed capability into hardware when it becomes ubiquitous or has to have high performance. The closer a system works at the boundary of theoretical performance, the harder it becomes to develop fallback strategies. There is a mechanism that does that switching between constant feedback (providing active resilience) and adaptability (providing latent resilience). The fundamental problem is this handoff mechanism at the architectural level is not well understood.

Today, handoff between latent and active resilience is enabled by use of various architectural layers. Reflexive processes (in the active layers) buy time, which keeps the system going so that other latent (or deliberative) mechanisms can adapt. Good built-in reflexes will keep the system alive until it can figure out at the high levels (latent) how to adapt. Back and forth between these two layers is important. We often can't build architectures that allow back and forth. Layered architectures do a very rigid thing today. For example, TCP/IP doesn't allow adaptability; you would not want to have it in this situation, yet it is an example of a seven-layer architecture. The human brain is amazing—things can become inconsistent across layers within the brain but consistency is restored. At the top layers (deliberative) they start solving for a different goal than at the bottom layers (reflexive), but both top and bottom layers with time seamlessly interact to converge on uniform behavior. In today's systems, if we build in more flexibility, the interfaces become more complex and might not work, but more rigid restrictions on how layers interact results in less likelihood of failure at the expense of less adaptability. The challenge is how to create the interfaces so there is flexibility of human brain-like functionality without interfaces to these various layers breaking.

The point of the above discussion is that architectures are a way of providing capabilities of active versus latent resilience but these capabilities are essential to enable next generation architecture development. Latent resilience is resilience without diversity (not doing something different); active resilience is taking a different action. Both these forms of resilience are realized by a wide variety of low- to high-level capabilities. With the following discussions, we list and discuss a series of capabilities that exist today, which were considered by the subgroup.

Capabilities-Enabling Latent Resilience

Capabilities-enabling latent resilience can be divided into two classifications: structural and behavioral. Behavioral latent resilience capabilities include various forms of machine learning, planning, deliberative automated reasoning implementations, and many others. Our group did not discuss these extensively. We dealt more with existing capabilities that could enable structural resilience. In Section 4.3, a roadmap of several capabilities common to our reference missions are presented that enable behavioral latent resilience. Structural resilience is the notion that we can incorporate certain design structure or process that will enable latent resilience.

Currently, a safety and mission-critical development process is used for our missions (Figure G-2). When we use the safety-dominated approach, we assume a rigid set of behaviors for the system and all V&V items can be accounted for. With this rigid assumption, we are not going to be resilient to the unexpected. Systems with behavioral latent resilient features are not verifiable.

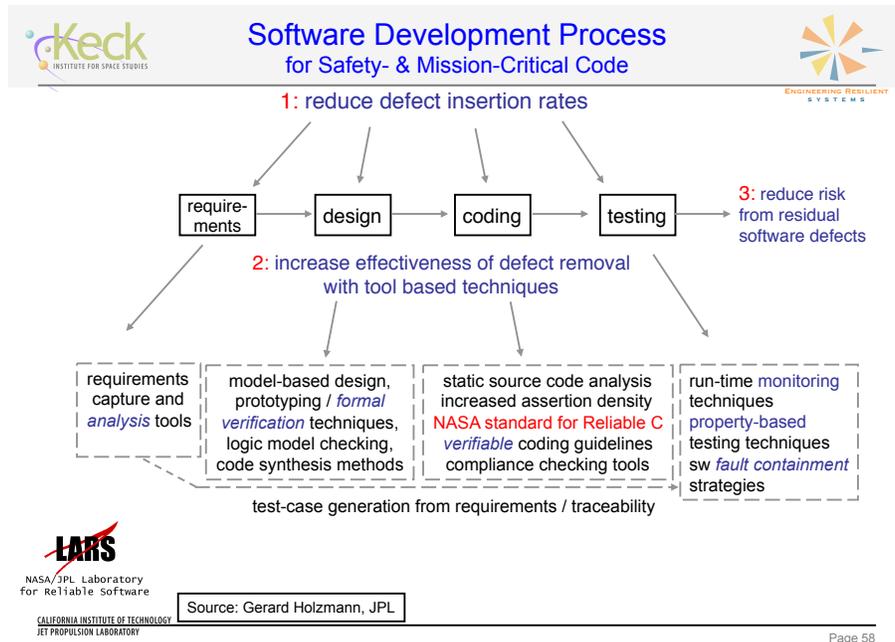


Figure G-2: Today's Development Process

Currently, we are routinely researching common formal verification tools for enabling structural latent resilience of our systems through verification of design. Examples of popular approaches are:

1. [SPIN](#) (Holzmann, 1997)
 - a. Explicit-state model checkers
 - b. Generate and explore every possible state
2. [Symbolic Model Checking](#) (SMC) (“Model checking”, 2013)
 - a. Manipulates entire sets of states at once
 - b. Set consists of all states that satisfy certain logical conditions
 - c. Such conditions are encoded as (ordinary) Binary Decision Diagrams (BDDs) and can be complex to code
3. [Bounded Model Checking](#) (BMC) (“Model checking”, 2013)
 - a. Use satisfiability (SAT) solvers instead of BDDs
 - b. SAT solvers implement decision procedures based on propositional logic
 - c. SAT solvers perform approximate verification by exploring the model only to a given depth
4. [Runtime Verification](#) (RV) (“Runtime verification”, 2013)
 - a. Is performed at run time
 - b. Is a lightweight verification technique that complements traditional a priori verification
 - c. Checks whether the current execution of a system satisfies or violates a given correctness property
 - d. Uses a monitor to decide whether the execution is correct
 - e. Monitors are automatically generated using a high-level specification language such as SALT (Structured Assertion Language for Temporal Logic)

- f. Can be used to check correctness of partially verified systems (verification based on assumptions about operational environment)

Other capabilities exist in the form of formal specification languages that can be used to synthesize correct design models. For example, we discussed a unique tool named [FORMULA](http://research.microsoft.com/en-us/projects/formula) (<http://research.microsoft.com/en-us/projects/formula>) that accepts a formal constraint-based language and from constraint specifications generates a model that satisfies these. In addition, Richard Murray's group at Caltech is actively developing mechanisms to perform both offline and online optimization and synthesis of system designs enabling structural resilience (Wongpiromsarn, Topcu, and Murray, 2010). Tools of this kind are currently research grade, but provide a basis to produce system designs with guaranteed analytical provable behavior.

If more verifiable architectures are desired, the group considered the need to dial in different thinking about layers to ensure structural latent resilience. It was observed that human low-level (e.g., reflexive) behavior has not changed much. These reflexive or homeostatic mechanisms of humans do not change and are rigid. If we start to think about layers of capabilities within a spacecraft, we realize that various levels of trust will exist in every layer. The more rigid or homeostatic spacecraft behavior layers would have the highest degree of trust, while the higher deliberative layers would then have less trusted behavior.

Implementation of the reflexive lower layer discussed above, is enabled with a large variety of technologies that are currently deployed but dismissed for use in deep space missions. Currently, our flight software systems are monolithic sets of multiple threads of sequential execution. The notion of encapsulating with components and abstracting away interface complexity using ports and connectors is currently state-of-the-art software engineering, but is not used in deep space spacecraft projects. Representation and synthesis of the component-port-connect code deployment model from standard architectural description languages (such as Unified Modeling Language [UML] (Object Management Group, 2011) and Architecture Analysis & Design Language [AADL] (Feiler and Gluch, 2012)) is useful and beginning to be used within the deep spacecraft domain.

Current JPL-developed planetary spacecraft do not reuse many software technologies that potentially could be taken advantage of. Rationale for not adopting existing software capabilities is that to achieve the reliability required, we must design and test from scratch. The group discussed a variety of software technologies (some already in use aboard the International Space Station) that could potentially be adopted for new projects. Some of the technologies covered include:

- Robotic Operating System ([ROS](#)) (Quigley et al., 2009) is an open source framework for rapidly developing and characterizing robotic systems. It is currently used in the implementation of Robonaut 2 aboard the ISS. ROS contains some machine vision software capabilities also, similar to the public domain [OpenCV](#) software for computer vision (Bradski and Kaehler, 2008).
- [Coupled-Layer Architecture for Robotic Autonomy \(CLARAty\)](#) (Volpe et al., 2001) is a software framework developed to integrate autonomy and control capabilities for robotics applications. It is open source and a collaborative university effort. Besides this particular framework, there are a wide variety of frameworks in existence, which we discussed briefly.
- Use of open standards to enable reliable middleware functionality and adaption of architectural standards were discussed. The Object Management Group (OMG) (www.omg.org) standards have matured over the years and are used reliably in various domains. Of particular interest was the Data Distributed Services ([DDS](#)) (Object Management Group, 2007) middleware standard and implementations. DDS is a distributed publish-subscribe middleware system with various real-time guarantees built in. For example, one can dial in Quality of Service (QoS) characteristic requirement constraints such as required latency. DDS is used for many applications and is primary technology for the Human Exploration Telerobotics Project demonstration flying on the ISS.

- At JPL, within Division 34, there is considerable interest in avionics standard ARINC 653 (ARINC, 2013). ARINC 653 is a standard for avionics real-time operating systems. The intention is to provide both time and space partitioning of software modules within a single CPU environment. Originally this standard was developed for military and commercial aircraft but JPL is researching use within a core FSW architecture currently in development. Space partitioning is simply providing isolation of memory spaces between software modules for safe operation. Partitioning in time restricts individual tasks to be executed within specific time slots. Use of the time slot-based scheduling restriction is potentially a problem for low-latency event driven systems.
- AFRL [Space Plug n' Play Avionics \(SPA\) \(Lyke, 2007\)](#) is another interesting standard that was developed within the AAIL and is maturing. SPA hardware and software concepts that make plug-and-play possible are:
 - Self-description: components describe themselves using eXtended Transducer Electronics Datasheet (xTEDS)
 - Discovery & join: automatic recognition when plugged into a system or networked
 - Satellite Data Model (SDM): software that binds together other pieces of SPA
 - Push-button tool flow: using web-based software tools to translate a user's ideas directly into a buildable spacecraft (e.g., wizard-driven dialogs)
 - SPA has been implemented in a variety of system forms since 2004, ranging from coffee cup-sized CubeSats to 400-kg (882-lb) tactical spacecraft.

Both Draper Labs and JPL have been funding the use of massively parallel (49- to 64-core) multicore processor R&D prototypes to demonstrate increased structural latent resilience capabilities. Both efforts use TRN algorithm deployment. Draper Labs has an advanced planetary landing system with emphasis on the lunar test case, whereas JPL's TRN testbed was put in place to develop advanced autonomous landing techniques for future Mars missions. Both institutions have been testing algorithms on the Tiler architecture that comes in a 64-core commercial grade chip with migration path to a 49-core, radiation-hardened, space-qualified chip, known as Maestro (Figure G-3) (Malone, 2009).

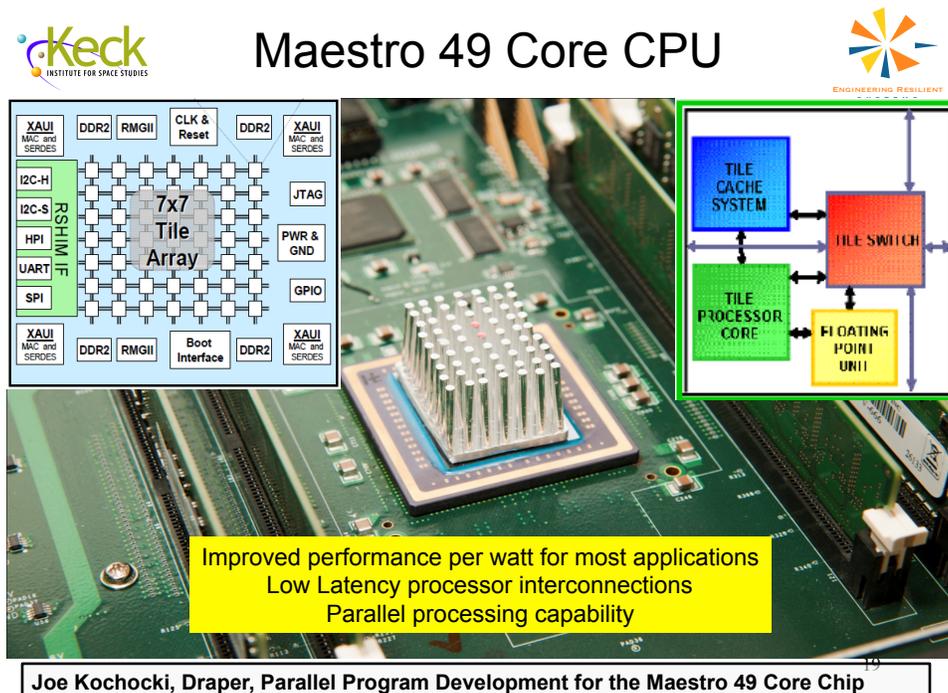


Figure G-3: Maestro 49 Core Rad-Hard Processor—an Enabling Technology

Multicore processors are general-purpose redundant computers. They have great potential in enabling the deployment of various high-level capabilities discussed later in this appendix (e.g., AI planning deployments, fault protection, etc.). Multicore provides the ability to place cores in various work configurations where they can be dynamically reconfigured. Abilities to turn cores on and off allow a greater fidelity to control use of power.

The JPL Strategic R&D task, Demonstration of Multi-Core System Software for Fail-Operational Flight Computing, used the JPL-developed TRN algorithms to demonstrate fail operational and graceful degradation fault recovery scenarios. The Tiler tile-64 chip was used for the initial effort and later the software was executed, under radiation test conditions, on a Maestro chip. For the *fail operational* scenario with hard real-time constraints, TRN state estimation was selected for adaptation and demonstration. Triple modular redundancy (TRM) was used to demonstrate redundant Kalman filters could fail and recover in real time without estimation data being affected. To demonstrate *graceful degradation*, we have selected the TRN function for landmark image data processing and correlating it with stored map data for adaptation. This capability is achieved by parallelizing the image processing function of the TRN code using multiple Tiler processing elements (as many as 40 processing elements) with each executing concurrently only on a subframe of the entire image frame. In the event of a core failure, the remaining healthy cores for landmark processing will continue to function and a new core instantiated with failed cores functionality to recover full image feature detection capability.

Draper Labs has performed similar demonstrations for JPL but has also characterized Maestro performance. Each Maestro core provides a 25–30% performance enhancement over RAD750 when operated in cache. The RAD750 is has been the JPL standard flight processor flown on missions such as MSL. More performance details are:

- RAD750 provides ~80 MFLOPS when operated at 133 MHz, 11 W nominal power
- One Maestro core provides 100 MFLOPS when operated at 260 MHz
- Maestro provides 50x = 5 GFLOPS at 260 MHz, 20 W nominal power

Improved performance of the Maestro and other future generations of multicore will be a significant enabling capability of future resilient spacecraft.

Capabilities-Enabling Active Resilience

As stated above, we consider active resilience to be reactive and rapidly triggered behaviors. Current deployment of autonomy and fault protection is therefore classified in this section as active resilience–enabling capabilities. A brief overview of autonomy techniques currently used within flight systems was discussed:

1. Automated planning
 - a. Onboard commanding and resource management
 - b. Automated response to faults/opportunities based on constraints and resources
 - c. Examples: CASPER (Chien, Knight, Stechert, Sherwood, and Rabideau, 1999), Remote Agent Experiment (RAX) (Bernard et al., 1998; Muscettola, Nayak, Pell, and Williams, 1998), RHex (Saranli, Buehler, and Koditschek, 2001) and CREST (Woods et al., 2008) robots, etc.
 - d. Seeing significant use for AUVs/UUVs
2. Smart executives
 - a. Shorter-term onboard response
 - b. Enables command conditionals, looping behavior, retries, etc.
 - c. Examples: Virtual Machine Language (VML) (Grasso, 2002), Spacecraft Command Language (SCL) (Buckley and Vangaasbeck, 1994), Titan (Williams, Ingham, Chung, and Elliott, 2003), etc.

3. Onboard data analysis
 - a. Identify new science targets or opportunities
 - b. Prioritize data for downlink (getting it to ground faster)
 - c. Data summary (gets summary information to ground faster)
 - d. Lots of visual image analysis; some hyperspectral analysis
 - e. Examples: EO-1 (Chien et al., 2005), AEGIS (Estlin et al., 2012), MER dust-devils (Castano et al., 2006), etc.
 - f. Swift astrophysics mission (Gehrels et al., 2004): slews spacecraft when gamma-ray bursts are detected
4. Automated navigation
 - a. Enable vehicle to avoid problems and reach goals efficiently (GESTALT (Goldberg, Maimone, and Matthies, 2002), DIMES (Cheng, Johnson, and Matthies, 2005), etc.)

As part of the active resilience capabilities, the group considered fault management architectures (National Aeronautics and Space Administration, 2012). All current missions utilize some form of the typical monitors/alarms and responses architecture. Many different designs have flown but most implementations use the following elements: error monitors (some form of sensing off nominal conditions), responses (some form of automated attempt to fix a fault), and some coordination mechanism (commonly called a *FP engine*) to manage monitor output to response input mapping.

Currently, there are three fault management architectures, which the focus group discussed: *monitor-response-inhibit* (Figure G-4), *goal-based execution* (Figure G-5), and *model-based execution* (Figure G-6).

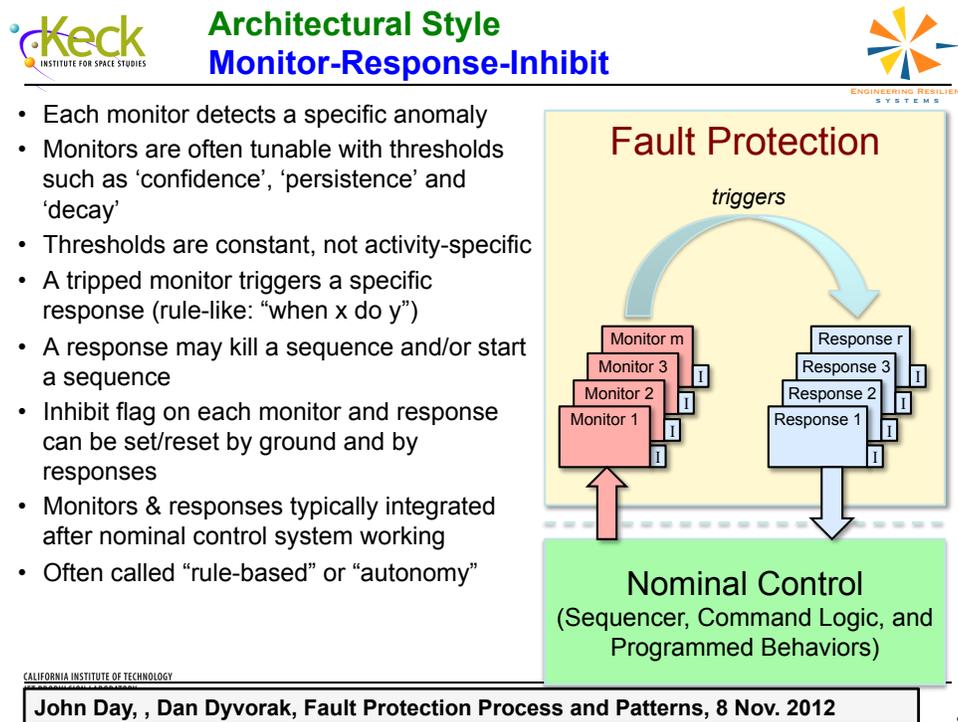


Figure G-4: Monitor-Response-Inhibit

The monitor-response-inhibit style fault protection is the one typically used in spacecraft with the addition of a FP engine for the delivery of monitor status to responses. With this approach, everything must be enumerated as to all the functionality and possible anomalies that can happen. It is essentially asking what the failure modes are and if they can be reduced to a manageable set. The approach is very labor intensive

but for simple systems, it is very testable since all the monitors and responses are known and the combinations can be tested.

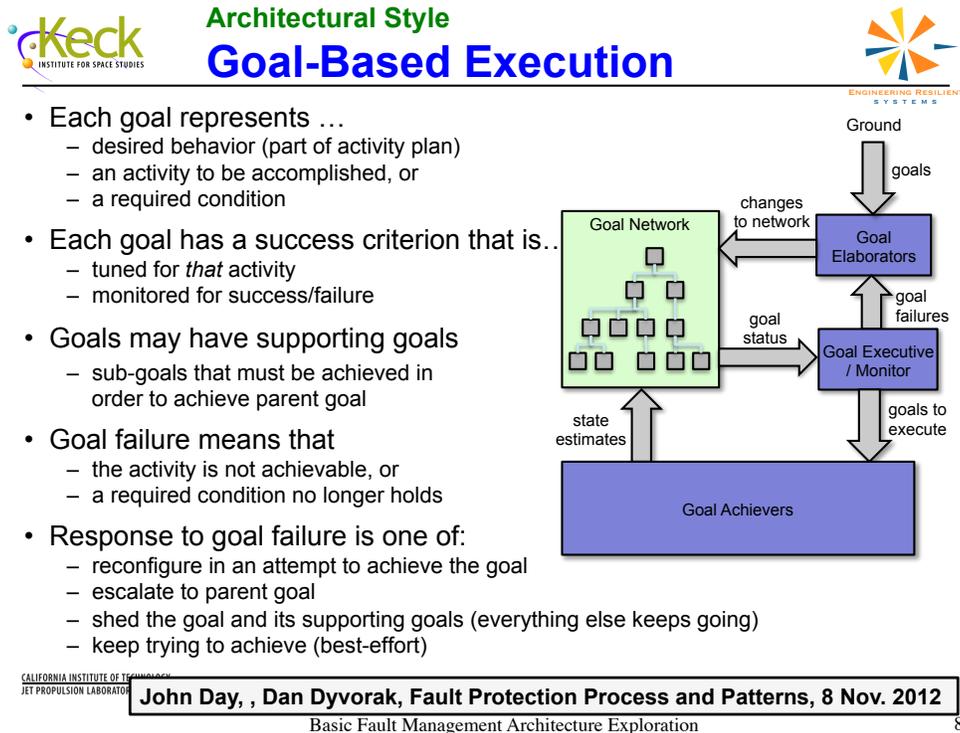


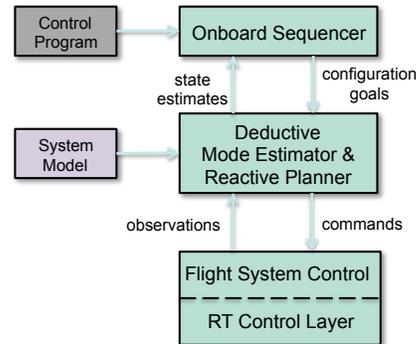
Figure G-5: Goal-Based Execution

The goal-based execution style of fault protection distributes the fault protection function amongst a network of goals. The notion of the fault protection engine and responses are replaced by a notion of goal networks. If a goal is constrained, either it does nothing or it is re-elaborated as other goals and this process continues. This style of fault protection was implemented at JPL as the Mission Data System (MDS) architecture (Ingham, Rasmussen, Bennett, and Moncada, 2005). It has the promise of scaling more effectively for larger more complex systems than the monitor-response-inhibit style; however, manual coding is still required to realize the functionality of the goals as tactics.

The third style of fault protection is the model-based reasoning style shown and explained below (in Figure G-6). This style of fault protection was implemented at MIT in the Titan architecture (Williams, Ingham, Chung, and Elliott, 2003). It is easy to conceptualize this approach, but implementing the models to realize a fully reliable system can be challenging.

In general, these methods of fault protection are nonscaling. The problem is when you have hundreds of monitors and related actions you never know what will happen if things interfere, etc. A persistent question is: “How do you validate it?” Some of the problems might improve by considering hybrid system architecture approaches. This proposes taxonomy in these three architectural styles. Potentially, we could start thinking about these styles as dimensions or layers. At the lower layers is one style, at the top layer another, etc. We should not presume in general that a given style is good for all layers. But again, the interplay between the layers is a good research question.

- The “system model” is a connected set of component models, each one describing component behavior for nominal and fault modes
- The deductive mode estimator compares observations to model-predicted state. If they are inconsistent ...
 - It deduces the most probable fault mode, or
 - It concludes “unknown system state”
- The reactive planner searches for alternate ways to achieve the goals
- There is very little fault protection code, per se, because fault detection, diagnosis and response result from reasoning over the system model



CALIFORNIA INSTITUTE OF TECHNOLOGY
JET PROPULSION LABORATORY

John Day, Dan Dyvorak, Fault Protection Process and Patterns, 8 Nov. 2012

86

Figure G-6: Model-Based Execution

Autonomy and Planners

The notions of active resilient and behavioral latent resilient capabilities blur when we start to consider autonomy. State machines are currently the most common mechanism used for autonomous behavior. The decision to go to safe mode is a reactive thing done based on system state, which enables active resilience and is usually implemented via a state machine. In more sophisticated systems, automated planners often sit at the top and have a global system perspective. At the next level down, another autonomy system serves as a smart executive, which is not looking as far. It has a shorter time window so its ability to reason about long-term plan objectives is limited, but it can do reasoning above executing set of commands—we consider it enabling active resilience. Planners are responsible for generation of longer time window plans, but may or may not be deliberative in their operation, so for our discussion we still consider them enhancing the active resilience of the system.

As part of the Capabilities Focus Group’s work, we generated a brief history of autonomy to example how this capability evolved. The history of autonomy is summarized in the following list:

1. Before the 90s:
 - a. Attempt to replicate human reasoning
 - b. Operators generated desired sequences manually
2. From the 90s:
 - a. UML simple state-machines (Harel, 1987)
 - b. Desire scheduling with resource allocations (comes along Automated Scheduling and Planning Environment (ASPEN) (Rabideau, Knight, Chien, Fukunaga, and Govindjee, 1999), CASPER, or EUROPA (Frank, Jónsson, Morris, and Smith, 2001) planners)
 - i) ASPEN and CASPER use a temporal constraint language to compute an optimized set of actions based on the constraints given. It is essentially an iterative tree search solution.

CASPER is designed for embedded online applications whereas ASPEN is designed for offline use.

- c. FDIR
 - i) Execution monitoring for off-nominal conditions
 - ii) Isolate and identify failures
 - iii) Fault responses (safing, fail-operational)
 - iv) Desire to generate command sequences from models automatically
 - v) 1999 demonstrated on DS-1 with spacecraft fault protection based on automatic code-generation techniques (Rouquette, Neilson, and Chen, 1999)
 - vi) Remote Agent Livingstone mode identification and reconfiguration (Williams and Nayak, 1996)
- 3. From 1995 to present:
 - a. Better models and not hand coded
 - b. Planning to enable low-level hardware configuration
 - c. i.e., CASPER doing ‘activity planning’—classical planners
 - d. Actions and goals with constraints; too hard for 1990s planners, now achievable
 - e. Good success in automated navigation for spacecraft
- 4. Example technologies from MIT group (Williams et al.)
 - a. Burton, “A Reactive Planner for a Model based Executive” (Williams and Nayak, 1997)
 - i) Developed for DS-1
 - ii) Reactive (online) planner
 - b. Burton (Wang and Williams, 2014)
 - i) Fast offline planner
 - ii) Algorithm decomposes problem and dependencies flow one way in an order
 - iii) Developing piece actions for large state spaces: goals to achieve task
 - iv) Goals changing over time; actions are temporal and constrained

A survey of AI planners developed outside of MIT, which is by no means all inclusive, resulted in this list:

1. UCPOP (Penberthy and Weld, 1992)
2. GraphPlan (Blum and Furst, 1997)
3. BlackBox (Kautz and Selman, 1999)
4. Fast Forward Planner (Hoffmann, 2001)
5. Fast Downward (Helmert, 2006)
6. CRIKEY/COLIN/OPTIC (Coles, Fox, Halsey, Long, and Smith, 2009; Coles, Coles, Fox and Long, 2012; Benton, Coles and Coles, 2012)
7. LPRPG/LPG (Coles, Coles, Fox and Long, 2011)
8. SGPlan (Hsu, Wah, Huang, and Chen, 2006)

Redundancy

The topic of redundancy was considered by Erv Baumann of Northrup Grumman, and the group considered some capability examples. The following traditional and additional levels of redundancy are proposed for long-duration space missions involving multiple identical (or sufficiently similar) spacecraft (with new additional levels indicated in ***bold italic***):

1. Collective (e.g., multiple aircraft or spacecraft in collaborative formation flight)
2. System level
3. Subsystem level

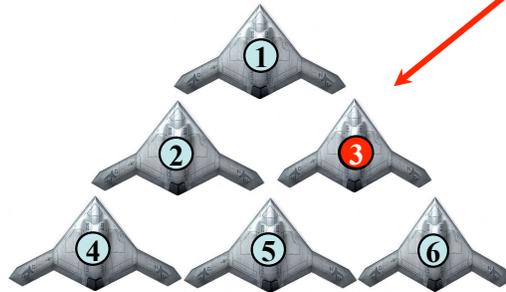
4. Unit/box level (called “Line Replaceable Units” by Air Force, “Weapons Replaceable Units” by Navy, and “Orbital Replacement Units” by NASA)
5. Board or component level (inside the boxes, some people combine this with level 4)
6. ***Subcomponent level (e.g., multicore or other highly replicated chip-level hardware)***
7. ***Lower tessellated/fractalized levels (at or below multicore level, including the ultimate evolution through microbots and nanobots to programmable matter)***

The idea of collective redundancy is best described by several examples. Consider the example of multilevel FDIR response to degraded flight control actuation system (Figure G-7). The scenario is as follows:

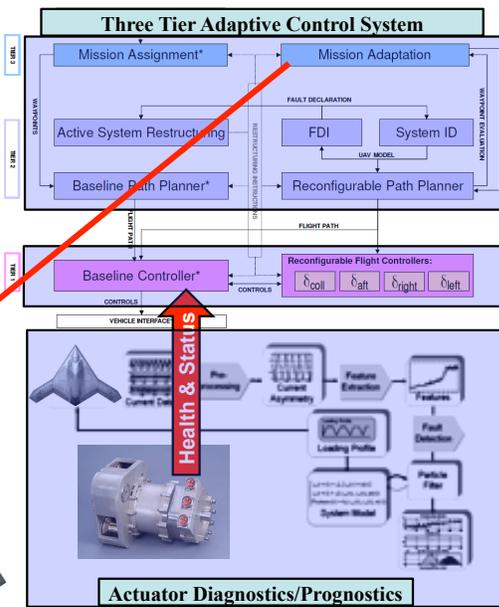
1. Flight group embarks on mission with all systems functional and all aircraft in flight group periodically exchanging health & status (H&S) information (Figure G-7 [top]).
2. Nearing the target area, one of the key aircraft in the flight group experiences degradation or loss of EMA control authority resulting in severely reduced flight-control performance and vulnerability to attack. Propulsion, sensor, and weapons systems are fully functional.
3. Control system reconfigures to maintain controllability and ‘baby the aircraft’ to manage/maximize remaining useful life (RUL) (Figure G-7 [bottom]).
4. Due to the critical nature of the damaged aircraft’s payload (primary ordnance and/or sensors required to destroy this high-priority ‘target of opportunity’) and time-critical nature of the mission, the flight group’s mission management software decides to press on while automatically reconfiguring its aircraft flying formation and tactics by moving the degraded aircraft to the inside of the formation where it will be less vulnerable.

In-Flight Detection & Mgmt of Degraded Actuation System

1. Flight group embarks with all systems operational.
2. Nearing the destination IHM detects degradation of control surface actuation and maneuverability in one of the key reconnaissance aircraft in the flight group.
3. IHM provides health & status diagnostic & prognostic information required by adaptive flight control system.
4. Control system reconfigures to maintain control and manage Remaining Useful Life (RUL) of the actuators.
5. Automated Mission Mgmt software automatically changes flight group formation and tactics to reduce vulnerability of degraded aircraft for remainder of mission.



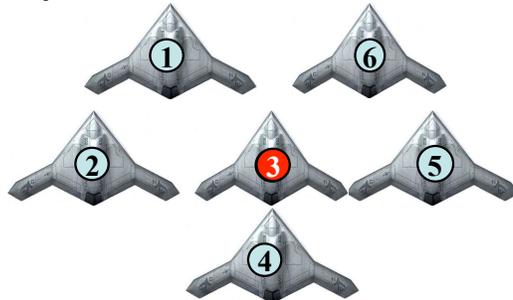
CALIFORNIA INSTITUTE OF TECHNOLOGY



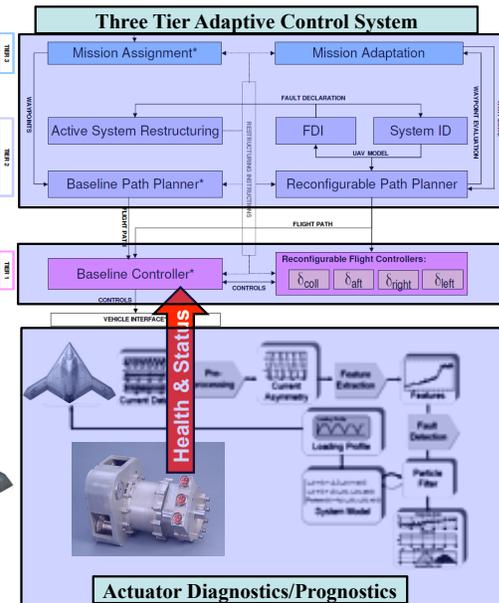
5. **Erv Baumann, Integrated Health Management Systems Architect, Advanced Programs & Technologies, Northrop Grumman Aerospace Systems**

In-Flight Detection & Mgmt of Degraded Actuation System

1. Flight group embarks with all systems operational.
2. Nearing the destination IHM detects degradation of control surface actuation and maneuverability in one of the key reconnaissance aircraft in the flight group.
3. IHM provides health & status diagnostic & prognostic information required by adaptive flight control system.
4. Control system reconfigures to maintain control and manage Remaining Useful Life (RUL) of the actuators.
5. Automated Mission Mgmt software automatically changes flight group formation and tactics to reduce vulnerability of degraded aircraft for remainder of mission.



CALIFORNIA INSTITUTE OF TECHNOLOGY



- Erv Baumann, Integrated Health Management Systems Architect, Advanced Programs & Technologies, Northrop Grumman Aerospace Systems**

Figure G-7: (top) Nominal flight formation; (bottom) flight formation after reconfiguration

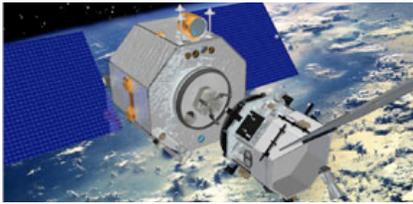
Erv Baumann uses the term *collective redundancy* to describe the idea that if you have multiple copies of physically separate units (e.g., a number of formation-flying UAVs or CubeSats), you can leverage that level of redundancy in a number of ways including the reconfiguration of flight group just discussed or the cannibalization example considered next.

In 2007, DARPA launched the Orbital Express mission to demonstrate autonomous on-orbit spacecraft repair (Figure G-8) (Whelan, Adler, Wilson, and Roesler, 2000). Several demonstrations were conducted between the two satellites. The ASTRO servicing satellite successfully replaced both the battery and flight computer in the NEXTSat prototype serviceable satellite. This illustrates the idea of collective redundancy and cannibalization of one spacecraft to repair another.



DARPA Orbital Express



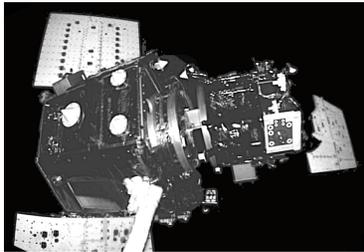


Orbital Express Mission:

- Two spacecraft:
 1. ASTRO, servicing satellite
 2. NEXTSat, prototype next generation serviceable satellite
- **Autonomous demonstrations:**
 - **Rendezvous**
 - **On-orbit refueling**
 - **Battery and Flight Computer replacement**

Orbit Express was a demonstration of autonomous satellite servicing in orbit

- Launched March 8, 2007
- Managed by: DARPA and NASA Marshal FligSpace Center



ASTRO and NextSat, taken from the robotic arm during the 4 km separation demon conducted June 22, 2007

Orbit Express Ref: http://en.wikipedia.org/wiki/Orbital_Express
<http://archive.darpa.mil/orbitalexpress/index.html>

Figure G-8: DARPA Orbital Express Mission On-Orbit Autonomous Repair

These two examples are special cases of what is considered to be ‘outside the skin’ (of the aircraft or spacecraft) redundancy. It generalizes the concept of redundancy to include any resources that are within reach via some conveyance or transport mechanism (e.g., a spacecraft flying over to a disabled spacecraft like Orbital Express or a lunar rover with a critical component being commanded to drive to a disabled lunar base so it can be cannibalized to save lives, etc.). This type of redundancy we have called collective redundancy.

Summary

Currently, we build resilience directly into our flight software by using conventional techniques such as exceptions, timeouts, and other low-level programming mechanisms to intercept errors caused by either system design flaws or changes in environmental conditions that our system was not suitably designed to tolerate. Unfortunately, this approach is not good for handling changing objectives and tracing to the root cause of a problem, and makes it hard to retrofit a legacy system since many of these techniques are pervasive and obfuscate the true intended functionality. The other method commonly used today to ensure resilience is human oversight. Operators, system engineers, and scientists, etc. will keep global oversight and provide intelligent responses to assure safe operation. Unfortunately, this approach is costly, can be error-prone, and often does not scale well. Even worse, in deep space scenarios, timely human interaction

can be problematic. An increasingly interesting approach to making a flight system more resilient is providing the capability for its flight software to self-adapt at run time to handle such things as system resource variability, environmental changes forcing nominal system functionality to change, and system faults.

The topic of self-adaptive software systems has been studied in a variety of application areas, including autonomic computing, robotics, control systems, programming languages, software architectures, fault-tolerant computing, and biological computing, but not yet in the planetary spacecraft domain. The capabilities subgroup realized this but did not have expertise in this area to perform a detailed study. We, however, considered three research self-adaptive software systems approaches that are promising for the realization of resilient systems. Rainbow is a self-adaptive software system from Carnegie Mellon University (CMU) that uses an abstract architectural model to monitor an executing system's run-time properties. A language called Stitch is used to configure Rainbow and provides a way to express constraints and strategies for adapting the run-time system to prevent violations of the constraints with respect to the model. It performs various levels of adaptations on the running systems. A nice discussion of Rainbow is presented in Garlan, Cheng, Huang, Schmerl, and Steenkiste (2004)..

During Workshop #1, several suggestions implied looking at bio-inspired software systems as a good way to provide resilience. Two interesting self-adaptive demonstrations from Michigan State University were discovered. An approach using digital evolution (leveraging the [AVIDA](#) digital evolution platform (Ofria and Wilke, 2004)) for evolving population of digital organisms (synthesized UML state behaviors as a set of interacting objects) subject to natural selection, where organisms are rewarded for generating state diagrams that support key scenarios and satisfy critical properties as specified by the developer. The SPIN model checker (Holzmann, 1997) and other tools were used to evaluate the state behaviors correctness or merit for the natural selection. The approach was successfully demonstrated to control autonomous navigation of a robot within a changing environment. A complete description is presented in Goldsby, Cheng, McKinley, Knoester, and Ofria (2008). The other interesting notion was that of an emulated digital enzyme as a communication mechanism—connecting parallelized reactive robotic control agents was demonstrated to explore the potential for evolving simulated controllers for the foraging problem. Properties of each robot and stimuli present in the environment are encoded in a digital format (molecule) capable of being manipulated and altered through programs (enzymes) executing in parallel inside each controller to produce the robot's foraging behavior. Evaluation of this design evolved strategies demonstrating various nature behaviors including the concept of using a primitive language for communications. The discussion of the project is presented in Byers, Cheng, and McKinley (2011). Both these methods leverage on having massively redundant and flexible elements at their core.

We considered redundancy in a different way than it is currently used within spacecraft today—the so-called 'outside the skin' scenarios discussed above. We deploy static or dynamic redundancy routinely today where subsystems within spacecraft are replicated. The distinction between static and dynamic redundancy is somewhat fuzzy, however, whatever term is used, redundancy of subsystems today is limited to a small number (such as 2 to 5) identical or similar units. To utilize redundancy today, some sort of logical switching criteria or continuous Triple Modular Redundancy (TMR) voting scheme is used. We speculated that resilient systems of the future require a new type of redundancy called collective redundancy where: (1) systolic arrays of interacting spacecraft (either connected or in formation flight) are capable of morphing configuration to preserve an acceptable level of functionality, (2) spacecraft are capable of cannibalization of units to effect repair, replacing degraded or nonfunctional units from one spacecraft with units from another spacecraft autonomously. The kind of redundancy needed for resilience is not yet realized.

We can gain additional effective redundancy by developing designs in which components are repurposed. For instance, science instruments/sensors might be pressed into service for health monitoring. Think, for example, of how valuable it might have been and how much time might have been saved if the Galileo imaging system could have turned and taken a look directly at the partially deployed antenna, versus

having to deduce its configuration from other data. This capability of reconfiguration to repurpose components is a promising enabler of graceful degradation potentially preserving nominal functionality. For example, imagine the real-life DS-1 scenario of a star tracker failing and repurposing a high-resolution science imager in use as a replacement star tracker (Rayman and Varghese, 2001).

We conclude the Capability Focus Group study by realizing that reliability, redundancy, reconfigurability, and recovery are the four elements necessary for resilience capabilities. Multicore processor devices are redundant and configurable; reliability and recovery within these devices is currently being demonstrated. Fail-operational modes have been demonstrated successfully in commercial-grade multicore processors. Software is reliable if built using strict constraints, however, architectural problems exist in sophisticated autonomy and adaptive software systems to assure safety-critical, high-reliability operation. Process does not scale well yet for systems or software. Autonomy and fault protection architectures are evolving in other domains (e.g., military aviation and space commercial applications) faster than NASA planetary or deep space missions.

The Capabilities Focus Group basically gathered information on a wide variety of technical capabilities, such as fault detection/management architectures, autonomy (planners), standards for software process, architecture and middleware, etc. A high-level survey of autonomy history and AI planners was captured and discussed. A list of these capabilities was generated and presented at Workshop #2, with the goal of integrating and generating more general capabilities that would meet common reference mission requirements presented in Workshop #2. The Section 4.3 Roadmap for Technical Development presents the ultimate outcome of this exercise as applied to commonalities of our three reference missions.