Extending the Information Power Grid Throughout the Solar System

Al Globus, CSC at NASA Ames Research Center
September 2000

Abstract

The Information Power Grid (IPG) is intended to integrate a nationwide network of computers, databases, sensors, and instruments into a seamless whole that appears to be part of a user's personal computer. In other words, with an advanced IPG you could negotiate temporary remote access to all of the computing power, software, specialized instruments, and information that you might otherwise have to buy outright or be nearby to use. This paper discusses extending the IPG beyond our home planet and throughout the solar system. This can provide several advantages. First, the IPG can provide interfaces to current one-of-a-kind solar system exploration spacecraft to help provide a virtual solar system continually available to all. Second, the IPG may help reduce launch costs and failure rates. Third, IPG-like capabilities will be necessary to exploit solar system exploration by the thousands of automated spacecraft enabled by radical reductions in launch costs. Expansion throughout the solar system will require the IPG to handle low bandwidths, long latencies, and intermittent communications which are not requirements for the current Earth-bound IPG. These characteristics of deep space IPG nodes may be hidden from the rest of the IPG by Earth-bound proxies.

Introduction

The Information Power Grid (IPG www.ipg.nasa.gov) is an instance of a computation, collaboration, and data "Grid." Grids are being defined and developed by a substantial international community that believes that Grids represent the future of scientific and engineering computing, collaboration, and data management. The IPG is a major driver in this community. In this paper, we summarize the IPG and then examine the current state of and potential IPG contributions to launch vehicles, robotic satellites, and extraterrestrial landers and rovers with an emphasis on problems the IPG might help resolve and characteristics that impact IPG design. We then discuss some of the challenges that the IPG must overcome on our path to the stars.

IPG

NASA Ames and partners are developing the IPG. The goal of the IPG is to make a nationwide network of computers, databases, sensors, and instruments seem to be part of your desktop machine. A similar integration has been largely accomplished for the hundreds of computers at the NAS supercomputer center at NASA Ames Research Center, but the IPG is extending this integration beyond a single building and throughout the U.S. aerospace computational community. The IPG must deliver reliable high performance while hiding the continually changing resource base, and implementation and configuration details of a widely distributed computing, archiving, and sensing environment. There are four primary components to the IPG:

- **Applications --** programs that use IPG resources to do useful work. Many scientific and engineering applications could benefit from the uniformity provided by the IPG. Examples include aerospace design simulations, flight software testing, image analysis, database mining, telescopic asteroid search, etc.
- **Programming Tools --** software that makes it easier to develop IPG applications by hiding some of the complexities of the underlying structure. Toolkits provide flexible frameworks for creating applications from modules, some of which may reside on remote computers. Object oriented programming is a particularly powerful mechanism for hiding complex lower-level structure.
- **Services --** software providing uniform security, authentication, resource scheduling, resource access assurances, data management, fault management, and other facilities provided by conventional operating systems on single computers. Grid Common Services are the heart of a "virtual operating system" extending to all IPG physical resources.
- **Physical Resources --** computers, visualization environments, mass storage devices, instruments, and networks that provide IPG capabilities. Specialized software on each resource makes it IPG compatible. Current physical resources include seven SGI Origin 2000 supercomputers, a 100 terabyte mass storage system, a LINUX cluster, four Sun SparcStations, and a 250 workstation Condor pool. These resources are distributed over three NASA centers (Ames, Glenn, and Langley). Condor (www.cs.wisc.edu/condor) is a cycle-scavenging batch system which runs jobs on desktop workstations at night, weekends, or other times the workstations aren't in use.

Through the IPG project, NASA joins several other national organizations working to build the Grid, including the NCSA Alliance (www.ncsa.edu) led by the National Center for Supercomputing Applications and the National Partnership for Advanced Computational Infrastructure (www.npaci.edu) led by the San Diego Supercomputing Center, together with an international community working on best practice and standards for Grids - The Grid Forum (www.gridforum.org).

Globus and the IPG

The current prototype implementation of the IPG uses Globus (www.globus.org - no relationship to the author) for most Grid Common Services. Globus also provides programming tools tuned to traditional applications. Most of these applications are numerical (usually Fortran) programs using MPI, the de facto standard Message Passing Interface. A Grid oriented MPI has been implemented by the Globus toolkit developed by Argonne National Laboratory (www.anl.gov) and the University of Southern California (www.usc.edu). Globus provides

Resource Management	A uniform interface to local resource management tools, especially batch queuing systems, using an extensible resource specification language to communicate application requirements.
Security	Public key technology and X.509 certificate single sign-on, authenticated resource allocation, and process-to-process authentication.
Information Infrastructure	The state of grid components is published over the network.
Communication	Unicast and multicast message delivery services permit efficient implementation on a number of underlying communication protocols. For example, this is used to provide a Globus-enabled version of the Message Passing Interface (MPI).
Fault Tolerance	A heartbeat mechanism provides the ability to detect the failure of specific machines and processes.
Remote Data Access	URLs are used to access remote files. Read, write, and append modes are supported.

Legion and the IPG

While distributed Fortran applications are of great utility for simulation and analysis of spacecraft and launch vehicles, the object-oriented approach of Legion (legion.virginia.edu) developed by the University of Virginia (www.virginia.edu) may be more applicable for a solar system wide IPG because of the great importance of data archives and instrument control. In the IPG model, Legion is a programming environment sitting on top of the (mostly Globus) Grid Common Services. While Globus has a "bag of tools" architecture where applications and implementations can choose the set of tools they like, Legion has a more structured, object oriented approach which is intended to scale to millions of hosts and trillions of objects.

Legion strives to achieve ten design objectives:

1. Site autonomy
2. Extensible core
3. Scalable architecture
4. Easy-to-use, seamless, computational environment
5. High-performance via parallelism
6. Single, persistent, name space
7. Security for users and resource owners
8. Management and exploitation of resource heterogeneity
9. Multiple language support and interoperability
10. Fault tolerance

No single policy will satisfy every user so flexibility is necessary and desirable. The Legion architecture supports this philosophy with these characteristics:

- Each hardware or software resource is a Legion object. Legion objects are independent, active objects that communicate with one another via asynchronous method calls.
- Class objects are responsible for creating new instances, scheduling them for execution, activating them, and providing metadata about them.
- Users can define and build their own class objects; therefore, Legion programmers can choose and change the mechanisms that support their objects. Legion contains default implementations of a number of classes.

IPG Research

In addition to bringing systems such as Globus and Legion into a production environment, there are several computer science research initiatives the IPG is pursuing that will be relevant to a solar system wide IPG:

- Co-scheduling -- where several resources must be accessed simultaneously to accomplish a given task.
- Reservations -- where a resource is scheduled for a particular time. Current batch schedulers usually run jobs at the first opportunity.
- Network scheduling -- while CPU and memory resources are routinely scheduled in today's IPG, network resources are not. Since remote spacecraft must share resources such as the Deep Space Network without communicating directly between themselves, this capability is crucial.

This sort of scheduling and reservation is usually accomplished with substantial human intervention. This is perfectly adequate for the small numbers of spacecraft in operation today. It will not scale well to the thousands or tens of thousands needed to truly characterize the solar system in detail, much less to exploit the vast riches of outer space. Although today's IPG is focused on distributed supercomputing resources, Bill Johnston, NASA Ames' head of the IPG, has a broader vision: "I think what the Grid is fundamentally about is collaboration and sharing of resources. And I don't mean just computing resources, but also things like making major data archives and major instrument systems available to our collaborators around the world." NAS News March-April 1999, volume 4, number. 2 (www.nas.nasa.gov/Pubs/NASnews/1999/03/Johnston.html).

Given the above description of the IPG as a way to make first-class computing, communication, data, and technology resources available to all comers, what do we gain by extending the IPG infrastructure throughout the solar system? The short answer is that it's like extending the Internet to spacecraft and installations throughout the solar systems *and* making these first-class resources available to both people and machines. However, the high cost of Earth-to-orbit launch prohibits large-scale cost-effective solar system exploration. Fortunately, the IPG may be able to help reduce launch costs.

Launch

The key to robust exploration and development of the solar system is vastly improved launch systems. The space shuttle, the only existing reusable launch vehicle (RLV) and the most capable of all launch vehicles, has a *demonstrated* failure rate of ~1% and a cost of approximately $22,000/kg to orbit with a full load. Commercial launchers, all of which are expendable, carry a similar price tag and have a much higher failure rate, although good failure rate data are hard to find. Some commercial launchers, such as Pegasus and other small launchers, are significantly more expensive per kg than the space shuttle. However, the cost of a Russian Proton launch can be as low as $2600/kg [Wertz and Larson 1996]. This nearly meets NASA's 2010 cost goal of $2200/kg to orbit. By contrast, the commercial airline industry charges on the order of $10/kg per flight and has a failure rate of approximately 1 in 2 million.

Not only is the cost of access to space very high and failure prone, access has not improved very much over the last three decades. Indeed, measured in person-hours per ton to orbit, the 1960's era Saturn V was significantly less expensive than today's launchers [Wertz and Larson 1996] perhaps because large lift capacity tends to be cheaper per kilogram. The Saturn V lifted the SkyLab space station into orbit with a single launch, in stark contrast to the dozens of launches required to lift the International Space Station (ISS) by today's family of launchers. SkyLab had perhaps half the pressurized volume and a quarter the mass as the ISS will have at completion. This lack of, or even negative, progress in launch vehicles must be decisively reversed for the space program to move beyond a very small number of incredibly expensive missions. Launch vehicle improvement is **the** issue for space development.

Recent reviews of problems encountered by the space shuttle both before and during launch [SIAT 2000] discovered major opportunities for the application of information technology in general, and Grid capabilities in particular. In addition, a surprisingly large fraction of launch failures are directly attributable to information technology failures. For example, the destruction of the vehicle and payload in the second Sea Launch mission was apparently caused by a software error.

By 2020, NASA intends to achieve $220/kg to orbit with a 0.01% failure rate. This should be sufficient to support high-end space tourism, driving travel costs to much less than the current ~$20 million for a tourist ticket to the Mir space station. Launch vehicles often operate very near the physical limits of materials and components. To operate near these limits safely and efficiently requires an intimate knowledge of the current state and history of each vehicle and all support systems, which in turn requires first-class, integrated data systems. Recent independent reviews have indicated that trend data are difficult to extract from existing shuttle data systems, and that some data are missing or incomplete [SIAT 2000]. Also, three recent space failures (Sea Launch second flight, Mars Polar Lander, and Mars Climate Orbiter) were caused, in part, by software or information processing failures. There is no doubt that the personnel involved are dedicated and capable, but the data systems are not what they could be.

Potential IPG Contribution

A fully integrated RLV data system based on the IPG might substantially improve launch vehicle cost and safety. This data system would include all relevant data in human and machine readable digital databases, large computational capabilities, model based reasoning, wearable computers and augmented reality for technicians, a software agent architecture for continuous examination of the database, multiuser virtual reality optimized for launch decision support, and automated computationally intensive software testing.

- **Human and machine readable digital databases:** materials, components and system interactions must be fully characterized and understood to insure launch success and to analyze launch failures. Since most current launch vehicles were developed before the availability of large and inexpensive digital data storage, where these data exist at all they are often scattered among incompatible databases and in some cases are not even machine readable. Consider the experience of the NASA Space Shuttle Independent Assessment Team:

  "*In order to assess the root cause of wire damage, trends were needed that distinguished the type of wire damage (i.e., insulation damage, exposed conductors, conductor damage) as a function of location of occurrence in the Shuttle. ... To provide this information, a team of 10 engineers and 3 quality inspectors worked for 1 week reviewing Problem Reports in the Kennedy Space Center Problem Resolution and Corrective Action system. This intense effort was needed because the data contained in the system lacked standardization and fidelity and needed to be assessed and interpreted by the engineers in order to be meaningful. In some cases, the data did not exist or could not be resurrected*." [SAIC 2000] . Thus, a trend report that a first-class data system should have generated automatically in a few minutes required approximately three person-months to produce.

  The space shuttle Challenger was destroyed and seven astronauts killed in an explosion when O-rings in a Solid Rocket Booster failed. The O-ring material was temperature sensitive and launch took place at ~35 degrees F below any previous launch experiencing no O-ring damage. [Leveson 1995] reports that O-ring partial failures were increasing over time in the previous 24 flights but this trend was never extracted from the anomaly database. Of course, the O-ring problem was well known to the SRB engineers but their judgment was overridden. [Tufte 1997] provides evidence that better display of the temperature-failure relationship, which would have been easier with a better data system, might well have convinced management not to launch at such low temperatures.

  In a system such as the shuttle involving three major NASA operational centers plus contractors, a single centralized database would be nearly impossible to implement. Thus, a widely distributed set of archives that can be accessed securely by both people and software, the sort of archives envisioned by the IPG, is necessary. Since NASA personnel occasionally report difficulty getting copies of data developed under NASA contract, integration of data developed with government funds into an accessible (Webbed) distributed archive should perhaps be required by all NASA contracts. IPG can provide enabling infrastructure for this, although this not the major issue. The major issue is metadata standards and standardized data encodings, all packaged in self describing objects such as XML (www.xml.org) provides. If the IPG were adopted, however, it would likely spur standardization.

Even if new RLV efforts are not as widely distributed as the space shuttle project, IPG-like wide area capabilities are important. Crucial data on material and component aging is often gathered in other operational environments, such as commercial and military aircraft. Also, manufacturers are often reluctant to transfer their data to external archives. Thus, a complete digital database of all relevant data would need to include secure access to databases maintained by component and materials manufacturers and their customers. These are unlikely to be located at the launch site.

- **Large computational capabilities:** once all relevant data are machine accessible, numerous opportunities for computational analysis will present themselves. The large computational capabilities being integrated by the current IPG effort can be applied. Beyond the traditional applications, such as computational fluid dynamics and finite element analysis, automated analysis of failure modes and automated software testing, both discussed below, can benefit from supercomputing capabilities.
- **Model based reasoning:** model based reasoning (aka model based autonomy) is currently used for real-time control of launch vehicles. This technique involves modeling launch vehicles as a set of interacting finite state machines. Model based autonomy simplifies flight control software since launch vehicle engineers build a model of their system that is examined by non-mission specific software to diagnose system state and generate commands to the vehicle (see figure 1). Altairis Aerospace Corporation used model based autonomy for the Delta II cryogenic propellant loading sequence, various aspects of Delta IV, and the Conestoga launch and subsequent failure analysis. NASA Ames is developing X34 and X37 flight experiments using Livingstone, model based autonomy software with more sophisticated diagnostic capabilities. To generate the failure modes used by the models requires large computer resources to simulate the physical operation of components. Also, given sufficient computer resources, the finite state machine models developed for real-time control could be used for an extensive analysis of failure modes created by global interactions. Finally, in principle, model based reasoning and other artificial intelligence techniques could be extended to predict component failures in reusable launch vehicles and automatically signal maintenance personnel to replace or refurbish components before failure. These predictions will require very substantial computational resources which could be provided by an IPG.

Figure 1 shows a finite state machine representation of a valve with four states: open, closed, failedOpen and failedClosed:

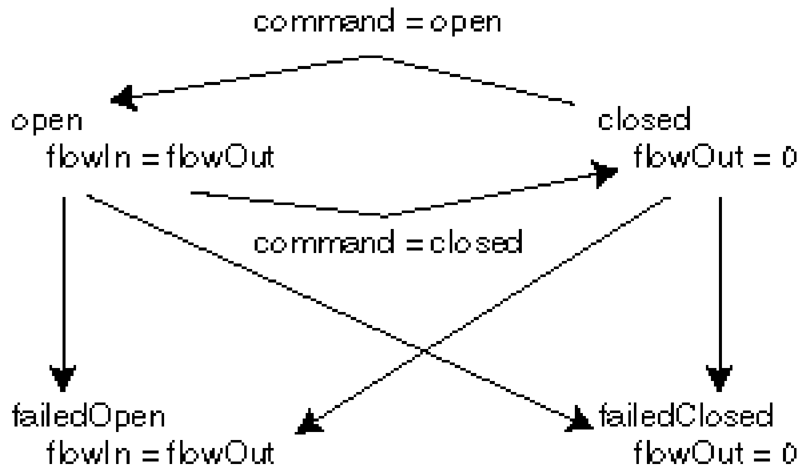

Figure 1: finite state machine representing a valve

- **Wearable computers and augmented reality for technicians:** wearable computers are physically integrated directly into a user's clothing. Augmented reality refers to see-through head mounted display technology that projects computer generated annotation onto the real world. Hands-on shuttle technicians are cut off from computing technology since loose items are prohibited, in order to avoid dropping things on fragile flight hardware. Even wedding rings must be taped to technician's fingers. Wearable computers would give technicians direct access to shuttle databases in the IPG environment. Shuttle processing also requires many inspection tasks to determine how shuttle hardware state has changed from pre-flight to post-flight. Augmented reality could be used to project preflight status directly onto the hardware being examined, allowing rapid and accurate determination of any changes. If a video camera were integrated into the augmented reality system, shuttle technicians could take pictures of questionable hardware for automatic analysis and comparison with pre-flight images by IPG computing resources.
- **A software agent architecture for continuous examination of the database:** once a machine accessible database of all of the data exists, then IPG computing resources can be used to automate inspection, trending, and analysis tasks using software agents. As the amount of computation that could be consumed by such database examination is essentially unbounded, one approach to gaining the necessary CPU hours without excessive hardware cost is to integrate all NASA desktop computers and workstations into a set of Condor pools integrated with IPG technology. Since there are many thousands of such computers, and each should be available approximately 17 hours per day based on University of Wisconsin experience, enormous computing resources could be applied to database examination.
- **Multiuser virtual reality optimized for launch decision support:** launching a rocket requires a series of critical Go/ NoGo decisions, some of which must be made very quickly and require substantial expertise. Such expertise may not always be available at the launch site. When making such decisions it is vital that all decision making participants have a common and accurate view of the current situation. Multiuser, widely distributed, immersive virtual reality technology such as that employed in the NASA Ames Virtual Windtunnel ([www.nas.nasa.gov/ Software/VWT](http://www.nas.nasa.gov/Software/VWT)) can place distributed decision makers in a common information space which, together with voice communication, can assist with critical time-dependent decisions based on

data provided by distributed IPG-integrated archives and sensors.

- **Automated computationally-intensive software testing:** software testing has emerged as a major problem in aerospace design and operations. Modern software has far too many states to be exhaustively tested. Developing testing schemes is extremely manpower intensive and error prone as the recent Mars Polar Lander loss demonstrated.

  One approach to automating software testing is random testing, although only certain aspects of a software system can be exercised. In this mode, software is subjected to randomly generated inputs. When the program crashes, the relevant data is collected and supplied to the developer. There is no practical limit the CPU hours that can be devoted to such testing. These tests are a good candidate for a NASA-wide set of IPG integrated Condor pools. This would turn every desktop PC and workstation at every NASA center into a software testing resource. Intel uses a similar scheme to test hardware designs. The Intel system runs circuit testing software on thousands of CAD workstations distributed around the globe when they are not otherwise engaged.

  The effective state space of a control program can be reduced a great deal using model based reasoning because the underlying finite state machines have far fewer states than an equivalent traditional program. Indeed, the state space of some subsystems is small enough for exhaustive testing. Thus, a second computationally intensive software testing task is automated state exploration for model based reasoning programs. Theorem proving requires large CPU resources and, especially, large memory spaces. Large single-address space parallel IPG supercomputers such as the SGI Origin 2000 (hundreds of gigabytes of physical memory) should be quite useful.

Solar System Exploration

In large part due to the incredibly high cost of launch, our solar system exploration program typically consists of a very small number (10s) of operational robotic exploration spacecraft at any given time. These are typically controlled by one-of-a-kind ground stations and a great deal of manual intervention. Attempts to reduce the number of ground controllers contributed to the recent loss of the Mars Polar Orbiter and serious problems in two other missions [MCO 2000], suggesting that automation requires a firm and rigorous foundation.

Model based autonomy has firm theoretical roots (see, for example, [Manna and Pnueli 1991]) and experiments in spacecraft control are progressing. The NASA Ames on-board Remote Agent Software, including the Livingstone diagnostic engine, controlled the Deep Space 1 spacecraft for approximately two days in 1999. While a thread bug in the executive software cut short the experiment, results suggest that autonomous spacecraft using this technique may be feasible. Also, Altair Aerospace Corporation's implementation of model based autonomy is due to be installed on the WIRE spacecraft for an engineering test in the near future. Autonomous spacecraft, which may occasionally require IPG resources to perform large calculations infeasible for onboard processors or access Earth-bound data, are a significant requirements driver for a solar system wide IPG. A second set of drivers are long latencies, low data rates, and intermittent communications.

Data retrieved from solar system exploration spacecraft are being placed on the World Wide Web. A particularly interesting example is the NASA Ames Lunar Prospector site (lunar.arc.nasa.gov). NASA intends to extend these beginnings into a "virtual solar system" where researchers and ordinary citizens can examine solar system data in an intuitive and easy to use manner over the Net. This vision requires integration of the spacecraft, landers and rovers gathering data, Web accessible data archives, and computational facilities for converting raw spacecraft data into four-dimensional (three spatial dimensions plus time) data-driven models of the solar system. This is a computationally intensive task and the IPG may be able to help.

IPG and Solar System Exploration

To get a feeling for the possibilities created by reduced launch cost, imagine a project to fully characterize near-Earth objects (NEOs), a project of some interest since these objects sometimes collide with Earth with catastrophic consequences [Lewis 1996]. There are believed to be about 900 NEOs with a diameter greater than 1 km, however [Rabinowitz 1997] estimates that there are approximately one billion ten-meter-diameter NEOs. Laboratory examination of meteorites and spectra from orbiting NEOs proves that these bodies are of extremely diverse composition [Nelson 1993]. To accurately sample such a large and diverse set of bodies would require the capture and return of tens of thousands of small objects [Globus 1999] and sample returns from thousands of larger objects. Current approaches to spacecraft control involving several ground control personnel per spacecraft will not scale. These spacecraft must be largely automated with the ability to use Earth-bound computers for complex trajectory, rendezvous, capture calculations, etc. Furthermore, routing all communications directly to Earth is probably impractical. This project contains all the problems encountered in a solar system wide exploration project with a slightly more tractable scope. We will use it as a model project for a solar system wide IPG. Figure 2 represents this project with relatively few exploration spacecraft:

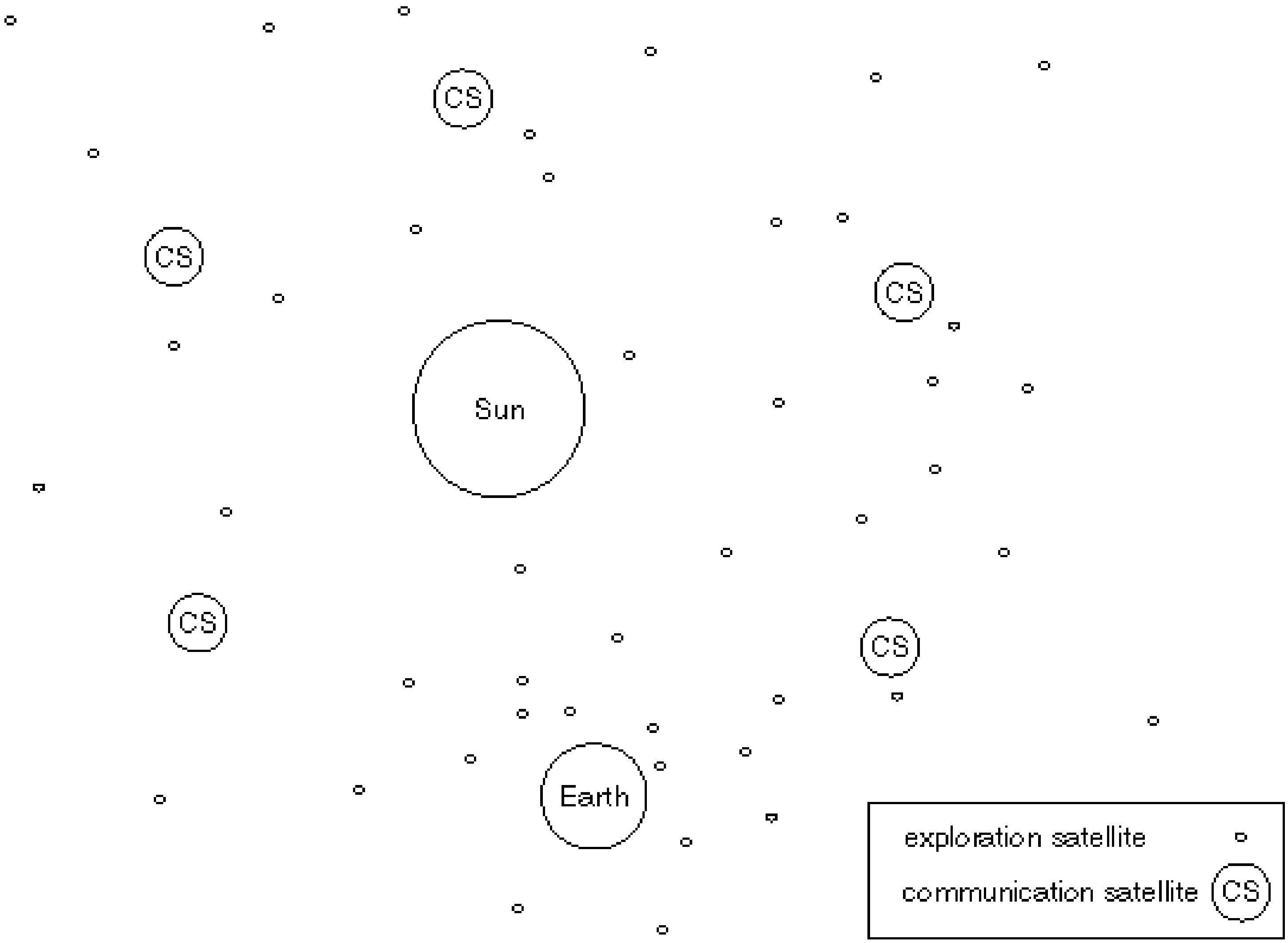

Figure 2: Satellite locations for NEO characterization

To minimize exploration spacecraft antenna size and provide communications to spacecraft on the other side of the Sun, we propose a number of communication satellites scattered along Earth's orbit. With a sufficiently large number of communication satellites, line-of-sight laser communication between them can provide communication with Earth and even between exploration satellites. Thus, the communication satellites form a massive extension of the Deep Space Network and perform a function similar to the Internet backbone. Scheduled reservations of the communication satellites are necessary to point their high-gain antennas at the exploration satellite needing communication at any given time. Reserved co-scheduling is necessary for communication satellites to point their high-gain antennas at each other to pass messages to and from Earth.

Each spacecraft, lander, and rover may be represented by a software object. Spacecraft, landers and rovers must be represented by terrestrial mirror objects (proxies) to hide latencies and to represent the vehicles when they are not in communication with Earth. These proxies must know the schedule of their remote reflections so that co-scheduling and reservations may be implemented properly. Once data have been safely stored in archives, normal Web access should be adequate for browsing, however more controlled access using IPG security will be necessary for applications that read the data, calculate more useful versions (e.g., mosaics of images), and insert the results back into the archive. Thus, the IPG becomes a set of (sometimes) access-restricted high-performance computing, data archive, and special instrument areas of the Web. This amounts to a democratization of high-end resources making them available to a much wider audience, reducing barriers to research and technology advancements, and increasing public support

It is reasonable to assume that the exploration spacecraft will be autonomous but require occasional large-scale processing because of the limited capacity of their on-board computers. Large-scale processing needs might include trajectory analysis, rendezvous plan generation, docking plan generation, surface hardness prediction for choosing sampling sites on larger asteroids, etc. After passing a request to Earth for large-scale processing, Earth bound high-performance CPU resources must be reserved to insure that the processing results are available the next time the requesting satellite is in communication. Most large-scale processing should be kept on Earth to take advantage of new developments in computer science and hardware production.

No matter how good the automation software, it is difficult to imagine that no human intervention will be necessary to operate a complex exploration spacecraft, at least in the near future. However, the finite state machines used for model based autonomy may include the unknown state. Spacecraft may be programmed to go into safe mode and contact Earth when important subsystems enter the unknown state. Indeed, this is normal behavior for Altairis models. This Earth contact could trigger a message to one of three Earth stations staffed by ground control experts. The Earth stations could be placed around the globe such that each facility is only open during normal working hours while still achieving 24-hour coverage. These Earth stations might effectively appear to spacecraft to be IPG nodes that act as proxies for human controllers.

Images and data captured by exploration spacecraft must be returned to Earth for additional processing and archival. Exploration spacecraft will spend large periods in transit followed by intense periods of data production during close encounters. Thus, network reservations on the communications satellite infrastructure are necessary as well as a mechanism to insure that archival space is available on Earth when the data arrive. For prompt processing into usable form, CPU reservations are also necessary. Thus communication, archival, and CPU co-scheduling is necessary.

Current IPG implementations assume nearly-continuous, high-bandwidth, low-latency communication. These assumptions are broken in a solar system wide IPG. Instead, large latencies, low data rates, and intermittent connectivity are typical of deep space communications. The Internet Domain Name system, which is an essential component in the operation of Grids and all other Internet applications, has an architecture that is intended to fail soft in the face of network partitioning: DNS is a partial state data manager capable of autonomous local operation in the face of network failures. This provides an experience base for attacking problems associated with a solar system wide IPG.

Solar System Challenges for the IPG

The key problems that the current IPG neither addresses nor is investigating are low bandwidth, intermittent communications, and long latencies. These problems may be addressed by adding proxies to the IPG architecture. The IPG currently uses proxies to deal with site security boundary protection systems such as firewalls. For interfacing with Earth-bound computers, proxies are located on the ground, hold information about the last known state of a satellite, and can make a best guess as to current spacecraft state. Communication between proxies and the rest of the IPG is simply terrestrial links and can hide the extremely low bandwidth between remote spacecraft and the ground. Proxies, being Earth bound computers, should be functional and accessible as much as any other Earth bound computer and thus may hide intermittent communications with remote spacecraft. Long latencies (>1000 seconds to Earth orbit on the other side of the Sun, ignoring retransmission delays) can be reduced to seconds or less, although only past and projected state of an instrument can be accessed. Nonetheless, since the proxy knows the satellite's schedule and may negotiate for communication resources, it should be possible to schedule satellite resources and even implement co-scheduling.

In the opposite direction, satellites requiring Earth-bound IPG resources, such as large-scale processing, can inform their proxies of the needed computations and the next communication time. The proxy can then request the calculations and store the results until the satellite has a scheduled communication window. As the number of spacecraft grows large and operations move farther from Earth, particularly on the other side of the Sun were direct communication is impossible, communication/computation satellites may be placed in various orbits to support exploration spacecraft without sending messages all the way to Earth. The spacecraft may form the Solar System's IPG backbone much as high-speed links and routers form the current Internet backbone. This backbone could also store and forward important information on space weather, for example, solar flares observed by near-Sun satellites.

If these problems are solved and large numbers of inexpensive exploratory spacecraft are integrated into a solar system wide IPG, it may be possible to create a market economy to drive the exploration of the solar system. A market economy requires a large number of producers and a large number of consumers, no one of which can control prices. If spacecraft and launch are inexpensive, relatively small organizations could operate exploration satellites. With the IPG keeping track of location and capabilities, a system of large numbers of relatively small university grants could provide the funds to purchase observations. In this model, the scientists do not purchase entire spacecraft, but rather a portion of a spacecraft's capability. SpaceDev, Inc. (www.spacedev.com) is pursuing a similar, but necessarily more limited, business model. SpaceDev is trying to fund a deep space mission by selling space/power/etc. on the spacecraft buss as well as by selling data.

A typical Solar System IPG interaction might look something like this: Dr. Potter wants a small probe to sequentially visit and sample 10 Near-Earth carbonaceous asteroids with diameter < 100 m and send back to Earth information about the samples. The probe must carry a sample analysis system weighing 10 kg, with volume 0.1 x 0.1 x 0.2 m, drawing 50 w power during operation, 2 hour operation per asteroid, and data transfer of 10 Mbyte per sample. He wants minimum cost for a mission between 6/1/2010 and 12/1/2011. The IPG suggests the following options:

- Option 1:

- Purchase: MagSail3 (with microthruster) by Space Cadets: $100,000
- Launch: Kilamanjaro EM Rail Launcher, 7am on 6/5/2010: $250,000
- Asteroid sampling: 4/7/2011 - 8/26/2011
- Detailed flight animation: <hyperlink>
- Data relay: NASA repeater 14: $10,000
- Total cost: $360,000
- Potential customers for unused capacity: Dr. Dumbledore, Dr. S. Black, and Dr. Granger.

- Option 2:
  - Lease: LightSail156 from SpaceDev: $40,000
  - Asteroid sampling: 6/7/2010 - 11/3/2011
  - Detailed flight animation: <hyperlink>
  - Data relay: NASA repeater 14: $8,000
  - Total cost: $48,000
  - Limitation: only 8 known asteroids may be visited in this period
    - Space Telescope 12 can perform a search for additional asteroids along the flight path for $5,000 with a 65% chance of finding three or more asteroids with a diameter < 100 meters. An additional $2,000 will be necessary to characterize each asteroid found.

Conclusion

Applying IPG technology to improving launch vehicle cost and safety and integrating our satellites, landers, and rovers into a solar system wide, integrated IPG should benefit most NASA programs and help support vastly expanded commercial space activity. This would be accomplished by integrating widely dispersed computational capabilities, databases, and instruments into a seamless whole, thereby substantially increasing efficiency, productivity and safety.

RLV improvements envisioned by NASA and assisted by the IPG may lead to huge numbers of exploratory robotic spacecraft. While current information technology approaches are marginally adequate for the small numbers of spacecraft in orbit today, these techniques will not scale well. Spacecraft could be integrated as nodes in the IPG, effectively extending the IPG into the vast reaches of our solar system. This would exercise emerging IPG capabilities such as reservations, network scheduling and co-scheduling, and require the development of proxy and other techniques for hiding the large latencies, low data rates, and intermittent connectivity typical of deep space exploration. This could conceivably lead to a true market economy for the exploration, and perhaps exploitation, of the vast resources of the solar system.

References

[Globus 1999] Al Globus, Bryan Biegel, and Steve Traugott, "AsterAnts: A Concept for Large-Scale Meteoroid Return and Processing," (www.nas.nasa.gov/~globus/papers/AsterAnts/paper.html), NAS technical report NAS-99-006. Presented at Space Frontier Conference 8 (www.space-frontier.org/EVENTS/SFC8).

[Leveson 1995] *SAFEWARE: System Safety and Computers*, Nancy G. Leveson, University of Washington, Addison-Wesley Publishing Company.

[Lewis 1996] John S. Lewis, *Rain of Iron and Ice: the Very Real Threat of Comet and Asteroid Bombardment*, Addison-Wesley Publishing Company.

[Manna and Pnueli 1991] Zohar Manna and Amir Pnueli, "*The Temporal Logic of Reactive and Concurrent Systems*," Springer-Verlag.

[MCO 2000] Report on Project Management in NASA, Mars Climate Orbiter Mishap Investigation Board, March 13, 2000.

[Nelson 1993] M. L. Nelson, D. T. Britt, and L. A. Lebofsky, "Review of Asteroid Compositions," *Resources of Near-Earth Space*, John S. Lewis, M. S. Matthews, M. L. Guerrieri, editors, the University of Arizona Press, Tucson and London, pages 493-522.

[Rabinowitz 1997] David L. Rabinowitz, "Are Main-Belt Asteroids a Sufficient Source for the Earth-Approaching Asteroids? Part II. Predicted vs. Observed Size Distributions," *Icarus*, V127 N1:33-54, May 1997.

[SIAT 2000] Report to the Associated Administrator Offices Space Flight, NASA Space Shuttle Independent Assessment Team, 7 March 2000.

[Tufte 1997] Edward R. Turfte, *Visual Explanations: Images and Quantities, Evidence and Narrative*, Graphics Press, Cheshire, Connecticut.

[Wertz and Larson 1996] James R. Wertz and Wiley J. Larson, *Reducing Space Mission Cost*, Microcosm/Kluwer.