Editor: Richard Doyle
Jet Propulsion Lab
rdoyle@jpl.nasa.gov

Virtual Collaborations with the Real:

NASA's New Era in Space Exploration

Daniel E. Cooke, *Texas Tech University*
Butler P. Hine III, *NASA Ames Research Center*

All science is computer science, say recent claims.[1] Historically, scientific breakthroughs typically occur in the presence of a major breakthrough in a human-made technology. With Newton, it was the clock. With Maxwell and Einstein, it was the steam engine. Today, it is the computer.

People base their view of nature on a new device that is both revolutionary and pervasive. Because we humans built the device, we can comprehend it. Consequently, the new technology often further serves humans as a well-understood analog of nature. In the present context, some scientists claim that the basic particle of the universe is information; that information is not an abstraction of reality—it is, in fact, reality.[1] While some of these views are indeed extreme, they form the basis for current discussions within the scientific community.

If new technologies can inspire a new understanding of our universe, we might ask how far and in what direction can computer science research itself be pushed? NASA is poised to play a significant role in answering this question. Traditionally, NASA has pointed the way for many new technologies. NASA has identified the directions that have led to the breakthroughs in flight and space flight that we now often take for granted, playing a key leadership role that has galvanized large research, development, and industrial communities.

NASA will need to play a similar role in tomorrow's computer science research, developing critical enabling technologies to support future missions.[2] To play that role, NASA Ames Research Center has recently changed its research focus to computer science (see Figure 1).

Editor's Perspective

In this installment, we describe emerging success stories of applying AI techniques to the challenges of space exploration. It is time to take a look at how NASA is structuring its programmatic investments in the area of AI and related technologies.

The focus is the Intelligent Systems Program, managed by NASA Ames Research Center. As the authors describe, the inspiration for this program is the new set of capabilities that will be required for taking the next step in space exploration: Moving beyond gathering science results from the relatively well-defined environment of planetary orbits, to doing the same in the highly uncertain environment of planetary surfaces.

Joysticking from Earth is no longer an option. The space platforms defining space exploration's next phase must be more capable, and AI will play a central role in creating these autonomous agents of our desire to explore and understand.

—Richard Doyle

Motivations

At NASA, we are undergoing a fundamental shift in the way we design exploration missions. Driving the shift is a change in the character of the science goals for these missions.

Science exploration missions can be characterized in terms of the distance from the instrument making the observation to the observation's target. Science observations accomplished at relatively great distances from the target are called *remote science*; observations done in close proximity are *contact science*. The former observations typically occur either during fly-bys or from orbit, with the latter typically performed in situ, with the instrument in physical contact with the target (see Table 1).

Until recently, most science exploration missions beyond lunar orbit were remote science, given mainly to the global mapping nature of the science goals. As the science goals begin to require higher-resolution measurements and close proximity to the target, the missions increasingly involve more contact science.

Even in the presence of large communication time delays caused by the finite speed of light, remote science can often be accomplished by preprogrammed action. That's because the environment in which the spacecraft operates rarely requires decision-making more rapid than the round trip communications time. In this environment, the only requirement for rapid onboard decisions is during unusual or critical maneuvers (such as Saturn ring-crossing) or off-

1094-7167/02/$17.00 © 2002 IEEE

Table 1. Autonomy required for different mission classes. Blue indicates we can do this class of mission adequately with current technology, black means we can do this class of mission, but not very efficiently, while gray means we cannot yet do this class of mission with current technology.

Mission class	Example	Distance	Decision timescale	Level of autonomy
Fly-by	Voyager	Remote	Slow	Pre-event programmed
Survey	Galileo	Remote	Slow	Pre-event programmed
Local sampling	Viking	Static contact	Slow	Remotely operated
Local exploration	Pathfinder/MER	Dynamic contact	Medium	Remotely operated w/reflexes
Intensive exploration	MSL	Dynamic contact	Fast	Short-term goal-directed
Global exploration	Europa Ocean	Dynamic contact	Fast	Long-term goal-directed

nominal conditions such as internal system failures. Because the spacecraft environment is predictable and well-modeled, a control strategy involving at most conditional branching can serve to manage critical decision-making. As a last-resort means of handling unexpected conditions, the spacecraft can go into a "safe-mode" from which it can systematically recover under guidance from the ground.

Unlike remote science missions, we cannot accomplish contact science missions with preprogrammed actions. When a spacecraft or instrument is in situ, it is physically interacting with its environment, a situation that requires a very short decision timescale with respect to the round trip communications time. Also, the environment becomes difficult to predict or simulate prior to the mission, unlike a vacuum environment in which the spacecraft is subject only to relatively predictable forces and effects, such as a well-modeled gravitational field.

Under the dynamic conditions of in-situ environments, a traditional control strategy is difficult (and expensive) to use. The number of conditional decision points becomes exponentially high even for relatively simple missions. We will need a new way of designing exploration missions, incorporating higher levels of onboard autonomous decision-making to accomplish future science exploration goals.

Figure 1. In the shadows of the world's largest wind tunnel, the number one priority at NASA Ames is computer science research.

Currently, robotic contact science missions are remotely controlled—teleoperated. Literally hundreds of Earth-bound engineering and science specialists provide the intellectual safety nets required for space exploration. This approach has succeeded due to trade-offs in the complexity and distance involved. During long-distance missions when the finite speed of light becomes a factor in communications, the subject missions have been compelled to remain comparatively simple. In these missions, the state of the art in embedded systems has sufficed for the autonomy required.

Based upon their relative orbits, round trip communications at the speed of light between Mars and Earth vary from six to 40 minutes. For a complex human or robotic mission to Mars, capabilities in teleoperation must increase significantly. Even if we placed astronauts at a space station at the Mars-Sun libration point, there is a 7.2-second round trip delay. (A libration point is any of five positions in the plane of a celestial system consisting of one massive body orbiting another at which the gravitational influences of the two bodies are approximately equal.) Currently, fine-grained predictive control can deal with time-delays of five seconds or so.[3,4] So, given a five-second delay, a human-machine predictive control system can effectively provide fine-grained control of a remote device. Operating beyond five seconds delay will require robust reflexive controls on the remote device itself. The device will need to manage its own movements reflexively.

Therefore, mission complexity and communication delays are the roots of NASA's motivations for advancing computer science research. NASA will need an understanding of causal relationships in the data acquired in real-time, a need for greater autonomy in our deployed systems, and revolutionary advances in the way humans and machines work as a system. NASA's Intelligent Systems Program is a national initiative, organized to respond to these needs. This article will attempt to provide a vision within which these elements converge and a better definition of the elements and the goal of each research category are realized.

The vision

In the next 50 years, space missions might deploy astronauts or mission controllers with "intelligent" machines to points near or on the surfaces of distant planets. The people involved will need to operate in a seamless relationship with also-deployed intelligent machines.

In deploying humans to the surfaces of distant planets, intelligent machines will assist the humans with exploration and mission operations. Machines will need to extend and magnify human physical and mental abilities. Among other duties, the machines accompanying the astronauts might need to serve the purpose currently served by earthbound mission operators—the people who remotely control missions.

In having humans occupy a space station near the distant planet targeted for exploration, humans would deploy and control intelligent machines on the planet's surface. The machines will need to give the humans a true sensory experience of actually being on the planet. These machines will extend and magnify human abilities by seemingly placing humans in remote environments. The experiences of the space station's human operators could be packaged and sent to Earth, giving earthbound scientists the same experience. These remotely deployed systems might provide sensory inputs directly to the nervous system of humans and intercept signals from humans as feedback controls.

Currently, the computing requirements to carry the data and intelligence on future missions combine with radiation effects and extremely low wattage environments to make safe, cost-effective long-distance missions unattainable. Revolutionary advances must occur in almost every area of fundamental computer science.

Levels of reasoning

We can determine the extent to which a system performs cognitive functions based on a framework of reasoning levels. In *The Math Gene*, Keith Devlin describes three types of reasoning found in living organisms. In *stimulus-response*, the most primitive form, an organism can process an external stimulus and determine an appropriate response. Some S-R activities are so primitive that they are viewed as reflex rather than reasoning. When you touch a hot stove, you instinctively pull away, with little or no consideration of the situation.

Other forms of S-R reasoning are not so reflexive. Consider the situation of facing an impending head-on collision in traffic (a stimulus). Given some ample period of available time prior to impact, in this situation you will most likely spend a few seconds considering the options for taking evasive action (the response) to determine the best possible response.

Stimulus-stimulus, a more sophisticated form of reasoning, occurs when an organism receives a stimulus and, in turn, produces a stimulus for another organism or some tool or machine. In the head-on collision situation, once you've determined the best option for evasive action, you will produce the stimuli to cause your vehicle to avoid the oncoming vehicle. Therefore, the response in the S-S reasoning is a stimulus to control the vehicle to avoid the collision. From an historical perspective, note the S-S reasoning required by humans when using tools in the agrarian and industrial ages. Inventing the tools and determining, for example, the role of the seasons in plant growth, requires a more sophisticated form of reasoning.

> To advance the state of the art in human-machine systems, we need advances in automated reasoning, human-centered computing, and intelligent data understanding.

One simple view of *offline reasoning*, the most sophisticated form of reasoning, is to envision humans as having a primitive brain that performs the S-R and S-S reasoning. This primitive brain deals with external stimuli and cannot originate thoughts that are not triggered by outside occurrences. Now envision a more sophisticated brain that spends its time monitoring the behavior of the primitive brain—reflecting on and analyzing situations.

In the head-on collision example, O-L reasoning might result in trying to determine how to avoid future head-on collisions. Using O-L reasoning, you might invent mechanisms on the road or in the vehicles that would reduce the possibility for head-on collisions. Perhaps these inventions come to mind as a delayed S-R function. Nonetheless, we can characterize the separate analysis and reflection leading to invention as O-L reasoning. Observation and sophisticated analysis leading to discovery and invention is O-L reasoning and represents man's creative ability.

The major successes in machine-based reasoning have occurred in S-R and S-S reasoning functions. Even systems capable of deciding effective workarounds in the face of system and subsystem failures are basically performing S-S reasoning. Furthermore, these successes typically arise in narrow and well-defined problem domains. O-L reasoning remains the exclusive province of humans.

To advance the state of the art in human-machine systems, we need advances in automated reasoning, human-centered computing, and intelligent data understanding. When we perform reasoning at any level, it is based on filtering data—observing a very small segment of the electromagnetic spectrum—and determining causal links in the data. Even when we recoil from the hot stove, we have quickly determined the relationship between our pain and the fact that it is being caused by our proximity to the stove.

Intelligent data understanding is key to our ability to construct future intelligent human-machine systems. Advances in automated reasoning that push the current boundaries of system autonomy are required so that machines can perform reflexive (S-R) activities robustly. The degree to which we can advance automated reasoning to fulfill other levels of reasoning (such as S-S and O-L) are key to the future success of space explorations.

Finally, the extent to which we can view the human and machine as a seamless system—where humans are free to do what they do best, such as O-L reasoning, and machines do what they do best—will also help determine how well we effectively explore distant places in space.

Automated reasoning

In the past, the success of semiautonomous system behavior, more often than not, has corresponded to how well system designers could predict situations the system

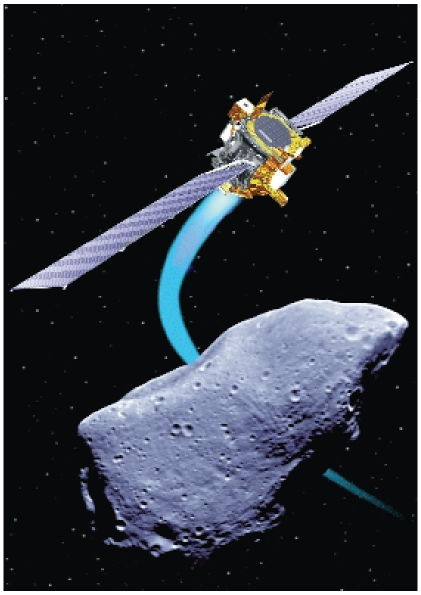

Figure 2. In 1999, the NASA Deep Space 1 Mission flew the Remote Agent Experiment, demonstrating the first use of Autonomy to control a spacecraft.

might encounter. If the situation occurs, the software provides a predetermined course of action. At a superficial level, we might view this approach as similar to raising a child. A parent might instruct the child in how to respond appropriately in some given situation. For instance, the child might learn that he or she should not strike a friend, even if the friend strikes first. The parent might attempt to predict a large number of varying situations and "program" the child with appropriate responses.

Raising a child this way is similar to the way semiautonomous systems have been programmed to operate on past missions. But with this approach, the resulting systems have significant difficulty contending with unforeseen events. Typically, the systems fail in these situations: They are unable to contend with this degree of uncertainty. This is a significant difficulty because exploration is fraught with uncertainties. How can anyone accurately predict all situations that might arise when engaged in exploration, particularly in-situ exploration? Furthermore, how can future missions vastly decrease their reliance on the earthbound safety nets represented by mission controllers?

Recent advances in model-based reasoning show great promise for dealing with the forms of uncertainty facing space exploration. The approach involves building systems that have a model of their environment. It resembles differing approaches to child-rearing that focus on raising children in a manner where they can operate using simpler guiding principles, such as, "treat others as you would like to be treated." These guiding principles provide for a more robust approach that can effectively contend with uncertainties. The system designer need not predict every circumstance that might arise. Instead, a modeling approach improves the system's ability to respond and adapt to uncertain situations. Through recent program development, NASA is addressing this important area of computer science research.

Intelligent data understanding

Currently, NASA receives two terabytes of data per day from Earth-observing satellites alone. NASA can acquire and store vast amounts of data, but the sheer amount is stressing our ability to analyze this data. We can view these vast data sets as empirical data. Scientists typically endeavor to reduce empirical observations to concise theories, which explain the observations. NASA's goals include revolutionary approaches that provide theory-based access to these data sets.

NASA's data are not always contained in a database. In fact, most data NASA acquires is contained in flat files that possess format information in their headers. Traditional approaches to data mining and knowledge discovery are, therefore, not always relevant to NASA's needs. A major result in this program element would be if we could reduce these datasets to much smaller representations of content of a more algorithmic nature. [We could view these algorithms as concise statements of the data—providing more manageable representations of the data that should lead to better understanding—and perhaps might be capable of reproducing the datasets.] Thus these algorithmic units might result in significant data compression.

The Santa Fe Institute is investigating the relationship between theories and the amount of data the theories explain. They are analyzing these relationships through an application of Kolmogorov's Complexity measure, called *algorithmic information content*. Given a particular message string, the programs that will print the string and then halt are identified. The length of the shortest program is called the string's AIC.

We can envision a ratio where the shortest program's size—the number of characters—serves as the numerator and the message's size—the number of characters in the program's output—serves as the denominator. For example, a program that computes millions of the digits of pi will result in a fraction close to zero. A message that is not the product of a formula or algorithm will simply be a print statement in which the entire message is a literal. In such cases, the ratio approaches one. One approach to data understanding might attempt to discover ways for analyzing data to identify the shortest program that can produce the data. These algorithmic units could then serve as the "theories" explaining the data and could result in data compression.

Fundamental results here should have wide application, providing new analytical tools to assist scientists in understanding space and Earth science data, and engineers in understanding vehicle and instrument maintenance data. Clearly, application to other types of data, such as Internet databases, is a potential side effect of research in this area. In terms of the vision we've discussed, intelligent data understanding is a crucial requirement that needs to be addressed for future space exploration. The ability to establish causal links in data is crucial—even at the S-R reasoning level.

On future missions, vehicle and personnel health and safety requirements will require the distillation and automatic analysis of large amounts of sensor data. Tomorrow's missions cannot rely on Earth-based controllers to perform data reduction and analysis. Furthermore, the astronauts will need to analyze and understand large amounts of scientific data as it is acquired during the mission. Quick analysis will let them perform just-in-time exploration, experimentation, and other scientific activities, based upon newly acquired scientific understanding.

Clearly, there is both a bandwidth and a time-delay problem. Given unlimited bandwidth in data transmission, we must still contend with the round trip time delays to Earth. Revolutionary advances to perform quick analysis and distillation to identify causal relationships in the data at its source are crucial to achieving the degree of autonomy needed on future missions that are both distant and complex.

Human-centered computing

Advances in automated reasoning will certainly affect NASA's ability to deploy robotic platforms into deeper space. These advances will also improve NASA's ability to deploy humans into long-distance exploration missions. As an example mission, consider the human exploration of Mars (see Figure 3). Because of the communication delays inherent in such a mission, astronauts and their mechanized physical and mental extensions will need to exercise greater autonomy.

The goals of human-centered computing research include system design approaches that take into account the level of intelligence and capability of the systems deployed, together with the cognitive and perceptual abilities of the astronauts. The result is optimal systems of humans and machines where the machines do what they do best, freeing humans to do the more creative activities that they do best.[6]

To further explore this revolutionary approach to systems design, consider past epochs of human experience.[7] In agrarian society, humans equaled physical labor. Because humans spent most of their time performing labor, they had very little time left to perform advanced problem solving, theory formulation, and the other more creative activities required for invention and discovery.

In industrial society, machines began performing physical labor and humans served as their brains. Machines extended and magnified the physical abilities of people. Humans were freer to perform advanced cognitive activities during this epoch, and science made great strides. In the information age, the human brain is extended and enhanced by a machine—the computer. Even trivial applications significantly extend human capabilities. Knowledge and the application of knowledge are embodied in software. For example, many people now prepare their taxes aided by software tools having much of a tax expert's skill and knowledge.

As system intelligence increases, computers can perform the more mundane and lower-level reasoning, freeing humans to perform more advanced and creative cognitive functions. In future exploration missions, humans cannot be mired in the details of mission operations or even vehicle health and maintenance. Humans must be free to perform a mission's discovery objectives. Humans excel at putting seemingly disjointed concepts together—the types of cognitive activities that are at the heart of invention and discovery. Computers do not excel at these kinds of activities, but do excel and outperform humans on more routine and sometimes tedious mental activities.

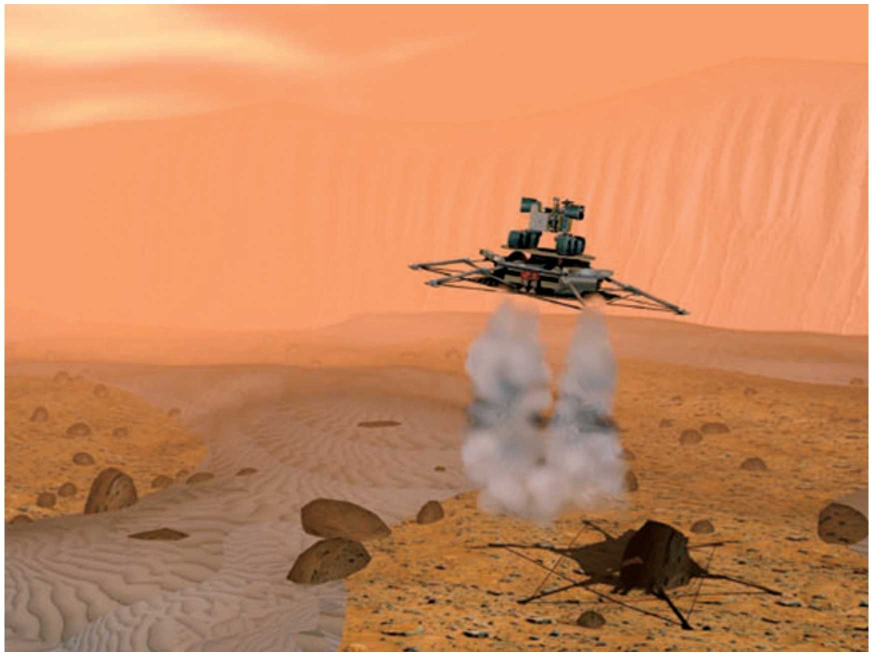

Figure 3. The Mars Smart Lander Mission will include an autonomous rover capable of traversing long distances with relatively infrequent command cycles from Earth.

Results in this area will affect not only human exploration of distant planets, but also the abilities of humans on Earth, performing such activities as mission operations and air traffic control. All NASA-relevant computer science research contributes to and converges under the human-centered computing research focus.

Nontraditional computing

Size, weight, energy consumption problems, and space hazards interfere with the ability to perform space-based computations. The possibilities of quantum and molecular computing provide answers to some of NASA's concerns about computing in space.

Offsetting the radiation and solar effects on computing is the massive parallelisms these approaches might offer. The size, weight, and power consumption concerns are also improved by these newer approaches to computing architectures. Perhaps the most important benefit is the new computational models and computer languages that these approaches might imply.

Revolutionary computing approaches differ radically from the traditional von Neumann and even the more conventional non-von Neumann approaches to architecture. As such, the computational models implied might provide radically new insights into problem solving—even possibly helping scientists find tractable solutions to problems for which only intractable algorithms are currently known. These algorithms might allow for feasible implementation within the constraints of current technologies.

More straightforward solutions to problems might result. (Currently solutions to these problems are approximate solutions—due to the intractability of the problems—making them much more complex to develop.) The revolutionary computing program element focuses not on building quantum or molecular computers, but on the computational models and languages implied by these approaches, as well as in the development of specific NASA-relevant algorithms that would allow for the immediate exploitation of these device technologies if and when they become available. (See the "An example" sidebar.)

An Example

NASA Ames Research Center has led in the development of a neural network-based Intelligent Flight Control system. News reports concerning the IFC software have shown a test pilot flying a plane with a simulated loss of a wing surface.

To simulate the loss of an entire wing, the test flight airplane can position an airfoil in front of the main wing surface. The airfoil carefully positioned creates turbulence that renders the main wing ineffective—as if it were completely removed from the fuselage. The plane goes out of control. When the IFC software is enabled, the pilot can regain control of the aircraft, even though it is "severely damaged." Clearly, the test pilot's skills are magnified and extended.

The IFC system is a good example of model-based reasoning—automated reasoning—insofar as the system is based on a model of flight. The system can integrate into a fly-by-wire aircraft and learn its flight characteristics through observation of pilot inputs and aircraft response. In doing so, the system exemplifies intelligent data understanding through its ability to establish and learn the causal links between inputs and aircraft response.

The system is also an example of human-centered computing because it magnifies and extends a human's ability to fly a seriously damaged aircraft. Extending the skills of a test pilot is one thing, but extending the skills and the pertinent mental acuity of a novice or nonpilot is a different matter altogether.

Several people—pilots and nonpilots alike—have flown a "full-up" simulation of an F-15 at NASA Ames Research Center (see Figure A). A full-up simulation is the type of simulator in which pilots are trained to fly new aircraft. The simulators are large, fitting only in a multistory bay, and provide realistic visual and motion effects. Furthermore, they are able to simulate varying effects on the plane's surfaces and indicate realistic aircraft responses to these effects. Of course one of the effects on the plane's surfaces are the pilot's inputs to the aircraft's control surfaces.

Figure A. Dryden Space Flight Center: F-15 modifications to test fly IFC software.

The subjects of the Ames simulation study were quickly taught and checked out on landing the aircraft under calm conditions from a good approach to San Francisco Airport (SFO). Most subjects could land the plane very well. Once checked out under normal flight conditions, the simulation is reset. The subjects are placed back on approach into SFO. Next, they experience a simulated failure of all control surfaces. The only operable control elements to the plane were the spoilers and the engines.

The plane is clearly out of control and efforts to regain control have no effect whatsoever. Finally, the IFC software is engaged. The controls are not as crisp as before. However, the subjects are typically able to regain control of the aircraft and perform hard landings at SFO. There would have been no injuries in these landings. The subjects' abilities were clearly extended and magnified by the IFC software: a good example of human-centered computing.

Organizing the community

The computer science community must face market forces, which could ultimately impede its ability to perform state-of-the-art research. To combat the potential for stagnation, the research community continues to need a major force to provide direction and leadership in key areas. In particular, due to the aforementioned market forces and in spite of excellent efforts to advance computer science research through programs funded by NASA and other agencies, the theoretical computer science community has, for the most part, lacked a significant and organized experimental community.

Without an experimental community, it is difficult to chart progress and provide convincing evidence of a theoretical result's significance. NASA seeks to advance the notion that hard application areas could be an excellent substitute for experiment. NASA has an excellent range of hard applications, and these applications converge with the applications needed in other agencies.

Just as experiments provide for the testing of theories in the physical sciences, hard applications can provide the experimental testbeds for the theories arising out of computer science. Therefore, in addition to providing funding for some of the most promising computer science research, NASA can help advance computer science research through its service as a pervasive and organized experimental community to test computer science results.

Software engineering research and practice

Basic research is most likely to result in prototypes and proofs of concepts. Prototypes can serve to perform preliminary tests against hard applications. However, the most promising results must be matured further so that the theoretical results embodied in software can be tested against more substantial applications and problems. The winning approaches must be matured—to a level of flight readiness or a similar level of production quality software.

Researchers are unlikely to produce near-production-quality software. Therefore, NASA is also determining better ways to transition the fundamental results produced by the research to products. In terms of software engineering practice, we believe that a different model of software engineering would help a great deal. A proof-of-concept arising from the theoretical community, tested against a hard application, will not necessarily be transitioned into practice.

To take an idea from proof-of-concept and actually use it on board an aircraft or a spacecraft is a major undertaking. To address this issue, some elements of software engineering need redefinition or refinement. Theorists are not likely to take their idea all the way to product. Software engineers will do that. The notion of *joint application development* should expand to include more than problem domain experts. This process should also include the theorists who developed the idea that is being taken to product.

This is not a new idea. Years ago, Richard Feynman, while working at Los Alamos, was dispatched to Oakridge, where engineers were building the plants to produce the materials for the atom bomb. The engineers needed to be briefed on the theoretical aspects and context within which they were working. After the briefing, the engineers could correct serious problems in their initial designs and ultimately construct the plants that served a major role in winning World War II.

Furthermore, efforts to identify formal classes of software based on their associated validation and verification requirements are needed. These classes should then serve as the basis for specialized process models and tools. The classes and their associated models will also be a research focus of the center.

Examples of classes:

- Ground-based information systems have more traditional verification and validation requirements and recommend well-known, existing software process models.
- Parallel systems require modeling beyond traditional verification and validation to discover anomalies due to concurrency, such as deadlock or race.
- Onboard flight systems require potentially all of the above plus extensive flight simulation and flight test.

Without process models and tools that are more sensitive to classes of software, the repeatable development of reliable software will continue to be difficult to achieve.

NASA is entering a new age of exploration. We are transitioning from science dominated by fly-by and orbital (remote science) measurements to in-situ or contact science measurements. We are transitioning from analyzing data primarily from single instruments to merging and extracting information from loosely coordinated fleets of spacecraft with multiple instruments. We are transitioning from centralized hierarchical control of spacecraft to mixed-initiative teams of humans and automation.

These transitions are driven as much by economics and policy as they are by science objectives. To accomplish these transitions and achieve the agency's mission goals, we need to deploy a new generation of technologies drawn from the computational sciences. With few exceptions, deploying these technologies will result in a revolutionary change in the way NASA designs and executes future missions, rather than an evolutionary change.

References

1. T. Siegfried, *The Bit and the Pendulum: From Quantum Computing to M Theory—The New Physics of Information*, John Wiley & Sons, New York, 2000.
2. D.E. Cooke and S. Hamilton, "New Directions for NASA Ames Research Center," *Computer*, vol. 33, no. 1, Jan. 2000, pp. 63–71.
3. Y. Yokokohji et al., "Bilateral Teleoperation: Towards Fine Manipulation with Large Time Delay," *Proc. 7th Int'l Symp. Experimental Robotics*, Lecture Notes in Computer Science 271, Springer Verlag, New York, 2000, pp. 11–20.
4. W. Cellary, "Knowledge and Software Engineering in the Global Information Society," Keynote Address, *Int'l Conf. Software Eng. and Knowledge Eng. 2000*, Knowledge Systems Inst., Skokie, Ill., 2000.
5. M. Gell-Mann, *The Quark and the Jaguar*, W.H. Freeman and Co., New York, 1995.
6. K. Ford, "Cognitive Prostheses," keynote address, *9th Int'l Conf. Tools with AI*, IEEE Press, Piscataway, N.J., 1997.
7. L.F. Penin et al., "Force Reflection for Time-Delayed Teleoperation of Space Robots," *Proc. IEEE ICRA 2000*, IEEE Press, Piscataway, N.J., 2000, pp. 3120–3125.

Daniel E. Cook is professor and chair of the Computer Science Department at Texas Tech University. He recently completed an 18-month assignment as the manager of NASA's Intelligent Systems Program, a cross-enterprise program led by the NASA Ames Research Center. He received a PhD in computer science from the University of Texas at Arlington. He currently serves as the Formal Methods area editor of the *International Journal of Software Engineering and Knowledge Engineering* and as the chair of the IEEE Computer Society Technical Committee on Computer Languages. He is a Senior Member of the IEEE. Contact him at the Computer Science Dept., Texas Tech Univ., Lubbock TX 79409; dcooke@coe.lttu.edu.

Butler P. Hine III is the manager of the Intelligent Systems Program, a NASA effort to develop intelligent spacecraft and vehicles, and technology to enable highly capable teams of humans and automation to solve some of NASA's most pressing problems. His research interests include instrumentation for space astronomy, image processing, machine vision, real-time high-performance computing architectures, telerobotics, and 3D visualization. He received BS in physics and mathematics from the University of Alabama and an MS and PhD in astronomy from the University of Texas, Austin. Contact him at bhine@mail.arc.nasa.gov.