# A Theory of Vehicle Management Systems

Michael D. Watson
EV43 Advanced Sensors and System Health Management Branch
Marshall Space Flight Center
National Aeronautics and Space Administration
Huntsville, Alabama 35812
256-544-3186
michael.d.watson@nasa.gov

Stephen B. Johnson
EV43 Advanced Sensors and System Health Management Branch
Marshall Space Flight Center
National Aeronautics and Space Administration
Huntsville, Alabama 35812
719-487-9833
stephen.b.johnson@nasa.gov

*Abstract*—With the increasing capability of computers, engineers have designed vehicles to perform ever more complex tasks. Whether fully automated, as with robotic space probes, or partially automated in conjunction with a crew, vehicles have become both more complex and more capable. To manage this complexity, designers have developed increasingly sophisticated vehicle management systems (VMS) to manage vehicle internal states, and to operate in its external environment. While often effective, design of VMSs has often been on an ad hoc basis. Using insights from information theory, complexity theory, and artificial intelligence, this paper develops a theoretical framework in which to understand the nature of VMSs. The theory defines the interaction of VMS functions and provides a mathematical formulation to assess the complexity of different VMS configurations.[1][2]

### TABLE OF CONTENTS



## INTRODUCTION

Vehicle Management Systems (VMS) have become increasingly important over time in space missions, due both to the demands for increased flexibility and capability of these missions, and the supply of increasingly capable computing systems to provide this improved functionality. VMSs include the management of uncertainties in vehicle state, which is the vehicle portion of System Health Management (SHM), and the management and control of vehicle components to achieve external goals, which we will term "System Operations Management" (SOM). SHM and SOM functions can be allocated to humans or machines, whether on the ground or on-board. To the extent these are allocated to the vehicle's machines (as opposed to crew), these are part of the Vehicle Management System. The increasing complexity of the tasks that space systems are asked to accomplish, and the software and operational procedures necessary to accomplish them, have made VMSs a necessity for exploration missions. This paper investigates the underlying needs and functionality of Vehicle Management Systems, so as to better understand, and ultimately to better design them.

To do this, we shall draw upon ideas from information theory and system health management theory. Since VMSs necessarily use information to manage complex systems, information theory provide important insights. System Health Management theory has evolved to handle internal uncertainties, and we extend its ideas to deal with external uncertainties with Systems Operations Management.



[2] IEEEAC paper #1382, Version 4, Updated 16 January 2007.

## Flexibility to Manage Uncertainty

Flexibility is required to support many spaceflight missions whether scientific, human exploration, or military. Scientific missions value flexibility due to the uncertainties of the targets of their investigations. If the target of investigation was fully understood, then there would be no reason for scientific studies. Given the unknowns about the object(s) being investigated, the scientific spacecraft investigating these objects need the ability to adjust their operational plans and modes to uncertainties and to any discoveries made. Human flight missions often require flexibility to handle changes in their technical environments (such as problems in integrating two systems in an International Space Station build sequence), or to the vagaries of human behavior in system operation. Military missions require the flexibility to take advantage of targets of opportunity or unanticipated enemy behaviors.

Since the late 1950s and early 1960s, when most missions featured pre-set and unchangeable operational sequences, designers have built increasing flexibility into spacecraft. Making this possible are advances in computing and communications technologies, which allow for re-programming of the spacecraft's computers to accomplish new tasks or to change the order or parameters of existing tasks. As an example, the Magellan spacecraft, launched in 1989, featured a dual redundant system that included 128 kilobytes of command and data handling memory, 32 kilobytes of attitude control memory, and a 1 kilobyte read-only memory. While seemingly very limited by today's standards, changes to parameters and flight software were common and necessary to enable the mission. Current spacecraft often have memories on the order of tens megabytes, and NASA's Constellation program anticipates gigabytes of memory for data storage.

While these increased capabilities provide greatly improved operational flexibility, and hence maximize a system's inherent utility, it comes at the price of (among other things) complexity. The software required to achieve changing goals and manage its own internal complexity is much larger than in the past, with greater diversity of tasks, all of which increase the probability of internal software faults and increasing the flight software's costs. Despite these drawbacks, the ability to maximize a system's capabilities has been an irresistible attraction, and system complexity has increased apace.

Vehicle management systems exist to manage the resulting complexity. We define vehicle management systems as "onboard systems that manage a vehicle's internal capabilities to accomplish the mission goals." This encompasses several functions. Management of internal states requires accurate assessment and tracking of the vehicle's internal hardware and software states. It also requires mechanisms to implement state changes and functions to accomplish mission goals, whether through onboard sequence commands, ground operator commands, or crew commands (in the case of crewed systems). Finally, vehicle management systems must have mechanisms to assess and interact with the external environment. Generally (though not always), the purpose of a vehicle is "outward-directed", to move from one location to another, to sense the external environment, and to interact with that environment in a variety of ways. In cases like the International Space Station, the system is also "inward-directed," to interaction with on-board experiments and life support systems. The system must interact with these physically internal experiments in essentially the same manner as with the external environment.

Management of internal systems ultimately means assessing and managing the health of a vehicle's components, and then using these components to achieve some goal. Maintenance of internal health is a pre-requisite to achieving any goal. To do this, many vehicles have algorithms for fault detection, isolation and response (FDIR). Many also monitor the performance of components and subsystems to detect degradations, leading to crew or ground operator actions to replace or switch out components, modify future system actions and modes to minimize further degradation, prior to any full-fledged failure. Prediction of future failure, whether by humans or machines, is "prognostics." All of these actions, whether flight or ground-based, are part of the vehicle's health management system. That portion which resides on board is a subset of the vehicle management system.

As vehicles have become more complicated, the number of internal states to be monitored, tracked, and controlled has greatly increased, which is one of the major reasons for the evolution of VMSs. The increase in number of states implies a corresponding increase in mechanisms to monitor the health of system components, which in turn are integrated by the VMS into an overall vehicle state. These can be sensors to track temperatures, pressures, currents, and such, or software algorithms to create and store software information that describes its internal states. In general, the ability to change vehicle states has been implemented through ground and crew commanding capabilities, and in cases where these cannot be assured, through autonomous FDIR. Responses in general, whether on-board or off-board, are managed through the VMS. By this, we mean that some responses to change vehicle states originate from the crew or ground operators, but are mediated through the VMS to actually change the on-board state.

External uncertainties are the other driver of VMSs. The Mars Exploration Rovers Spirit and Opportunity provide good examples of the kinds of external complexities that require sophisticated responses, particularly in navigating the system in its local environment (around rocks, steep slopes, etc.), and deciding which instruments are best used to interact with specific environmental features such as rocks or soil. Other systems also interact with their external environments, though often in simpler ways. In the case of

Mars rovers, or future Constellation missions with humans and machines interacting on the Moon, the external environment is incompletely characterized, and thus the actions of the vehicles doing the "exploring" cannot be fully pre-programmed. The incompleteness of our knowledge of the environment external to the vehicle requires sophisticated real time or near-real-time interactions between system (meaning crew and ground operators) and the environment to be explored or observed.

To the extent that the vehicle itself is required to autonomously interact with or move around in this semi-unknown environment, a vehicle management system is the mechanism for performing these actions. Now and for the foreseeable future, these activities are either not automated or semi-automated, with responses made automatically based on limited information input. A vehicle with no on-board intelligence whatsoever would be virtually impossible to operate, or at best extraordinarily slow. The movements of the Mars rovers are tracked in a few meters per day, so as to be sure that the machines do not put themselves into situations in which they can no longer perform their tasks. One of the major objectives for future systems is to increase system autonomy so that planetary rovers can autonomously move around various obstacles to a target location in terms of kilometers per day in rough terrain. This would greatly increase the scientific potential of such systems. To do it requires a vehicle management system capable of far more sophisticated behaviors than those currently available.

For a vehicle management system to be effective, the increased complexity of having a VMS must outweigh the complexity of the subsystems and components that it manages. To date, there are few effective measures of this complexity, but information theory provides some insights into how this might be assessed.

## APPLICATION OF INFORMATION THEORY

VMS is concerned with managing the total vehicle state in response to both external environment and internal system conditions. This requires the vehicle state to be represented within the VMS to form a basis of decisions to maintain human life and mission objectives. Interestingly, it also requires an estimate of external state, at least to the degree necessary for the vehicle to interact with it. Information theory provides the basic definitions to determine the information necessary to represent the total vehicle and external state. The goal is to reduce the uncertainty (entropy) of this state estimate to provide for efficient and accurate solutions. For the purposes of this analysis, we will focus only on the vehicle state, though the calculation of the external state would be similar.

For a vehicle, assume that V is the Vehicle State, where, V = $v_1, v_2, \ldots, v_n$. (n Vehicle States). The uncertainty of the Vehicle State, V, is represented by the entropy of the state information:

$$H(V) = -\sum_{\mathrm{n}} \mathrm{p}(\mathrm{v_n})\log_2 \mathrm{p}(\mathrm{v_n}),$$

where $p(v_n)$ is the probability that the vehicle is in state $v_n$). [3]

Assume vehicle states are distinguishable, and so can be represented uniquely. Then, the absolute minimum number of bits (b) to represent V is b = H(V), the information entropy of the vehicle states. To keep the vehicle state representation manageable, the number of bits is desired to be minimal. [3]

The information entropy increases with the uncertainty of state, the information provided with each state variable. If all states have equal probability, then the entropy is a maximum when the uncertainty of the state is a maximum. [1] As the state becomes known, then uncertainty and entropy reduces.

Entropy of the total vehicle state can be found by calculating the entropy of the individual vehicle systems. For a vehicle with n total states, and a uniform distribution of these states, the entropy can be found as a summation of the individual subsystem entropies (with $b_i$ = an individual state) weighted by the probability of a given state value, $\left(\sum_{i=1}^{k} \frac{b_i}{n} H(\frac{1}{b_i},\ldots,\frac{1}{b_i})_{b_i}\right)$, plus the vehicle entropy calculated over the probability of each subsystem state $\left(H(\frac{b_i}{n},\ldots,\frac{b_k}{n})_k\right)$:

$$H(V)_n = H(\frac{b_i}{n},\ldots,\frac{b_k}{n})_k + \sum_{i=1}^{k} \frac{b_i}{n} H(\frac{1}{b_i},\ldots,\frac{1}{b_i})_{b_i} \ . \quad [1]$$

Thus, the vehicle entropy can be directly calculated from the subsystem entropies. As knowledge of the subsystems increases, the uncertainty and the associated entropy decrease. This simple model assumes that the subsystems are defined independently, which is not true for a space vehicle. The relationships between the subsystems influence other subsystem states such as thermal conditions or electrical power. Standard design practice for vehicles treats subsystems as independent with relationships at defined interfaces. Therefore, an assumption of independence can be maintained with an interaction term added to the summation as to correct for the subsystem interactions. This interaction term requires definition to determine the magnitude of effect on the entropy, but is assumed small for this theoretical treatment. We also assume uniformity of probability distributions for this first-order analytical treatment, but further work should assess the impact of non-uniform distributions.

Information theory also provides insight into distributed versus centralized VMS architectures. The Data Processing Inequality [3] indicates that communication paths and processing nodes should be minimized and is given by:

$$H(V) \geq H(V:C) \geq H(V:Y)$$

where C := State after communication
Y := Final output state after processing

This forms a Markov Chain which shows the information content is reduced (uncertainty increases) with each communication and processing path. This is due to uncertainty being added due to noise, interference, and error (environment induced, coding errors, electrical failures, etc). Thus the information initially provided does not include any information on these uncertain states and the knowledge of the subsystem is reduced by the increased uncertainty due to the increased communication and processing of the data. This must be balanced with considerations of information density and information processing time. Information density is the amount of information required to represent the complete state of the vehicle. The amount of data can be quite large for complex systems and grows as the complexity of the vehicle systems increases.

Lost information is in the form of bit flips or lost bits (b = e ≠ 0,1) due to noise. The Error Entropy H($p_e$) can be calculated based on the probability of error occurrence. [3] The Error Entropy grows with the Vehicle State space:

$$H(p_e) + p_e \log(|V| - 1) \geq H(V \mid Y)$$

The entropy for the VMS System Management Loop can be calculated as:

$$H(S) \geq H(S:C_2) \geq H(S:P) \geq H(S:D) \geq H(S:\mathrm{Pr}) \geq H(S:C_1) \geq H(S:M)$$

where S is the System state
M is the measured state
P is the calculated performance state
D is the calculated diagnostic state
Pr is the calculated prognostic state
C1 is the communicated state from measurement to performance
C2 is the final communicated state

This assumes that each state is not conditioning but is determining unique portions of the state information. No communication-induced information entropy has been accounted for between performance, diagnostics, and prognostics (i.e. no entropy between algorithms on the same machine). The inequality above shows that if one ignores the added uncertainties due to added communication paths and data, the certainty of the knowledge about the system state improves, and hence the system entropy decreases with each processing step, as each processing step is intended to collate and make consistent data about the system state.

Information theory provides guidance on the partitioning of the management functions. To minimize the entropy of the vehicle state requires a minimum number of communication channels and processing nodes. However, these goals are not always compatible. A centralized computing system minimizes the processing nodes, but may increase the uncertainty of the communication channels and increase entropy. This is due to the dependency of communication entropy on the distance of the communication. Free space (e.g. radio frequency) or guided (e.g. electrical wire or optical fiber) communication signal to noise ratios are dependent on communication distance as the power decreases (and hence noise and entropy increases) with distance. Thus, subsystems requiring long communication lines have higher entropy. At the other extreme is a totally distributed system where all processing is done at the lowest possible level.

When considering the number of interconnects between nodes, this may maximize entropy for both processing and communications. Processing entropy increases with the number of processors. Communications entropy increased with the number of communication lines. Distributed systems can reduce the communication distance of most individual inter-node connections, but necessarily increases the number of communication lines. Minimizing entropy requires a few processing nodes that minimize both the processing and the communications between nodes. Assuming that processing can be reduced by distributing the processing while minimizing the communications between nodes implies a central vehicle management computer connected to a few system control nodes, as this architecture should minimize communication entropy while keeping processing entropy low. If the number of distributed nodes is small, then the increase in communication lines is relatively small compared with the number of lines interconnecting the distributed nodes to the sensors/effectors. However, the communication line distances are substantially reduced such that the small increase in communication lines (which increases communications entropy) is more than compensated by the reduction in communication distances (which reduces entropy). Each distributed processing node calculates the state of a given subsystem. Thus, the vehicle computer can calculate the vehicle state by summing the state of each subsystem and including entropy for the communication and processing functions. For any given subsystem in this architecture, a two tier processing uncertainty is present: subsystem processor and vehicle processor. Communication uncertainty is also kept to two levels: sensor to subsystem processor and subsystem processor to vehicle processor.

Another factor that affects vehicle processor partitioning is information density. Information density affects processing latency as well as processing resources such as memory. As the density increases, the processing resources required to process the information to determine the current state increases. Assuming processing speeds are fixed, then increased information means longer processing latencies.

This tends to move systems toward distributed processing solutions to reduce the amount of information processed by an individual processing node and therefore allowing state computations to occur in parallel, hence reducing overall processing latency.

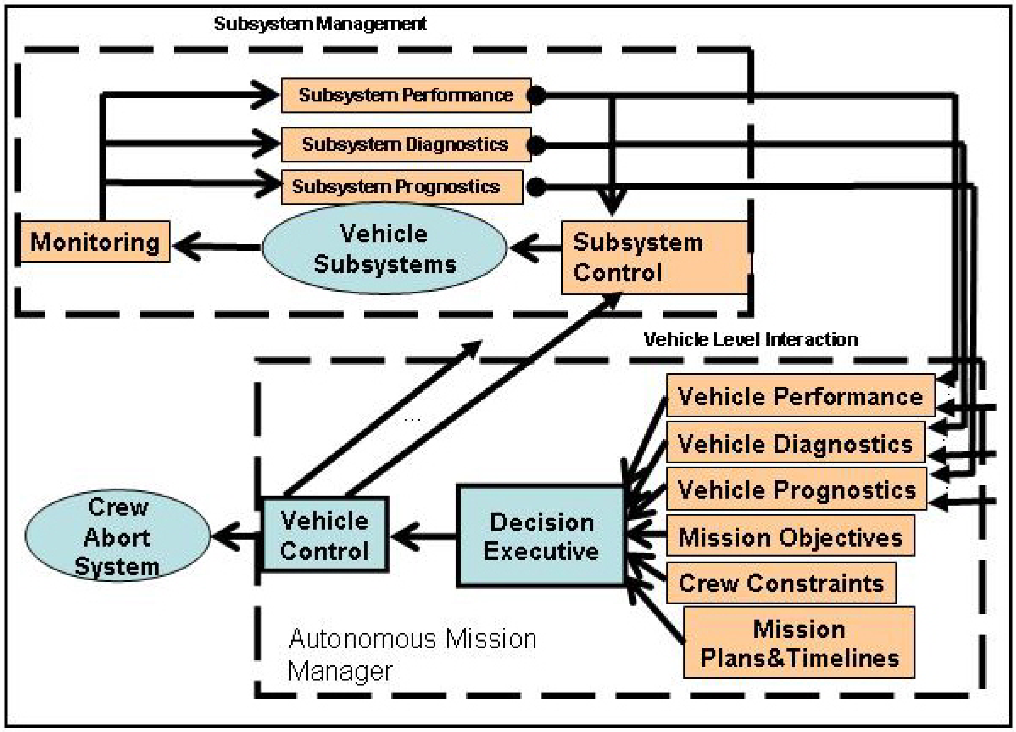
Communication latency also adds to overall processing latency. As with information density, implementing a distributed system also tends to lower processing latency. In a centralized system, long physical communication paths and short communication paths all converge on the centralized system, often forcing many communications onto single serial paths. Distribution allows communications to occur on parallel paths, reducing message path congestion and consequent slow-downs that increase communication latency.

Distributed processing and communication also tends to decrease signal path lengths, which in itself decreases information entropy in comparison to long signal paths. It is well-known in signal processing that bit error rates increase with path lengths, thus increasing entropy. To minimize the added noise, and hence the added uncertainties and potential system states (new states with erroneous bit-flips), distribution of processing nodes allows for processing nearer to the original signal sources, decreasing the number of errors, and hence entropy due to signal noise.

Finally distribution of processing may itself decrease entropy. Distribution means that each processing node processes less information (a lower information density). This implies (if designed properly) less complex processing capabilities, which introduce less entropy into the system than nodes that process higher information densities. Since complexity tends to increase in a non-linear and potentially exponential manner with information density, reduction of information density at any one processing node should decrease overall system complexity and hence entropy. Put another way, a distributed system should lower system entropy insofar as information density is never too large at a given processing site, as compared to a more centralized architecture with higher information densities per processing node.

A balance can be seen between processing entropy, information density, communication entropy, and communication latency. A system that balances the number of processing nodes with information density on each node provides low entropy and therefore an efficient processing system. Similarly, a system with minimal communication paths minimizes both communication entropy and latency. Figure 1 illustrates a conceptual VMS architecture based on this assessment.

**Figure 1: VMS Conceptual Architecture**




Intuitively, modestly distributed architectures appear to provide the most efficient calculation of the vehicle state for a complex spacecraft. This is because compared to a centralized system, the lower information density per processing node, combined with the decreased system latency and communication entropy through shorter path lengths, should outweigh the entropy increase due to the increased number of paths. A massively distributed processing system is likely to have a higher entropy due to the massively higher number of communication paths, whose collective entropy increase outweighs the progressively more modest improvements due to shortened communication paths and lower processing densities per node. Proof (or disproof) of this contention requires quantification of the various factors involved.

Information theory provides a framework in which to assess different VMS architectures in terms of physical attributes. These include the length of communication links, the number of sensing and computing nodes, the amount of processing performed at each node, and also the number of algorithms required at each node. This does not exhaust the design space. We must also discuss the functions that must be performed in each node, which then drive the requirements for the physical properties that information theory allows us to assess. The next section assesses these functions.

## VMS FUNCTIONAL ARCHITECTURE

Vehicle Management Systems, as described in the first section of this paper, manage the internal health of the vehicle so as to ensure the vehicle components can perform properly, and the use of the vehicle components to perform some goal with respect to its external environment. We use the term "external" in the generalized sense of interaction with things external to the system itself, which could include experiments physically internal to the system, but functionally exterior, such as on the International Space Station. Crewmembers are, in this sense, "external" to the vehicle, though they are "internal" to the mission. We will consider crewmembers to be external to the vehicle management system, since by definition they are not part of the vehicle, but rather interface with the vehicle, though in a more direct manner than ground operators, for example.

Vehicle autonomy simply means the internalization of functions necessary to achieve mission goals, where the vehicle is part of a larger system (including crewmembers and ground operators) that is intended to achieve a mission. From a functional view, it is important to understand the functions necessary to achieve mission goals. It is then a system design trade based on cost, technology maturity, schedule, and performance factors to determine which goals are best achieved through allocation of functions to the vehicle as opposed to crewmembers, ground operators, or other machines outside of the vehicle in question.

As more functions are allocated to the vehicle (making it more autonomous), then the vehicle entropy will increase along with the complexity, which is represented by the increase in the number of possible vehicle states. It is important to recognize that the total *system* entropy may be unchanged in this allocation of functions to the vehicle, because the system entropy includes the entropy of the crewmembers and the ground operators! The fact that we allocate functions to the vehicle merely makes more of this actual total entropy (number of possible system states) visible, since calculation of "human-produced" system states has never been attempted or even considered, to the authors' knowledge. Put another way, if the system includes humans, the humans contribute to the total number of possible system states, as well as to the uncertainties about those states.

System health management (SHM) deals with the functions of monitoring and managing the health of complex human machine systems. A theory of system health management has been described elsewhere [2], and thus will not be described in depth here. However, some aspects of this theory are of relevance, in particular the description and relationship of the SHM's operational functions.

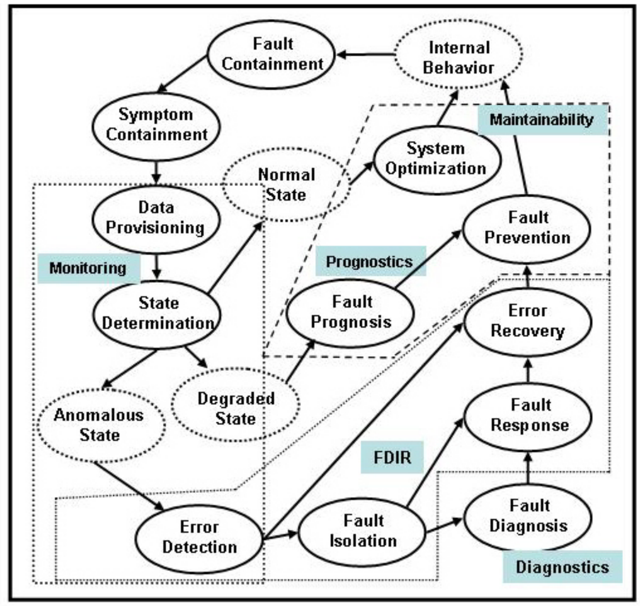



**Figure 2: System Health Management Operational Functions**

Essentially, during system operations, the health management system must contain faults and their symptoms, monitor system states, determine whether these states are normal or anomalous, isolate their location, determine their causes, respond to the fault, recover from the fault's effects and return to normal operation, and then prevent their future occurrence. The health management system also monitors degradations and predicts future problems so as to prevent them from occurring. The functions are connected a characteristic control loop, since these functions must occur continuously through the life of the system in performance of its mission. They can be allocated to machines or to humans.

In our context, SHM is one of the two major functions of a VMS. As defined above, the other is System Operations Management (SOM), which is the management and control of vehicle components to achieve external goals. This too occurs in a characteristic control loop, and for the same reasons as SHM functions---use of the vehicle to achieve mission goals is a continuous process that requires ongoing adjustment throughout the mission. Both SHM and SOM functions can be allocated to humans or machines, whether on the ground or on-board. When allocated to the vehicle's machines (as opposed to crew), these are part of the Vehicle Management System.

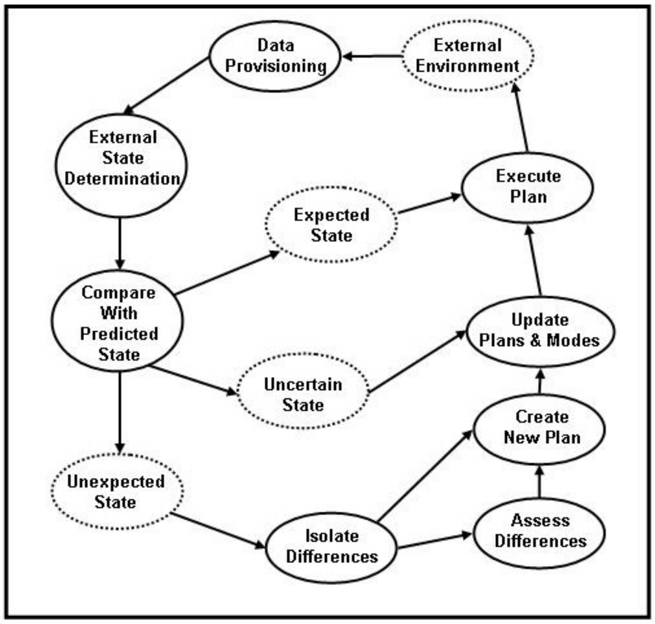



**Figure 3: System Operations Management Operational Functions**

SOM functions assume that vehicle components are healthy, or at least are being managed by other system functions. The SOM loop begins with acquisition of data about the external environment, converting that into a usable "external state" and then comparing that state with the system's expectations about that state. When the actual environment matches expectations, then the vehicle can perform as per its original plan. If the external environment does not match expectations, then the differences must be isolated and assessed so that a new plan can be created and then executed. There are also situations in which the information about the external environment is incomplete or uncertain, but may well have been planned. For example, the Voyager project understood quite well that the vehicle would capture information about the external environment (Jupiter, Saturn, Uranus, Neptune, etc.) that was simply unknown before. This was known ahead of time, and plans were created accordingly, but these had to be adjusted to account for the discovery of new moons, rings, and other targets of interest. Typically the comparisons of the external state with expectations about the external state, along with all of the resolution of differences and creation of new plans, have been done by mission operators, and in crewed missions, with the crew. Gathering of data and execution of plans typically are heavily automated for space missions, but could include human (ground or crew) activities. For crewed missions, the crew is typically involved in significant ways in the execution of plans, for the simple reason that the missions were planned from the start to require significant human involvement. However, automation of functions is often beneficial when it can be accomplished, leaving the crew to deal with those functions that require their participation.

Comparison of the figures for SHM and SOM show that the management of internal and external vehicle functions is very similar. The vehicle must acquire data about its internal and external environments, compare with expected or desired states, make decisions about how to respond to those differences, and based on the available resources, execute the appropriate actions. Humans or machines can perform these actions, but over time, the desire to lower long-term operational costs drives decisions to allocate more and more of these functions to machines, and hence to more capable and complex Vehicle Management Systems.

## CONCLUSION

Vehicle Management Systems are an important and growing facet of space systems, but have received relatively little theoretical attention. As with many other aspects of engineering, VMS's have been developed in practice, with the theory lagging. However, the complexity of these systems, and of the systems they manage, is beginning to tax "cut and try" methodologies. Space system designers and operators need a theoretical framework to cope with these increasingly sophisticated systems.

Both quantitative and qualitative approaches are appropriate to this theoretical task. The complexity of VMSs, which can be estimated by calculating the entropy of vehicle and external states, is a significant issue that needs to be addressed through appropriate architectural design, as well as more typical verification and validation approaches. This provides insight into appropriate physical decomposition of a VMS. Functional approaches help designers and operators properly decompose VMSs into logical classifications, which greatly aid architectural division, as well as operational decision-making, whether by humans or machines. This paper provides a starting point from which other engineers can expand these ideas to better understand and design these complex and critical systems.

## REFERENCES


[1] Information Entropy, Wikipedia, http://en.wikipedia.org/wiki/Information_entropy, accessed 6 December 2006.

[2] Johnson, Stephen B. "Introduction to Integrated System Health Engineering and Management," Proceedings of the First International Forum on Integrated System Health Engineering and Management, Napa, California, November 2005.

[3] Nielsen, Michael A., and Isaac L. Chuang, *Quantum Computation and Quantum Information*, Cambridge, UK: Cambridge University Press, 2000, pp. 500 - 525.


## BIOGRAPHY

**Dr. Michael D. Watson** was born in Lexington, Kentucky in 1964. He earned a BSEE from the University of Kentucky in 1987 and a MSE in Electrical and Computer

Engineering majoring in optics at the University of Alabama in Huntsville in 1996. He earned a PhD in Electrical Engineering at the University of Alabama in Huntsville in 2005. He is currently Chief of the ISHM and Sensors Branch at the NASA Marshall Space Flight Center where he has worked since 1989. Prior to this he served in the United States Air Force in 1988. Dr. Watson has been a member of International Society of Optical Engineers (SPIE).since 1995.

Dr. Stephen B. Johnson is a Health Management Systems Engineer for the Advanced Sensors and System Health Management Branch, NASA Marshall Space Flight Center, and an associate research professor with the National Institute for Science, Space, and Security Centers at the University of Colorado at Colorado Springs. He was formerly a faculty member in the University of North Dakota's Department of Space Studies, a researcher with the University of Cincinnati's Space Engineering Research Center, co-owner of Dependable Systems International, and researcher and control systems engineer with Martin Marietta and Northrop. He is the author of *The United States Air Force and the Culture of Innovation, 1945-1965* and *The Secret of Apollo: Systems Management in American and European Space Programs*, and numerous articles in system health management and the history of science and technology. His current research involves dependable space system design and operations, space industry management and economics, and the history of cognitive psychology and artificial intelligence. He received his BA in physics from Whitman College in 1981, and his PhD in 1997 in the History of Science and Technology from the University of Minnesota.