An Agile Space Paradigm and the Prometheus CubeSat System

Nicholas A. Dallmann, Jerry G. Delapp, Donald C. Enemark, Thomas D. Fairbanks,
Clifford M. Fortgang, David C. Guenther, Stephen L. Judd, Gayle M. Kestell, James E. Lake,
John P. Martinez, Kevin P. McCabe, John M. Michel, Joseph M. Palmer, Mitchell W. Powell,
Dean A. Prichard, Michael C. Proicou, Heather M. Quinn, Robert S. Reid, Edward B. Schaller,
Daniel N. Seitz, Paul S. Stein, Steven A. Storms, Erica A. Sullivan, Justin L. Tripp,
Adam Warniment, Robert M. Wheat
Los Alamos National Lab
P.O. Box 1663, Los Alamos, NM 87545; 505-665-9879
ndallmann@lanl.gov

ABSTRACT

The Los Alamos National Laboratory (LANL) Agile Space Program (ASP) has developed a paradigm intended to enable new, low cost, rapidly deployed space systems. Aspects of this paradigm in the areas of requirements definition, tailoring of risk, and controlling the costs of reproduction and operations are discussed. A history of the LANL ASP is provided. A high level description of the Prometheus CubeSat system along with its constituent components is included. A description of the hosted payload capability offered by Prometheus Block 2 is provided.

AGILE SPACE

Certain missions are excluded by the established high-reliability space system development approach due to its high costs and long timelines. We have been experimenting with tailoring satellite system development to reduce these barriers. This approach will be referred to as the *Agile Space Paradigm*.

It should be noted that this paradigm is not intended to replace the traditional approach. For systems that require very high reliability, like manned spaceflight and critical national assets, the traditional approach is proven. The efforts described here focus on enabling the subset of missions for which additional risk would be accepted and a possibly more limited capability tolerated if the cost and timeline could be drastically reduced.

When low system reproduction (manufacturing) and operations costs are realized, non-recurring-engineering costs are the most significant component of system development (possibly disregarding launch). Therefore, development schedule is a critical driver for controlling the total cost of ownership of a new satellite system as well as for being responsive to new missions. There is a theme throughout the efforts discussed in this paper of heavy software and hardware re-use for risk reduction and cost savings. Much of the software and hardware is common amongst the satellite and the ground station as well as amongst their constituent subsystems.

Design for manufacturability and testability is critical as it saves costs. Additionally, we have the goal of partnering with and performing technology transfer to industry. This will require designs that are modular, easily built, and easily tested.

The paradigm is, in part, intended to provide areas where risk and cost can be traded. An

ASP goal is to demonstrate the ability to tailor risk and provide a continuum of development options from the very rapid, low cost, higher risk to very low risk, higher cost. The operating point within this continuum is then driven by mission needs and available budget. Low reproduction costs open the space of improving system reliability to include redundancy at higher levels of system integration (satellite and ground station).

Requirements Definition

Staying focused on the mission is a critical theme. Too often, customer requirements are specified at too low a level to permit flexibility in the design. The requirements, at times, come from previous systems and/or the desires of many disparate groups. They often tend towards the limit of the physically possible rather than what is minimally acceptable. Due to the high costs of space systems, meeting all desires of possibly many stakeholders is often a driver causing a spiral of increasing cost.

We define requirements based on a small number of use cases or operational scenarios. These are the "level 1" requirements. A minimum set of critical, high level, but still quantitative, performance requirements are flowed from these scenarios to provide a common understanding of the capability to be provided. These are the "level 2" requirements. Small compromises on requirements at this level can lead to enormous cost savings. Regular, detailed, communications between the customer and the design team are very important to a common understanding so that quick, informed, decisions can be made as trades are encountered. However, the design team retains complete control of the level 3 requirements and therefore the apportioning of the level 2 requirements amongst the subsystems.

It is important the design team be enabled to independently handle implementation issues, resource allocation, and risk reduction plans. However, for a rapid, risk tolerant development; it is equally important the customer have a small program office with the authority and willingness to make rapid decisions with respect to level 2 requirements, funding, and schedule.

Tailoring Risk

The willingness on the part of the customer and the development team to accept risk can provide substantial cost reduction. We believe that often another cost-increasing spiral comes from the high cost of the system driving extremely high reliability requirements that further increase costs.

It is still a satellite, it will be out of reach once launched and issues with the hardware or launch software could render it useless. Qualification is therefore required. Our goal is to develop methods for intelligently reducing the level of part and subsystem qualification by retaining the most valuable and cutting the least valuable activities. It is critical that potential failure modes are understood and, at LANL, a team of experts is relied upon to design the system to handle those modes.

Reducing Production Costs Along with Size and Mass

Traditional ultra-high-reliability, space-qualified components are significantly more expensive than their commercial counterparts. Employing commercial off the shelf (COTS) components can therefore greatly reduce the reproduction costs associated with the satellite. This is fairly obvious. However, there are also aspects of indirect costs savings as well as risk reduction from the use of COTS parts that may not be as obvious.

COTS components can vastly increase the level of integration and therefore significantly reduce the size and mass of the satellite. This leads to a reduction in launch costs that can have dramatic effects on total cost of system ownership, especially for constellations.

COTS usage can also streamline development and reduce risk at final integration. The lead-time and cost of components become low enough to permit early and frequent testing with hardware that is true to flight. For example, in the LANL Agile Space Program, it is the intention to have a test satellite and ground station sitting on the desk of each of the software developers. Better testing and more frequent testing at the full system level allows issues to be found and corrected early in development, making the final satellite build nearly free of issues. Since the software developers are testing the system in its entirety on their desks, there is little concern or risk when performing functional testing on the flight vehicles. Yet another benefit from the lower costs is the reduced pressure on yield. Although the team should endeavor to understand any failure, the fact that subsystems are significantly less expensive permits a higher number of spares and therefore a more rapid build process. This is especially true at higher volumes.

Allowing the use of COTS components can increase capability. Traditional space components are usually a few generations behind the state-of-the-art. It is the intention at LANL to regularly incorporate new technologies in our systems to avoid parts obsolescence and to provide an ever increasing capability. The goal during this is to hold to a constant or decreasing reproduction cost.

Despite all the advantages, COTS parts must be used with care. Our small team draws upon significant space experience and part radiation testing expertise.

Reducing the size and mass of components is required for CubeSats. However, we hope to scale the approach and technologies to larger small satellites. This increased volume efficiency could provide significant aperture increase for a given satellite size.

Simple, Automated Operations

Once the cost of building and launching satellites and building ground stations are minimized, operations for the lifetime of a system may become the most significant cost element. Many traditional satellite systems require regular manning, sometimes around the clock by multiple highly trained individuals. For certain critical systems, continuous human monitoring will probably remain the correct answer for long into the future. However, for the reduced cost, risk tolerant efforts discussed here, simplicity and automation that reduce manning time to a minimum are key cost reducing goals.

One of the goals of the Agile Space paradigm is to keep system operations extremely simple. This entails designing a system that is easy to use and for which operator error cannot cause damage. It should be possible to train a new user to operate the system in much less than a week. A strict focus on the mission is required to keep the system simple enough to permit this. There is likely a trade between features and ease of use. The goal is that system operators do not need to be traditional space operators and it can be a secondary, part time duty. It also enables potential tactical control of systems in support of national security missions. One should be able to walk away for days or weeks and, upon return, be able to easily operate the system.

Another goal is to develop a “configured, not scripted” system. The satellites in such a system do not receive regular detailed scripts including all the actions and times for the next period. The satellite instead receives a list of tasks and acts upon them automatically as time permits. For example, if it were desired that a satellite take a picture of the same point on the ground at each opportunity, then simply the latitude and longitude of the point and perhaps an elevation or range threshold is all that should be required. The satellite can handle the rest. Then, if a user does not change the configuration, the satellite will continue to take pictures of that point for days, weeks, or any duration until the tasking is changed. Regular schedule development is not required. The algorithms that determine what the satellite should do next operate on board the satellite. This lends itself to a high level of system automation. A user can sit down and configure the system for long into the future.

The ground system has a configurable, simple, set of automatic messages it can send in short emails, text messages, etc. This gives status of the system, successes, warnings or errors that require attention, and possibly just simply reminders that a user could do something if they wish.

Developing Capability after Satellite Delivery and While on Orbit

Software and FPGA firmware development often becomes the pacing tasks in developing a new space system. Although many systems have been capable of significant reprogramming while on orbit, it is our understanding that it is rare for this to be well used. The Agile Space paradigm assumes throughout the development that some, or possibly most, of the software capability will be developed after hardware delivery for launch. This pushes for a highly automated, easy to use, and very safe software upload capability.

Testing full mission capability with high levels of code coverage can take significant time and money. The ASP approach is to launch the system with failsafe software and focus heavily on functional hardware and minimalist reliable failsafe software. It is critical to ensure that the system will fall back to the failsafe mode should it be required. The failsafe software should be relatively easy to test as it is capable of little more than enabling hardware testing and uploading new software.

A competency that has developed amongst the LANL ASP team is performing system functional testing and debugging new capabilities on-orbit.

New Types of Missions Enabled

This approach is intended to enable many new missions. Some examples of areas include:

1. New science missions
2. Emerging national security threats
3. Specialized missions or missions underfunded for the traditional space approach
4. Technology demonstration
5. Large constellations for coverage and redundancy
6. Organizations that are not traditionally space-focused can own and operate satellite systems

What is a CubeSat?

The disruptive concept of a CubeSat was developed in the late 1990's at California Polytechnic and Stanford universities.[1] The key enabling innovation of the CubeSat is the standard container/dispenser (see Figure 1). The dispenser is qualified to the point where launch providers are now regularly giving, or selling for low cost, a ride to orbit as a secondary flyer. The satellite is of a standard 'unit' size. 1U is a 10cm cube with a mass, originally, of 1kg (1 liter).

The initial focus was in enabling university teams to launch their own satellites, providing a great educational opportunity. After some notable university successes, the possibility of using CubeSats for national security and science missions began to be explored.

The most common dispensers contain 3U behind a door. 3U has therefore become one of the most common sizes. All of the LANL CubeSats to date have been 1.5U and therefore two fit within a 3U dispenser. Keeping to this form factor saves on launch costs because two 1.5U satellites can be launched for every 3U of dispenser volume.

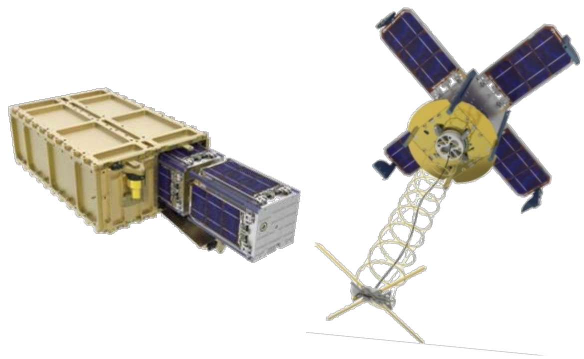

Figure 1: Two 1.5U Prometheus Block 1 CubeSats Being Ejected from a NASA NLAS Dispenser and then after Timed Deploy

The LANL Agile Space Team is working in CubeSats but is not limited to them. LANL is hoping to be tasked with a mission that would permit the application of approaches learned from CubeSats to larger but still small satellites. Developing with COTS parts in a CubeSat form factor has led to satellite systems that are very small. Most of the systems are scalable to larger satellites and will reap tremendous benefits in the area of volume and mass efficiency. Clear aperture to the edges may be possible for small satellites.

HISTORY OF THE LANL AGILE SPACE PROGRAM

The LANL Agile Space team has launched 12 CubeSats and is expecting to launch at least 10 more in 2016. These have been part of the completed Perseus project and the currently active Prometheus project. Here, a brief history and goals of these programs is provided before a focus on the technical details of Prometheus.

Perseus

Figure 2: Perseus Satellite and Ground Station Components

In 2008, LANL began its first CubeSat project, called Perseus. Perseus was designed as a system including ground assets and satellites (see Figure 2). The development took about 6 months. Satellite reproduction costs after NRE was estimated at $25k. The goals of the Perseus system were to:

1. Demonstrate the ability to build and launch a useful satellite quickly and at very low cost.[2,3]
2. Demonstrate a satellite system simple enough to be operated and maintained by non-space experts with little training.[2,3]
3. Demonstrate a tactically relevant communications capability to a CubeSat with an extremely modest ground station footprint.[2,3]
4. Validate the Agile Space management and development methodology.[2]

On December 8, 2010, four Perseus CubeSats were released into a roughly circular 300 km orbit at a 34.5° inclination. The lift vehicle was a SpaceX Falcon 9.

All four satellites and ground stations performed flawlessly throughout their three-week lifetime. The system met all of its goals.

Prometheus

Figure 3: Prometheus Block 1 Satellites Just Prior to Stowing and Integrating into the Dispensers

Following the success of Perseus, in 2012 the LANL Agile Space Team began work on the next phase, called Prometheus (see Figure 3). Prometheus has the goals of:

1. Demonstrate the ability to build and launch a useful satellite quickly and at very low cost.[2,3] Focus on maintaining low reproduction costs.
2. Demonstrate a satellite system simple enough to be operated and maintained by non-space experts with little training.[2,3] Focus on highly automated operations to control costs.
3. Demonstrate a tactically relevant communications capability to a CubeSat with an extremely modest ground footprint.[2,3] . Increased data rates over Perseus.
4. Provide sufficient operational time on orbit to assess:[3]
    a. Potential concepts of operations for a tactically controlled space system
    b. Costs of the system.
    c. The operational utility of a CubeSat system.
5. Validate the Agile Space management and development methodology.[3]

In November, 2013 eight Prometheus Block 1 satellites were dispensed to a circular 500 km altitude, 40.5° inclination orbit from the upper stage of a Minotaur 1 rocket launched by the Department of Defense's Office of Operationally Responsive Space (ORS).[1]

Block 1 demonstrated successes include:

1. Both configured and scripted tasking demonstrated on the satellite.
2. Doppler correction of ephemeris.
3. Regular secure communications achieved with all eight of the Block 1 satellites and maintained for many months.
4. Regular, fully automated, easily configured "lights out," operations at multiple ground stations.
5. Remote, networked control of ground stations.
6. Autonomous system anomaly resolution.
7. Regular automated and easy to use code upload and reprogramming of all microprocessors and software defined radio (SDR) FPGAs.
8. Automated file transfer from ground station to satellite and satellite to ground station (see automatically downlinked picture in Figure 4)
9. Developing software capability after delivery and launch. Testing that software on orbit.
10. Attitude control for both Sun pointing and ground point tracking.
11. Manually variable data rates.
12. Fully encrypted communications.

Although successful, Prometheus Block 1 was not perfect. On orbit testing of Block 1 has provided many lessons for Block 2. This is in line with the rapid, risk tolerant development approach being employed. The goal is to continually fix issues and improve the capability while holding to a relatively constant reproduction cost. This type of incremental, heritage building development is commonplace in traditional space programs but is sometimes lacking amongst CubeSats. Prometheus Block 2 is underway and will benefit from these lessons learned.

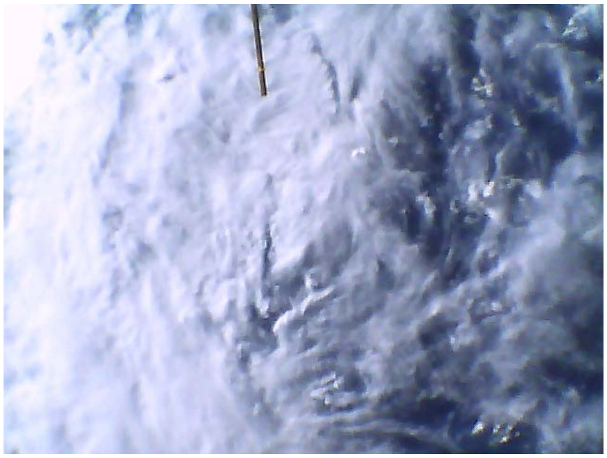

Figure 4: Photo of Heavy Cloud Coverage Taken by Block 1 via a Script of a Selected Location

PROMETHEUS SATELLITE ARCHITECTURE

Prometheus Block 1 is actively being tested and improved on orbit. Prometheus Block 2 is in development for a planned launch date in 2016. The following sections are intended to give a status and overview of the Prometheus satellite and ground station technologies.

Structure

The Prometheus structure (see Figure 5) has been developed with emphasis on accessibility and modularity. The system breaks into three major pieces. The 'top' is the housing for the analog processing and antennas. The 'middle' is the card cage housing the software defined radios (SDRs), command and data handling (C&DH), and attitude determination and control system (ADCS) subsystems. The 'bottom' is the power system. All subsystems in all three pieces are connected to each other by a single backplane. It is intended that each subsystem be as independently testable as possible.

A novel deployment system was developed for Block 1 and will be improved upon for Block 2. It is easily reset without violating the satellite, allowing for repeated testing.

Prometheus has a goal of very low (<$150k per satellite) reproduction costs. For Prometheus Block 2, one of the incremental areas of improvement is in reducing the complexity of manufacturing. While Block 1 made steps towards streamlining manufacture, it still included much hand labor. Specifically, printed circuit boards are mostly assembled out of house, but there remained additional touch labor at LANL to complete the final assembly. Also, there was still some hand wiring required in the solar panels and power system. For Block 2, as much as possible, by including more flex circuits and connectors, printed circuit boards will come to LANL from out-of-house assembly ready for final inspection, testing, and then to be placed into the satellite. For Block 2, there is a goal to reduce the number of mechanical components by half. Rapid subtractive machining has been mandated for virtually all metal structural components. All plastic components will be printed using low outgassing thermoplastics, as was pioneered on Block 1, via a modern fused deposition modeling (FDM) processes. These efforts will vastly reduce the cost and lead-time of individual components. LANL plans to continue these trends into the future to continually reduce reproduction costs. There are added risk reduction benefits. The ease of fabricating and replacing components allows for additional spares and reduced time lost due to issues during satellite final integration and test. The less one needs to violate the satellite to replace or debug a faulty component, the better. Schedule remains the most important driver.

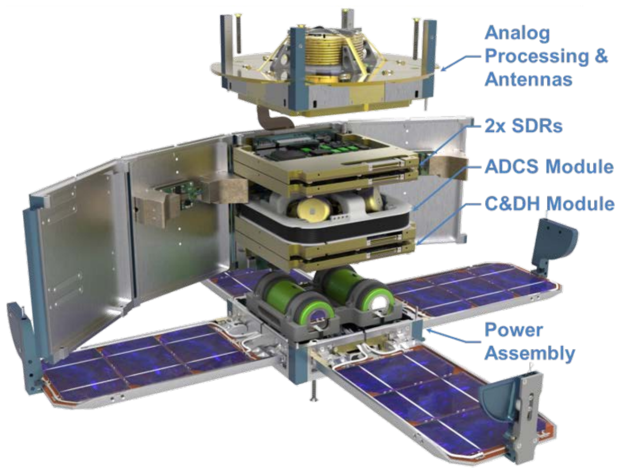

Figure 5: Prometheus Block 1 Internal Structure Showing Modularity

Subsystems

Figure 5 shows the layout of the individual subsystems within the satellite. There are two software defined radios (SDRs) with corresponding antenna and analog processing chains, an attitude determination and control system (ADCS), and a command and data handler (C&DH).

Power

The power system includes solar panels that are bonded and welded using techniques developed at LANL for its small satellite efforts. It houses the converters for all the major voltage rails within the satellite. The batteries are charged from the solar panels when illuminated. The charging has maximum power point tracking built in for maximum efficiency. Prometheus is designed to not require active attitude control to be functional. In Failsafe mode, in which there is no attitude control, the satellite draws a low amount of power and, given that there are solar cells on both sides of the panels, there is sufficient power for Failsafe operations and charge recovery under most orientations. Block 2 will have twice the solar panel area (compare Figure 6 to Figure 5). This will provide for the capability additions from Block 1 to Block 2 in the areas of ADCS and SDR improvements, as well as provide power for a possible hosted payload.

Software

The automation, ease of integration, and low cost requirements for Prometheus gave the LANL team a unique opportunity to design the application software and shared libraries for Prometheus from scratch. A standard microprocessor (a 32-bit ~200 MHz ARM) was selected for use within all subsystems. Therefore, a great deal of code could be shared amongst all the subsystems. We initially assigned individual hardware boards (e.g., C&DH, ADCS, SDR) to software developers for board bring-up and design validation. We quickly realized that there were many common functions needed on all boards, so we transitioned to creating a common code base shared across several board-specific applications. Networking, file I/O operations, low-level hardware configuration, and many basic commands are applicable to all subsystems and the code for these is shared. This provided a large reduction in overall system development cost.

For Block 1, a software bank structure was developed in which every subsystem has a failsafe software storage area (that can never be overwritten) and two banks for uploading new application code. The hardware has been designed with multiple levels of watchdog to ensure that any issues with new code upload reliably causes a predictable return to the failsafe code. This permits on-orbit development of capability with little or no danger of damaging or losing a satellite should imperfect code be uploaded. For Block 1, all subsystems in many of the satellites were updated many times. At the time of this writing, the team has absolute confidence that they can test a new capability on orbit and, if a mistake is made, the satellite will return to failsafe with no negative consequences.

Prometheus utilizes a single networking layer, developed at LANL for small satellites, to interconnect the various microprocessors within the system. This network code is part of the common code base shared across all subsystems. LANL assumed that the application software will have a significantly longer life than any specific system or piece of hardware. Therefore, an early goal was to abstract interconnecting the system away from a particular satellite or ground station implementation. The networking model was designed to be independent of the hardware so that new links can be added and operated over different and possibly not yet defined hardware standards in a seamless way that is invisible to the application software engineer. This network extends from the ground station to each subsystem in the satellite.

The networking layer is a static 'circuit' protocol derived from Asynchronous Transfer Messaging (ATM). It is called "Satellite ATM" (SATM) because of design changes, such as its reduced header overhead, relative to standard ATM. SATM permits the creation of virtual circuits between any nodes within the system. Variable length messages are easily interleaved and moved between network nodes without forcing a complex application level decode and header parse burden on any node. This is particularly beneficial to the software radio. For example, a circuit can be established to provide apparently seamless communications between a computer in the ground station and a microprocessor in one of the subsystems within the satellite. As will be discussed in the payload hosting section, this can be extended to connecting payload developers at the ground station directly to their payload. The intervening network, number of hops, and hardware implementation is unimportant. We have routed SATM traffic over SPI, UART, and RF links.

All software for Prometheus was developed at LANL. The goal, partially realized in Block 1, and expected for Block 2, is to have a satellite and a ground station on the desk of every developer. The team stores code in a common repository and frequent merging and system level testing by all developers is performed as part of their development process. This leads to more thorough and more frequent testing as well as vastly increased confidence at the time of final satellite integration and functional testing.

Command and Data Handler

The command and data handler (C&DH) is the central hub within the satellite. From a network point of view this is transparent, but at the hardware level the C&DH controls the power to the individual subsystems, monitors the health of the batteries, controls initial solar panel deployment, receives commands from the ground, and stores the system logs.

In Block 2, one of the principal functions of the C&DH is to manage targeted activities (for example, taking a picture or establishing a link with a ground asset). The Prometheus system is fundamentally configured and not scripted. The target manager is configured with a simple list of latitude and longitudes of interest along with rules governing different modes of operations. It continually propagates the orbit of the satellite and the position of the ground assets, and when line-of-site access is possible, it begins the target activity. When hosting a payload in Block 2, the CDH additionally schedules payload actions (e.g., power-on, power-off, extract data, etc.)

Testing of Block 1 revealed that an independent source of Satellite Vehicle (SV) ephemeris is important. Prometheus Block 1 relies on ground-based tracking to determine the ephemeris and uses Two Line Element Sets (TLE) provided by Joint Space Operations Center (JSpOC). Immediately, after launch and deployment, all 8 Block 1 SV's were close enough to use a common TLE. As time progressed, the SVs spread out, TLEs were not sorted out between Prometheus and the 20 other CubeSats dispensed from that launch, and communications began to fail due to Doppler correction errors. LANL, owning the entire system, was able to respond in a few weeks by updating its radio firmware at the ground station to make additional frequency measurements and developing a toolset to update TLEs based on these measurements. Also, the operation of the satellite relies on on-board propagation of the orbit and since the predictive quality of a TLE for a low altitude satellite decays with age (over several days), the target manager (scheduler) requires that the TLE be refreshed regularly on orbit.

Block 2 Prometheus will include a GPS receiver module. This will enable improved instantaneous orbit knowledge and independent calculation of our orbit ephemeris. On-orbit orbit determination of our ephemeris, using the GPS data, avoids issues with age of externally-provided TLEs. In Block 2, we expect to establish initial communications based on launch provided state vectors. Once the GPS operation has been confirmed, the ground station will receive ephemeris from the satellite during each pass.

COMMUNICATIONS SUBSYSTEM

Ground to space and space to ground communications for Prometheus are facilitated by an encrypted half-duplex radio subsystem developed entirely at LANL for small satellites and CubeSats.

The concept was to develop a radio that can communicate with itself. This permits nearly exactly the same hardware, firmware, and software to be used in both the satellite and the ground station. There is actually a second set of satellite flight boards within the Prometheus ground station. This provides a tremendous reduction in development cost and testing. The networking layer runs over the communications subsystem.

The radio can be separated into two logical parts; analog processing and digital software defined radio (SDR).

Analog Processing

The analog processing includes a portion of the transmit and receive paths between the digital data converters and the antenna. It resides on its own printed circuit board so that it can be independently modified for different missions.

The carrier frequency on both transmit and receive can be independently set, allowing a channeling scheme for multiple satellites or frequency agility should interference limit the quality of communications. One of the design decisions in Block 1 and retained in Block 2 is that the SDR operates with a common intermediate frequency (IF). Therefore, should a need arise to operate at different carrier frequencies, this can be accomplished by modifying just the analog processing. Small changes are accomplished by software controlled configuration, large changes by limited redesign. A future goal of the system is automatic frequency adapting to avoid interference.

The ADCs and DACs operate at baseband. For the transmit path, the DAC output is up-converted to the desired carrier frequency. The receiver is a super-heterodyne optimized for minimal noise figure and high sensitivity. This facilitates an important system goal of a very small ground station footprint.

Digital Software Defined Radio

The digital portion of the radio is made up of a microprocessor—the same model is used system wide—that configures and controls a high performance SRAM-based FPGA. Potential upset of the SRAM interconnect fabric in the FPGA is mitigated through multiple means. As with all subsystems within the Prometheus satellite, the SDR is fully reprogrammable on-orbit. Safe reprogramming of the microprocessor and FPGA has been demonstrated on Block 1.

The Prometheus radio is designed for weak signals or long distances. Thus the SDR is optimized for low signal to noise ratio (SNR).

The FPGA performs the required digital signal processing (DSP). The current algorithms programmed into the FPGA operate at the in-phase and quadrature (IQ) baseband. The algorithms include programmable IQ-to-IF up-conversion, programmable IF-to-IQ down-conversion, low SNR packet acquisition, carrier and time synchronization, modulation and demodulation, forward error correction (FEC) channel coding, and some of the cryptographic pieces. The computational requirements are quite demanding, but the use of a modern high performance low power FPGA makes this possible even within the limited resources of a CubeSat.

A LEO satellite's communication system will see large Doppler shifts due to the high range-rate-of-change. Operating at minimum SNR demands a coherent demodulator. However, acquisition and tracking of the carrier in the presence of high Doppler shift is challenging. Both the satellite and the ground station are capable of pre-correcting Doppler. On the satellite, both the satellite and ground station positions are continually recalculated by the target manager, providing real time frequency updates to the radio on board the satellite. Or, at the ground station, the positions can be calculated by the GUI, providing the same updates.

There is a default data rate the satellite expects for initial communications. However, to maximize transferred data volume, a critical capability of the radio subsystem is varying data rate based on the channel capacity. Manually changing the data rate with a ground station command was demonstrated on-orbit with Block 1. For Block 2, the application software will automatically adapt the data rate based on signal strength metrics produced for each reception. In the future, it is planned that adapting the waveform will permit even better channel capacity utilization.

Prometheus Block 1 was a narrow bandwidth communications system. For Block 2, the communications will be direct sequence spread spectrum to reduce the power flux density per bandwidth on the ground in order to conform to spectrum regulations.

A prime goal of Prometheus is a modest footprint for ground assets. This has led to a radio designed for communication over highly disadvantaged links. Low SNR, low error rate communications is paramount, while bit rate is secondary. However, the Prometheus SDR is very flexible due to the on-orbit reprogrammable FPGA and microprocessor, and due to high speed and high dynamic range analog-to-digital (ADC) and digital-to-analog (DAC) converters. If the hardware and physics permit, support for new missions with different radio requirements can simply be uploaded. For instance, using the same Prometheus SDR hardware, it is possible to support bandwidth-limited high data rate applications using completely different modulations, DSP algorithms, and FEC channel codes. This would be possible with larger ground station antennas. Such an upgrade could be made after launch.

Satellite Antennas

Custom novel deployable antennas were developed for Block 1 and are being updated and refined for Block 2. There are two SDRs each with their own independent antenna. One SDR has a higher gain antenna at a higher carrier frequency and is intended for higher data rate or more disadvantaged ground assets. This antenna requires the ADCS and pointing.

The high gain antenna for Block 1 and Block 2 is a helical that is compressed to about 5% of its deployed volume when stored in the dispenser (see Figure 6). The low gain antenna, at a lower carrier frequency, is a more isotropic crossed dipole.

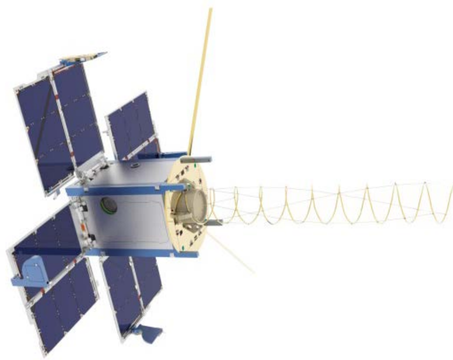

Figure 6: Prometheus Block 2 Showing Deployed Helical Antenna and Crossed Dipole Antennas

SATELLITE ATTITUDE DETERMINATION AND CONTROL SYSTEM

The Prometheus Attitude Determination and Control System (ADCS) was developed entirely at LANL specifically for the Prometheus project and future Agile Space small satellite programs. The software libraries were developed from scratch to facilitate a configured-not-scripted system.

There are four basic modes of operation supported by the Prometheus Block 2 ADCS:

1. To support high data rate communications, the high gain antenna must point to and track a ground station throughout a pass.
2. To maximize available solar input, when the Sun is in view and communications are not occurring, the solar panels are oriented normal to the Sun.
3. When neither the ground station nor the Sun is in view, the ADCS can be set to reduce its power draw.
4. For future hosted payloads, a nadir pointing mode is being added to Block 2.

Transitions between these modes are controlled by the target manager software on-board the satellite. The actions of the target manager are controlled by configuration data periodically uploaded by a ground station. This configuration data includes: locations of the ground assets, radio communications parameters, and some additional configuration values. The location of the satellite, the location and access to the ground assets, and access to the Sun are continually calculated on board. The target manager uses this information to choose between target-pointing, sun-pointing, nadir-pointing, and free-floating operations of the ADCS. The target manager operates at 1 Hz; in each iteration, it propagates the location of the satellite and the location of the ground sites in Earth-centered inertial JD2000 coordinates. From this, access to ground sites as well as access to the Sun is calculated. The target manager software will, in real time, compute the desired attitude (the command quaternion) of the SV.

One of the major accomplishments of Block 1 was developing the ability to efficiently perform on-orbit testing and characterization of the ADCS. The ADCS software on multiple Block 1 satellites was updated many times as improvements were implemented and new features were added. An ADCS test is defined by a fairly simple, human readable, configuration file. The Prometheus team developed the capability to fully automate the process of up-linking configuration files, executing an on-orbit test, and downlinking the resulting data. To start a test, the ground station is configured to uplink a SV configuration file and downlink the corresponding logs created during the test. On the next pass, the ground station automatically uplinks the configuration file and commands the ADCS to run it. Over subsequent passes, the ground station automatically downlinks the log files.

On-orbit tests of the ADCS from Block 1 were successful. Sun and ground tracking modes were demonstrated and much was learned in the process. Our experience indicated that increased actuator control authority as well as the addition of another sensor, not dependent on the Sun, are required to support all desired maneuvers reliably at all times. These additions are in the Block 2 design.

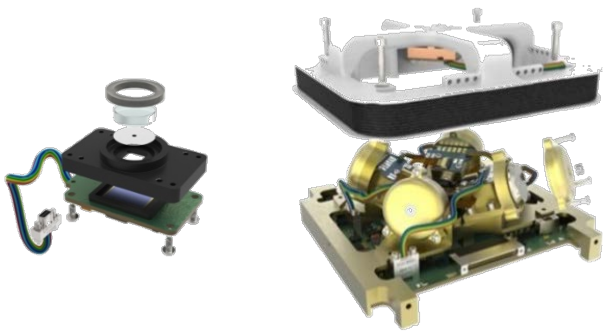

Figure 7: Sun Vector Sensor (Left) and Block 1 ADCS Module (Right)

Navigation Library and Control

A library of functions was written to support the ADCS and target management systems for Prometheus. The library includes basic vector, matrix, and quaternion operations, models of the Earth's magnetic field, Sun location, orbit propagation, coordinate system transformation, time transformations, Earth surface point propagation, attitude determination and control, etc. For Block 2, basic image processing, star catalogs, and pattern search operations will be added to support the addition of a star field sensor (SFS).

The ADCS is configured with a file containing the SV TLE, a list of ground locations (latitude and longitude) at which to point the antenna, and parameter matrices for control loop gain, SV moments, and sensor/actuator rotation. The ADCS computer runs its control loop at 1 Hz.

Attitude Determination and Sensors

For each iteration of the control loop, the current orientation of the satellite (determined attitude quaternion) is updated. Models of the Earth's magnetic field vector and Sun ephemeris are run on-board in real time to produce reference vectors. Measurements of the Earth's magnetic field vector; the Sun vector, if available; and an integration of the on-board gyro are used in correlation with the reference vectors to determine the attitude.

In Block 1, the attitude sensors include three independent, orthogonal, Sun Vector Sensors (SVS); a vector magneto-resistive magnetometer; and a 3-axis MEMs gyro. The SVS was designed at LANL for the Prometheus project (see Figure 7 left panel) because, at design time, a suitably small SVS with sufficient field of view and resolution was not commercially available. The magnetometer and gyro are commercial components.

On-orbit testing of Block 1 successfully demonstrated attitude determination. However, for reliable attitude determination when not in view of the Sun, LANL decided that Block 2 should include an additional sensor not dependent on the satellite's location in its orbit. Block 2 will, therefore, include the same sensor suite as Block 1 with the addition of a self-contained star field sensor (SFS) (see Figure 8 left panel). This compact SFS will provide a periodic, 3-axis attitude fix. To keep the sensor simple, a simple, LANL-developed, "lost-in-space" pattern match algorithm will be performed with an on-board star catalog. The SFS will provide reliable attitude determination at any point in orbit. The SFS will also facilitate significantly improved precision in our attitude determination for improved performance of the Prometheus communications mission as well as for future missions requiring more stringent pointing requirements.

Attitude Control and Torque Actuators

Given the command quaternion (from the target manager) and the determined attitude quaternion, the 'error quaternion' between the command and determined attitude is calculated during each iteration of the control loop. This error is used, in conjunction with configurable loop gain matrices, to determine the torque required to correct the error in the attitude via an optimal motion.[4]

Block 1 includes two types of torque actuators. The primary control actuator is a set of four kinematically redundant reaction wheels arranged in a pyramid formation. The second actuator is a single torque coil intended only to dump small amounts of angular momentum and not for active control. The torque coil on Block 1 has not been tested to date as vehicle angular momentum after dispensing was sufficiently small.

For Block 2, the torque coil has been replaced by three orthogonal torque rods. The principle purpose of the torque rods will be to dump all vehicle angular momentum at the beginning of the mission and, thereafter, as needed. Also, to improve the reliability of the pointing, the range of maneuvers possible, and to support a 3U CubeSat with a hosted payload, the angular momentum storage of the wheels will be significantly increased for Block 2 (compare Figure 8 right panel to Figure 7 right panel).

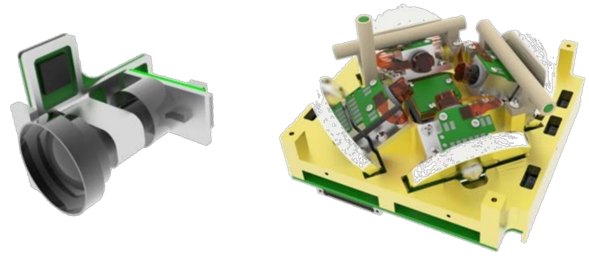

Figure 8: Star Field Sensor (Left) and Block 2 ADCS Module (Right)

SATELLITE TESTING

Radiation Testing of COTS Parts

An enabler for controlling costs is the use of commercial off-the-shelf (COTS) parts. Modern COTS parts are reliable, inexpensive, and available on a short timeframe. One of the problems with COTS is, however, the possibility of susceptibility to the space environment. LANL has extensive expertise in house in radiation testing of microelectronics and the Prometheus team takes this very seriously. If selected components do not have previous radiation test or flight heritage, then that are tested prior to flight use. The goal of radiation testing was to give a high confidence while keeping testing costs low. For example, random samples were tested and the traditional practice of lot testing components was not performed assuming lot-to-lot variation is generally small for high volume commercial components.

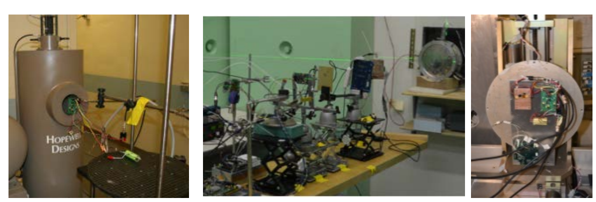

Figure 9: Radiation Testing: Gamma TID at LANL (Left), Neutron SEE LANL (Middle), Heavy Ion SEE at LBNL (Right)

Figure 9 shows setups at three different facilities as components are tested for Prometheus Block 1. Similar testing is being performed on new components as they are added to the Block 2 design. Most modern CMOS electronics are relatively hard against total ionizing dose (TID) in that charge trapping oxide volumes are small. Also, for LEO satellites, the TID levels are low (only a few krad for missions of a few years in duration). For Prometheus, only a few components were tested for TID. Many components (>20) were tested for single event effects.

Parts were not necessarily ruled out if they did not pass a traditional radiation test. The important results of these tests were the failure symptoms the parts would exhibit. Many of the components were protected by additional analog circuitry and via supervisory functions on rad tolerant FPGAs.

Functional and Environmental Testing

Although controlling costs is the key enabler, one must always remember that it is still a satellite. The LANL team believes strongly that testing the fully integrated satellite in a relevant environment is critical to success. The testing plan for each flight satellite includes an initial baseline functional test. This is followed by a battery of environmental tests including vibration and thermal vacuum. Finally, functional testing is repeated once again to verify full flight readiness.

Verifying full mission capability under all possible scenarios a highly automated, configured-not-scripted, satellite system might encounter would require an extremely costly test campaign for each satellite. Also, it would be very difficult to get software coverage and coverage of the hardware ranges if operating in a mission configuration. Functional testing is therefore focused on hardware and critical failsafe software functionality. A single, multi-tab, LabVIEW program was developed to permit basic in-family functional testing and logging of all hardware subsystems within the satellite (see Figure 10). This testing capability was developed in parallel with the subsystem development and used to test and debug the subsystems as they were developed. All subsystems provide a common interface to this capability. This is another re-use advantage the LANL team reaped by developing the whole system. In Block 1, the connections were made via external debugging connectors available for each subsystem. The flight code operating on board the satellite supported the LabVIEW interface. The critical flight software includes wakeup, secure connection, code file upload, and new code loading. It is also critical that the satellites will automatically fall back to the failsafe software should there be issues with new code loads. This is capability is tested on each satellite as part of the formal satellite functional test plan.

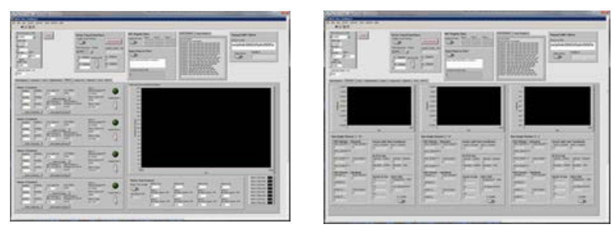

Figure 10: Hardware Testing: Reaction Wheel Test Tab (Left) and Sun Vector Sensor Test Tab (Right)

LANL has a full satellite environmental testing capability available in house. Prometheus is fully qualified to the General Environmental Verification Specification (GEVS) based on a relatively harsh launch and on-orbit environment.[5] The approach utilized an Engineering Qual Model (EQM) satellite put through qualification testing. After this unit passed, all flight units were subjected to acceptance level testing. All units were put through 3-axis random vibration and lengthy thermal vacuum cycling (see Figure 11).

Figure 11: Block 1 Random Vibration (Left) and Thermal Vacuum (Right) Testing

In parallel with the satellite testing and available after launch is an “EDU Lab” which includes a satellite and a ground station. The mission capable code is tested for functionality in this environment prior to upload to the on-orbit satellites.

PROMETHEUS GROUND STATION

The ground station for Prometheus will only be briefly discussed here. The team has written another paper that provides significantly more detail.[6]

The Prometheus ground station is easy to set up and use. It has a single graphical user interface (GUI) providing an integrated controller for the ground station and constellation (see Figure 12). It is a highly automated system. Mission operations as well as software uploading, file downloading, and general satellite maintenance are all performed automatically by the ground station as opportunity permits.

The required additional development costs for the ground station were very small. As was stated earlier, the radio inside the ground station is the same as the radio within the satellite. There is heavy hardware and software re-use. The navigation library developed to enable the target manager and ADCS on board the satellite is the basis for the antenna rotator control, Doppler pre-computation, satellite access determination, etc. This permits significant testing of these functions.

Part of what makes the ground station so easy to use is that it is principally, as with the rest of the system, configured instead of scripted. The user can easily set up tasking and walk away. The ground station will automatically create configuration file(s) for the satellite and upload them at the next opportunity. Since the ground station can simulate ahead of time virtually everything the satellite will see in the future, Prometheus Block 2 will include a GUI component that will provide a fast simulation of the satellite’s tasking into the future. This will provide the user with the option to potentially change priorities or other configuration rules to mold the results prior to committing to a new configuration.

For Block 1, the ground station could be moved in a few roller cases. For Block 2, the goal is to reduce the volume by a factor of two. However, another ground asset, called a field unit (FU) could be used as a ground station and would easily fit, along with a laptop for control, in a small backpack.

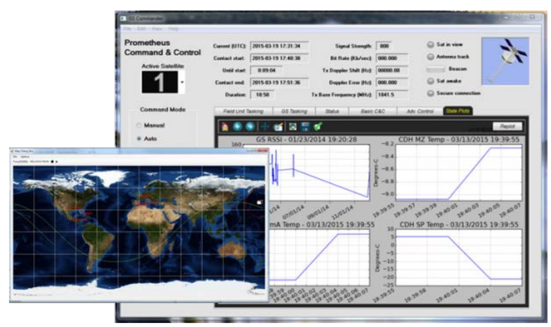

Figure 12: Screenshots of the Prometheus Ground Station Graphical User Interface

HOSTING PAYLOADS

There are many examples of CubeSat payloads that, due to a bus failure, never got an opportunity to be tested properly on orbit. Satellites require significant effort, experience, and facilities to realize an acceptable probability of meeting the fundamental requirements of turning on, surviving the environment, and communicating commands and data with the ground.

After the launch and initial success of Block 1, the team was approached by developers, internal and external to LANL, interested in hosting payloads on Prometheus. Since Prometheus is a mission driven development, payload capabilities were not included in Block 1. However, the team came up with a novel approach that, with little modification to the satellite hardware, can provide a hosting capability. All Block 2 satellites will include this capability.

Hosting on Prometheus will enable payload developers to focus their efforts on the payload, permitting them to leverage the existing investment and the continually evolving technology of Prometheus. More consistently successful CubeSat missions will help the entire small satellite community.

Prometheus Hosting Concept

Prometheus is a 1.5U satellite permitting two satellites to fit within a standard 3U dispenser. Keeping Prometheus at 1.5U has been a continuous engineering challenge but is an important part of controlling launch costs for future constellations. When hosting a payload, a 1.5U payload volume will extend the satellite to 3U (see Figure 13).

Block 1 has demonstrated reliable communications of commands and data files to and from the satellites via inexpensive, easy to use, and highly automated ground stations. Prometheus has attitude control, a power system (with significant margin in Block 2), and an autonomous software target manager that will be extended to control data collection or other actions required by a payload. All of this capability will be provided to payloads.

The hardware changes for hosting are simply the addition of a bolt-hole pattern and two ruggedized connectors. These are located at the end of the satellite away from the antennas (see Figure 13). A 51-pin connector provides general purpose digital logic lines to each of the satellite's subsystems as well as access to the internal power rails. A second connector provides access to the battery and charging circuitry. An interposer board (see Figure 14), will provide the interface between these connectors and the payload.

Two paths for payload hosting are under development. These are described in the following two sections. Both are focused on ensuring the bus will have the highest probability of successfully turning on and communicating with the ground. This will provide the highest probability of success for the payload. The payloads will be isolated (switches on interposer) during initial on-orbit turn-on and the failsafe software will not include payload support. This will permit consistency of testing the bus, and all missions will inherit from previous analysis and testing of the Prometheus satellite running its failsafe software.

There is an additional advantage enabled by this approach to hosting payloads. The connectors provide comprehensive access to the satellite subsystems for easy, non-invasive testing. For Block 2, a "docking station" (see Figure 15) is under development. This is a significant step towards future plans for automated testing during higher volume manufacturing.

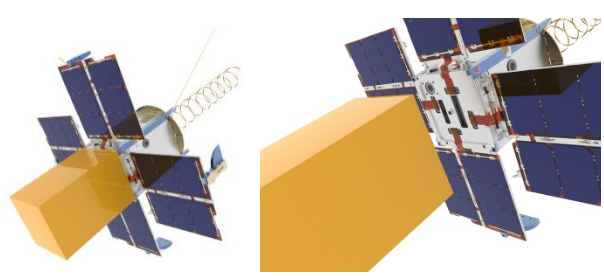

Figure 13: Hosted Payload Volume in Orange (Left) and Volume Retracted to Show Interface (Right)

Hosting Option 1: Standard Interface

Some payload developers may be interested in a simple documented interface they can design to. This option will employ a standard interposer board that will provide power (one or two rails) and communications to and from the payload (one or two basic standards like UART). Additional functionality would be added to the application software on the C&DH within Prometheus to facilitate the payload. Cost savings would be realized in that no additional development on the host side would be required.

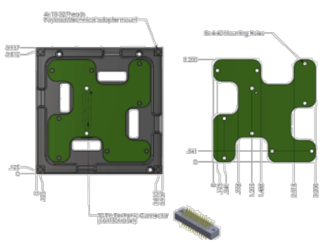

Figure 14: Hosted Payload Interposer PCB

This option will be the lowest cost, but would be somewhat limited in capability. It is expected to be of particular interest to payloads that are early in development and can be modified or baselined to interface with Prometheus.

LANL is currently in discussion with several potential payload providers. Attempting to meet the desires of all of these with a single interface would require large software and firmware efforts, a complex interposer, a very complex and lengthy ICD, and would be unlikely to fully satisfy anyone. This is where in the option described in the next section comes in.

Figure 15: Prometheus Block 2 atop the Test Docking Station

Hosting Option 2: Flexible Interface

Many providers are, for the most part, looking for satellites to host an existing or nearly complete payload. Therefore, they have already chosen their payload-specific hardware and data formats, have different power requirements, and generally desire to have very different interaction with the satellite. Some payloads, may also have very complex support requirements

As was discussed in the previous sections of this paper, most of the Prometheus subsystems (radios, C&DH, ADCS) utilize a common digital template including the microprocessor and supervisor FPGA. The 51-pin hosted payload connector provides lines to the FPGAs on each of the subsystems. Therefore, given a custom interposer board and additional software within the subsystem, data could be communicated directly to and from any of these subsystems using virtually any standard or custom format. Also, since access to the batteries is available, a custom power rail, additional batteries for high short term power requirements, etc. are all options.

This option would require software and FPGA firmware work as well as hardware design of a custom interposer board. However, there would be no required hardware changes to the Prometheus satellite. For most payloads, it is believed that the required development will still be relatively small.

As our experience hosting various payloads grows, we envision the need to include the payload as part of the SV network. This requires including one of the Prometheus microprocessors on the interposer to participate on the network. This processor could "relay" commands and data to a specific payload or be used to perform processing on behalf of the payload.

Given this, the payload could be provided independent virtual circuits to/from the ground to facilitate direct communications between the payload developers and their payload. This allows payload functionality to be modified and improved without requiring modification and requalification of the base Prometheus flight code. This will also provide many of the capabilities inherent in the system library software, probably the most attractive of which will reliable software upload and reprogramming on orbit.

APPLYING THE APPROACH TO LARGER SATELLITES

The LANL Agile Space Team has been developing and demonstrating its approach to satellite development with CubeSats. Standardizing to a dispenser provides frequent, low cost, launch options for CubeSats. However, there are limits to what can be done in the CubeSat form factor. Principally, CubeSats lack the flexibility to support large apertures.

From working in the CubeSat form factor, the hardware developed for Prometheus fits within a volume of 10cm×10cm×17cm (standard 1.5U size). If a larger aperture was required to support a given mission, the C&DH, ADCS processor, telemetry radio(s), and power converters could all remain within this volume. Scaling would possibly be required for the battery storage, the solar panel area, and reaction wheel assembly. Each of these elements are particularly well suited to easy scaling given the current designs for Prometheus Block 2. It is therefore expected that if LANL were to apply its approach from Prometheus to a larger small satellite, that very good volume efficiency of a primary, large aperture, payload could be achieved.

REFERENCES

1 The CubeSat Program. (2014, February) CubeSat Design Specification Rev. 13. [Online]. http://www.cubesat.org/images/developers/cds_rev13_final2.pdf

2 (2011) LANL Launches Four Tiny Satellites into Space. [Online]. http://www.lanl.gov/science/NSS/issue1_2011/story2c.shtml

3 Ethan W. Mattox. (2014, April) Special Operations Takes the Fight to the High Ground. [Online]. http://www.thespacereview.com/article/2491/1

4 B. Wie, H. Weiss, and A. Arapostathis, "Quaternion Feedback Regulator for Spacecraft Eigenaxis Rotations," Journal of Guidance, Control, and Dynamics, vol. 12, no. 3, 1989.

5 GSFC, "General Environmental Verification Standard (GEVS)," Standard GSFC-STD-7000A, 2013.

6 John Michel et al., "A Portable Autonomous Ground Station to Support a Constellations of CubeSats," AIAA Small Sat, 2015.