Plug-and-Play Satellite (PnPSat)

Session: Missions and Applications
Long Paper

Don Fronterhouse[1]
Scientific Simulation, Inc.
2951 Marina Bay Dr. #130-306, League City, TX 77573
Jim Lyke[2]
Air Force Research Laboratory, Space Vehicles Directorate
3550 Aberdeen Ave SE, Kirtland AFB NM 87117-5776

E-mail:donald.fronterhouse@kirtland.af.mil, james.lyke@kirtland.af.mil

[Abstract] The Air Force Research Laboratory's (AFRL) Space Vehicles Directorate has initiated a program to create the first satellite based entirely on the principles of plug-and-play as represented by the Space Plug-and-play Avionics (SPA) approach. Unlike other satellites, PnPSat is designed to be constructed extremely rapidly, based on design descriptions that can be eventually produced automatically from a push-button tool flow. The plug-and-play satellite (PnPSat) employs modular components, from the structural panels to the guidance and health/status devices, taking full advantage of the self-describing mechanisms inherent in the SPA approach. Panels contain SpaceWire routers and multiple connection sockets to support the arbitrary arrangement of spacecraft components on the panels and the connections between panels. In most regards, PnPSat reduces the integration of a satellite to a simplified assembly process, analogous to the assembly of components on a personal computer in which components are enumerated by the host as they are added. Since all components are based on the same, self-describing interface, the proliferation of disparate simulators and emulators are sharply reduced, and a unified "test bypass" mechanism is provided to facilitate hardware-in-the-loop simulation of a single component or the entire satellite at any point during integration of the system. This talk will describe the background and status of the PnPSat development program.

1. Introduction

To accelerate the development of complex space systems, fundamentally new approaches will be required over those used in conventional spacecraft today. The idea of "plug-and-play" (PnP) suggests an ease of integration is possible, and indeed the concept has found popular use in terrestrial systems. Evolution in aerospace systems has been decidedly more measured, cautious, and incremental. To achieve the benefits of PnP, it is necessary to "invest silicon" into the interfaces of components within a system, not to improve the performance of these components, but rather to improve the ability to more quickly make use of them. Given the considerable expense of radiation-hardened spacecraft components, the notion of diverting machine cycles into interfaces and away from primary performance may seem like profligacy. Or not, perhaps, in considering that the majority of cost in a space system can be traced to labor, and intelligent interfaces can reduce much of the labor associated with component / system integration.

---

[1] Chief Scientist.
[2] Technical Advisor, Space Electronics Branch, AFRL/VSSE, senior member.

The AFRL's Space Vehicles Directorate began a research program to understand the complexity of space systems and how, through technology, it would be possible to create them much faster. The core ideas of our approach, referred to as "Space Plug-and-play Avionics" (SPA), employ intelligent interfaces to accelerate component integration. In principle, SPA components are analogous to the USB components of a personal computer in that they embody the concepts of self-organization (networks are formed by simply plugging components together), self-description (through the use of electronic datasheets embedded in components), simplified connections to devices, and interchangeability of devices. SPA software concepts (namely the "Satellite Data Model" [4]) encourage the creation of PnP "awareness," striving in effect to make operational flight programs more insular to differences in components. A companion set of "push-button toolflow" concepts [5] promote the rapid construction of spacecraft through a set of ideas very analogous to menu-driven consumer product purchasing or more approximately to electronic design automation, in which ideas are germinated in a "capture" process and evolved into a buildable specification.

The ideas of SPA sidestep decades of "incrementalism" in favor of an architectural "clean sheet approach." Though applicable on a limited scale (and some limited demonstrations are under development), the ideas are more compelling when they can be conducted on the scale of an entire platform. Spacecraft are built to perform missions, and missions are driven by payloads. Unfortunately, many payloads, even in new procurements today, are based on non-PnP legacy interfaces. Our early attempts to create a PnP platform were met with strong resistance, as replacing a legacy payload interface was viewed as inevitably adding cost, complexity, and risk. Resolving this particular form of the "chicken vs. egg" dilemma could only be resolved through a new experimental development, designed as a Plug-and-play (PnP) satellite (PnPSat) from first principles.

This paper is organized as follows. In the next section, we review relevant background technology and events shaping the emergence of the PnPSat. The following section will describe the philosophy behind PnPSat, the mission template around which it has been cast, and the management approach for the project. The next section will discuss the spacecraft subsystems, and finally we will review the status and future plans for PnPSat and follow-on activities.

2. Background

AFRL's Space Vehicles Directorate has conducted several studies showing the confluence of a number of space electronics technologies are enabling to the objectives of creating a spacecraft rapidly, including microsystems (the combination of microelectronics, advanced packaging, and microelectromechanical systems), high-performance computing on orbit (HPCOO), and reconfigurable systems approaches (such as field programmable gate arrays, adaptive wiring, and software-definable radio concepts). One of the technologies involved the notion of "peel-and-stick" or appliqué sensor modules (the idea was not strictly limited to sensors, but the term "stuck"). The sentiments of "peel-and-stick" components evolved an approach referred to as Space Plug-and-play (PnP) Avionics (SPA) [1]. In the SPA concept, complex components are encapsulated with simple but intelligent standard electrical interfaces, such as USB (SPA-U) and SpaceWire (SPA-S) [3]. The principles of the SPA approach are summarized as follows:

1. Component Physical / Functional Encapsulation

The concept of encapsulation is important, as it serves to hide complexity within modular compartments, presenting inasmuch as possible an apparently clean and simple interface. In fact, much of the "magic" of PnP occurs below the surface. SPA components, for example, carry their own descriptions, referred to as XML-based electronic datasheets (xTEDS). As it is not normally necessary for a personal computer user to be concerned with the inner workings of common components (i.e., mice and keyboards), the xTEDS mechanism in SPA makes it possible for components to carry their own documentation.

2. Self-Forming Networks

Self-describing components can be used to automatically construct networks. In SPA, devices are "endpoints," connected together through hubs (SPA-U) or routers (SPA-S), and the structure of the network is induced through assembly and automatically inferred by the system. The "endpoints" range from traditional bus components, such as gyros and reaction wheels, to payload elements, such as cameras. Even spacecraft structures can be viewed as components, perhaps sub-networks of SPA endpoints and hubs/routers. The paradigm of a "machine-negotiated interface" was felt to be especially liberating for spacecraft developments, which have a notorious reputation for cost and schedule overruns. Reducing the need to "think" to first order allows system developers to concentrate on core challenges in developing a complex platform without being mired in the myriad details of simple components.

The supporting mechanisms within SPA to support self-forming networks are encompassed in hardware and software features. In hardware, endpoints ideally need only form a connected topology in which relative ordering is unimportant. This insensitivity to location in the network frees the system developer from worrying about where the network "needs" to place components, except for the "real-world" constraints that sensibly apply (i.e., such as the need to ensure that reaction wheels are placed in orthogonal/orthonormal geometric relation). The binding of geographic information can also be generated in principle automatically, derived from the placement of components in a PnP system. Power distribution is distributed, with much of the burden of switching being placed on SPA hubs/routers.

For that matter, a spacecraft need not have a central processing and power distribution functions, as it is common practice in spacecraft to have a centralized "command-and-data handling" (C&DH) element for processing and an "electrical power subsystem" (EPS) for power management and distribution. Eventually, through SPA, the artificial constraints induced by a fixed network structure can eliminated.

3. Plug-and-play "awareness"

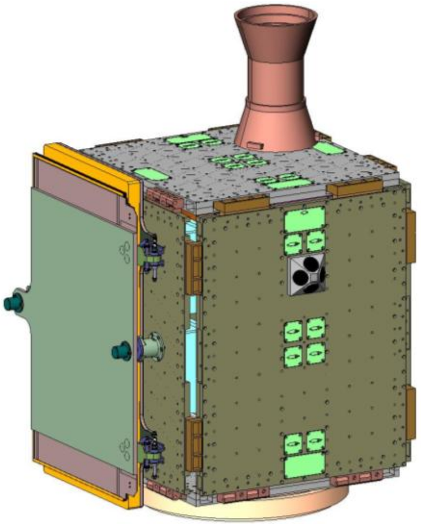
It is then only necessary for components to "understand" each other relative to the features or "services" they require of each other. In the development of SPA, this need gave rise to the concept of the Satellite Data Model (SDM) [4], which provides self-discovery and self-configuration capabilities to SPA. SDM provides a number of key mechanisms that organize a network and the devices it contains, in a manner similar to web services [6]. In this sense, even the most complex software applications can be viewed of as compositions of primitive transactions, based on communications between SPA elements using the "service descriptions" contained in the xTEDS. Software applications at the spacecraft level, to be effectively "plug-and-play," are written to harness the SDM infrastructure. One interesting side effect of this constraint is that "pieces of software" also typically contain xTEDS. Similarly, from the view of hardware components in a spacecraft, SPA devices (endpoints, hubs, and routers) must be developed to enable SDM to organize the spacecraft network automatically. For example, in the SpaceWire-based form of SPA (SPA-S), this requirement gave rise to a need to define a PnP protocol, whereas the automatic enumeration mechanisms already exist in the USB form of SPA (SPA-U).

4. Hardware-in-the-loop simulation (HWILS) and test bypass

A powerful concept in SPA, though not directly related to "plug-and-play," involves the ability to support the injection of synthetic information or instrumentation of data from components. In the simple example of a SPA thermometer, it is possible to substitute the ambient temperature values generated by the thermometer with a desired control value in a way that does not perturb the existing SPA network. Conceptually, the ability to insert or examine the state of variables throughout a SPA network is similar to the boundary scan principles in JTAG interfaces [7]. When coupled with a simulation infrastructure, the test bypass facility of a network of SPA devices becomes an *in situ* hardware-in-the-loop system. The philosophy behind test bypass is that rapid system development benefits (if it does not in fact require) improved abilities to dry run part of all of the system and to expose hooks to improve test and debug in the event of inevitable irregularities that are like to occur in development.

3. The Need for PnPSat

To gain a better appreciation of the considerations driving the PnPSat approach, we review conventional wisdom and the emergence of smaller spacecraft, oriented on shorter-term tactical needs. PnPSat represents an extension of the ideas of small, tactical satellites and modular design approaches, with an infusion of technology concepts that aim to simplify and accelerate the construction of spacecraft. The pursuit has not been without obstacles, and these have given rise to the need to consider PnPSat as a "clean sheet" approach, an alternative to a protracted, incremental technology insertion strategy.



Space electronic systems have evolved gradually over nearly half a century, and new architectures have been seldom attempted. A contemporary spacecraft is typically a polyglot of legacy hardware ("it worked before") fastened to new hardware, often requiring extensive additional engineering for each interface. These are in effect like a complex hardware/software "glue," applied liberally to achieve the desired effect. Even when a program offers a stubborn insistence on the use of an interface standard (popular choices include MIL-STD-1553 and RS-422), the likelihood that any two independently developed components would work in the same network without significant additional work is highly unlikely. Components, even if their interface designs are simple, are not typically designed to "explain themselves" to a system, but require significant human effort to reconcile. Spacecraft are not designed to accommodate unknowns or late changes in a system's design. Wiring harnesses are painstakingly defined to implement specific configurations. For these and many other reasons, the engineering of systems may be robust at one level (designed to operate reliably in a harsh environment), but they are at the same time fragile to change. And the longer it takes to develop a system, the more likely indeed it will be necessary to change the system in some way. As such, typical spacecraft are very expensive and take a long time to develop.

Small satellites, in principle, are believed to be less expensive, quicker to develop, and faster to checkout on orbit and bring online to satellite operators [8]. Small satellites remain a controversial proposition in military space. Some equate "small" with "responsive", but there is a conventional wisdom that suggests that capability scales with larger spacecraft.

Small spacecraft are formed in much the same way as larger spacecraft, subject to the same complexities and integration challenges, albeit at a smaller scale. The challenges might be met decisively through the use of SPA, but the level of commitment has been too different perhaps from that dictated by conventional wisdom to attempt. Breaking this cycle, we ultimately felt, would not occur until someone attempted a PnPSat.

4. The PnPSat Spacecraft Approach and Developmental Philosophy

And attempted we have. PnPSat represents the first spacecraft of its kind, not from outward appearance but from first principles as platform based on a self-organized network of self-describing components. It is modular, but the application of modular approaches in spacecraft is not a new concept. PnPSat can be viewed as the combination of modularity and complexity hiding. Most of its wiring harness will be invisible, recessed within panels.

PnPSat is a pure science and technology experiment to establish the necessary technologies to implement an evolving breed of software defined systems. Over the last several years we have touched every aspect of satellite design and construction, as well as test and operation to find those areas that inhibit the six-day spacecraft. What we found is that we must simplify the interfaces by hiding complexity. In PnPSat we are applying the principles of plug-and-pay to the mechanical, electrical and software interfaces.

What is a plug-and-play satellite? It is a modular satellite with open standards and interfaces, self describing components, and an auto-configuring system. This results in system integration and testing tasks that can be automated and are themselves simplified. Modular spacecraft structures also allow components to be mounted either on the inside or outside on regular grids. We are currently using a 5 cm x 5 cm grid. The system also employs modular flight software that is both easy to maintain and can be reused for various satellites as well as is intrinsically autonomous. Our goal is to have a satellite capable of maintaining its own health and status and only needs to talk to the ground by exception and for user tasking. High-performance-computing-on-orbit (HPCOO) provides gigaflops of processing to the user to support both the autonomy and on orbit processing of sensor data. We want to be able to provide to the user not only raw data as appropriate but also processed information. We also are working on tactical user interfaces that allow the user to task a satellite based upon that satellite's capabilities. We are working with experimenters to develop plug-and-play experiments. Since this is a science and technology satellite, we use experiments as payloads. Distributed power systems support plugging in a battery on one panel and solar arrays on another. Main bus power and charging grids are distributed throughout the spacecraft allowing access to the main power grid from anyplace on the spacecraft. This access is protected with circuit breakers in case something goes wrong. We are also investigating plug-and-play launch vehicle interfaces.

4.1 Requirements

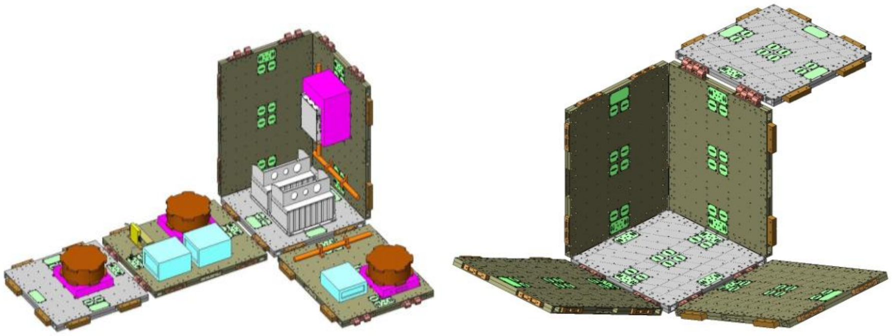
PnPSat requirements fall into three basic categories: the overall Responsive Space program requirements, the PnPSat program requirements, and the primary system capabilities that need to be demonstrated. The Responsive Space program requirements include demonstration of the viability and maturity of a modular plug-and-play architecture. It is important to be able to transition TRL-6 technologies to other satellite programs.

The primary system capabilities include being able to demonstrate rapid design, assembly, and test. Of course, we must be able to demonstrate modular plug-and-play, including both SPA and the Satellite Data Model. We must also demonstrate distributed systems including power, thermal, computing, and control. On the software side, we must be able to demonstrate robust autonomy including both dynamic schedules and activities.

4.2 PnPSat Architecture

One way to look at the PnPSat architecture is to break the spacecraft into three basic parts. First, the basic bones of the spacecraft upon which all components are attached. This includes the spacecraft structure, the power grids (both main and charging), the SPA infrastructure, and thermal control. Second, we add components that provide robust performance including the autonomous flight software; the quantity of high-performance computing; power generation and storage; guidance, navigation, and control components; and the communications radios for both tactical and TT&C. Finally, we add the mission sensors that provide customization for warfighter needs. From the perspective of building and testing the satellite, we must consider assembly, integration, and test; the ground systems; and the launch systems.

The PnPSat structure features modular panels to support quick assembly and the flexibility to mount components in multiple places. There are standard plug-and-play mechanical and electrical interfaces that can accommodate 48 experiments, and the components are located on either the interior or exterior surfaces. A tactical satellite requires approximately 25 to 28 components, which provides us with sufficient flexibility to mount the components based upon mass, thermal, power, and FOV requirements, among others. Electronics infrastructure and harnessing is recessed within each panel to increase available footprint and volume for the plug-and-play components and experiments. Locking hinge joints allow panels to rotate about the



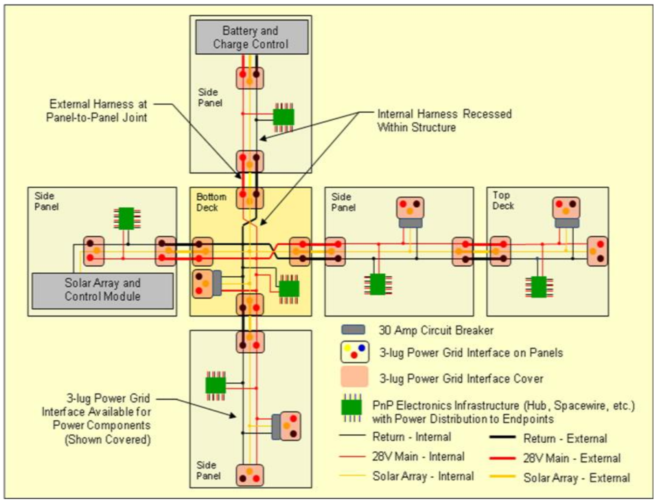
hinge line for easy access to the interior. Inter-panel jumpers, which harnesses across joints, allows the plug-and-play electrical network to remain intact throughout assembly, integration, and test. This means that we can determine if a component is working as it is assembled on the spacecraft. Currently, the panels are machined from 6061-T6 aluminum. The current structure is 51 x 51 x 61.2 cm and weighs 34.7 kg excluding the launch vehicle adapter.

One of the advantages of the folding PnPSat concept is that it can be changed easily to various configurations to support requirements for different stages of the project. At first it can be opened up into a flat configuration for internal components to be mounted and tested. Then it can be closed, while still active, for the external components to be mounted and tested. Panel to panel joints are pinned to allow panels to be rotated from the horizontal flat to vertical folded configuration. Then securing the joints with bolts provides for a rigid structure. Individual panels or sets of panels can be integrated and tested in parallel.

One of the modularity keys is to have a standard simple mechanical interface between the components and the structure. We have established a simple, standard mechanical interface to increase the flexibility and to speed integration. We have initially selected a 5 x 5 cm grid pattern that goes completely across the internal and external surfaces of all panels. The holes are threaded to support #8-32 fasteners. The hope is that eventually new components and experiments will be designed to accommodate this interface. In the meantime, existing components can be integrated with a simple adapter plate. This is the approach we are using on PnPSat to match legacy components to the modular structure.

The SPA electronics infrastructure is recessed within the interior of each panel including boards and inter-board harnessing. The power and data services provided to each of the eight SPA endpoints on each panel are handled by the robust hub. Panels are networked together, including power and data using the inter-panel harnessing. Once the SPA infrastructure has been installed and tested the panel halves are bolted together to form an EMI tight enclosure.

Each of the eight SPA endpoints has a standard electrical interface for components and experiments. For PnPSat the standard electrical connector is a 25-pin micro-D containing data (both Spacewire and USB), power (up to 4.5A @ 28v), time synchronization pulse, test bypass interface, and single point ground. Endpoints can be located on either the interior or exterior surface of the panel. Batteries, solar arrays, and power supplies have access to the power grids through 2-lug interfaces.



4.3 PnPSat Components

There are 25 PnPSat components plugged onto the structure. These include two coarse sun sensor assemblies, three reaction wheels, three magnetic torque rods, a fine sun sensor, a magnetometer, two batteries, FITS solar array, GPS radio, two packages of HPCOO processors, an Intelligent Data Store, and a TT&C radio. We believe all components that plug onto the structure should be plug-and-play. Our initial studies have shown that by recessing the electrical infrastructure and harnessing inside the panels, we significantly increased flexibility for component and experiment mounting.

To enable a plug-and-play power system, the bus power grid is composed of two separate grids: the main power grid, and the battery charging grid. These grids extend across all of the panels, allowing batteries, and the solar arrays to be connected to their grids from anywhere on the bus. High power components can gain access to the main power grid via 30 amp circuit breakers. The battery charge control electronics and the solar array controller are also SPA components. By separating the charging and main power grids, we enable a Phoenix mode, where even if we disconnect the main power grid due to low battery charge, we can still use opportunistic photons to charge the batteries. After the batteries reach sufficient charge, the battery and charge control electronics reconnect the battery to the main power grid and the satellite reboots.

The SPA infrastructure consists of the Appliqué Sensor Interface Module (ASIM), robust hub, hardware in the loop router (for ground testing only), the SpaceWire router, and the high power circuit breakers. The ASIM is used to interface legacy components to the SPA network. The ASIM also has two major functions. First, it is charged with the care and feeding of the attached component. Second, it presents a standard plug-and-play interface to the SPA network. The ASIM contains the xTEDS that defines the devices' data products, accepted commands, supported interfaces, and services provided. This allows each component to be self describing to the SPA data network. In addition, the ASIM provides a very accurate, real-time clock, and the hardware-in-the-loop test bypass interface.

One of the fundamental changes being implemented in PnPSat is the concept of a data-centric architecture. Traditional systems engineering is component centric, relying upon a detailed component interface control document (ICD) to enable system configuration. SPA enables us to focus more on the data rather than the details of the component. Data can be described, moving from the more fundamental to the more specific, as the basic physics, measurable quantities measured through a measurement process yield variables and qualifiers that we provide names and formats, and gather all of this up into the ICD. Now if we were able to agree upon the meaning of measurable quantities - for example, attitude or position or pressure or temperature - and place that in a Common Data Dictionary (CDD) for all to share and place the variable names and qualifiers and their formats in the xTEDS, we could then implement a standard SPA interface and get rid of the ICD. In this way, we have defined both a plug and a play interface, where that data interface is based upon a common standard (CDD) of what data means that is distributed to all, a standard data interface expressed in a standard language (XML), and the electrical interface based upon a common SPA standard.

The robust hub provides both a USB hub and endpoint power distribution and monitoring. Each SPA endpoint can be supplied up to 4.5 A @ 28 V protected by a circuit breaker. In addition, there is a current monitor on each endpoint with a soft breaker that can be set based upon the power required for that component as described in its xTEDS. In addition, the robust hub provides control of the high power circuit breakers. The robust hub uses an ASIM to provide power interfaces and control functionality, much like any other component.

4.4 HPCOO Components

PnPSat will be the first space implementation of the AFRL-developed Wafer Scale Signal Processor (WSSP) high-performance computer. There are six processors per chip, organized either as voted triplets with 6 MB of EDRAM each, that can detect and correct SEUs or as six independent processors with 2 MB of EDRAM each. In addition, each chip has 2 SpaceWire ports and FIFO interfaces. This processor has been completely synthesized and produced providing greater than 1 GFLOP per watt at 125 MHz. PnPSat will be flying eight of these chips, providing 24 GFLOPS of processing power.

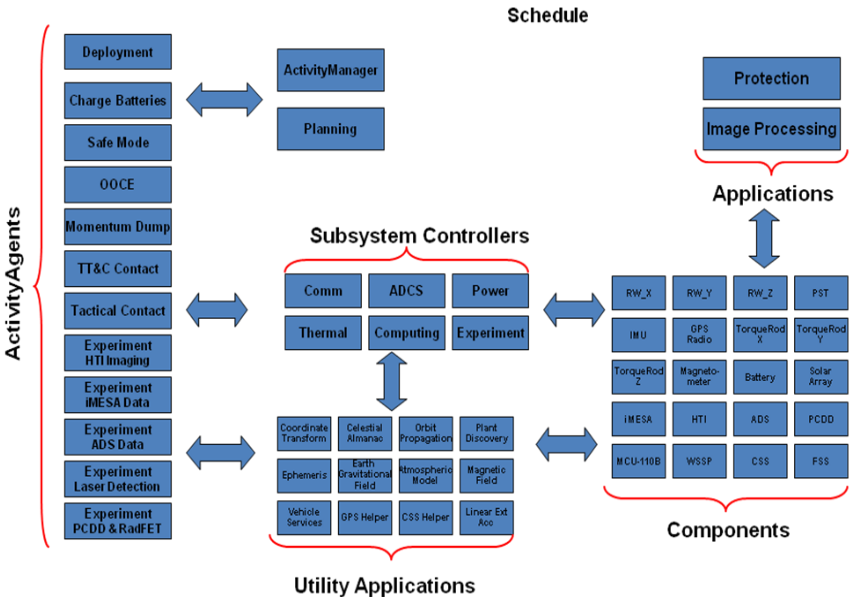
To provide on-orbit data storage, we are developing an Intelligent Data Store (IDS) that is fully SPA compliant. The IDS uses a Vertex 2 FPGA, with up to 4, 32-bit Microblaze processors and 512 MB (with a potential of 11 GB) of error corrected flash and 128 MB of error corrected RAM. The IDS resides on the SpaceWire high-speed data network and runs SDM applications and provides file storage and retrieval for system configuration data, application executables, and telemetry data.

4.5 Autonomous Flight Software

To assemble a spacecraft in two to three days means that we will not be able to write any custom software. Our focus on modularity includes the ability to develop software applications before the satellite mission or the specific components of the satellite are known. To facilitate the independent and concurrent development of hardware devices and software applications, we have developed a sideware application called the Satellite Data Model (SDM). SDM allows for the last-minute integration of independently developed hardware and software while supporting self configuration and self discovery. SDM is the play side of modular plug-and-play. It also provides a support model for fault tolerance to loss of devices, loss of software applications or services, and loss of SDM components.

There are five applications or managers that comprise the Satellite Data Model. A Processor Manager resides on each processor in the computing system and provides for the orderly execution of tasks on that processor. It is responsible for the underlying messaging services, and for dynamically selecting applications (tasks) to execute that are compatible with its resources. Tasks to be executed are posted to a Task List maintained by the Task Manager. The Processor Manager periodically reviews the current Task List and requests those that match its available resources. The Task Manager then assigns the task to one of the responding Processor Managers for execution. The Processor Manager also provides a heartbeat to the Task Manager to guard against processor failure. The Data Manager maintains a database of all xTEDS that have been registered by components and applications. The Data Manager provides a query and discovery mechanism whereby other applications can determine what data is available in the system, who provides it, and how to get a hold of it. The ability to be able to find single data elements within the data system is a key capability of a data-centric architecture. If the data network is composed of two or more sub-networks (for example SPA-U and SPA-S), a Sensor Manager is used to bridge the two networks. Finally, a Network Manager is used to discover the elements of the data network, their addresses, and in the case of SpaceWire the path routing between any two elements on the data network. We call the Satellite Data Model sideware because it is only used to discover the network and the data elements in it. Once a message has been subscribed, the data flows from producer to consumer as peers and SDM steps aside and does not get in the way.

There are two ways to look at the flight software architecture. The SDM discovery model provides for a flat architecture, where any application can get or provide data from/to any other application or from/to any component as necessary. This is an extremely flexible architecture, but more difficult to manage. We also have a more hierarchical architecture that is composed of controllers, agents, and managers that is conceptually easier to manage. It is important to remember that controllers do not own the devices they use to provide control. For example, the ADCS Controller does not own the reaction wheels, but does use them to control spacecraft attitude.



The PnPSat flight software core functionality is implemented as a group of autonomous activities. An activity is defined as a function that requires coordination of multiple subsystems and needs to be scheduled. Implementing flight software using SDM provides usability beyond just PnPSat. There are five basic categories of flight software in the hierarchical model. Subsystem controllers (for example Power, Communications, Computing, Thermal, ADCS, and Sensor) support both planning and commanding interfaces. System order is maintained by an Activity Manager that keeps the schedule and places activities to be executed in the schedule based upon time window and priority. We break priority into both a base priority associated with an activities importance to the satellite mission and an urgency that is time-dependent. For example, an activity to charge the batteries becomes more urgent the greater the depth of discharge. When it is time for an activity to be executed, the Activity Agents enables the associated Activity Agent. Activity Agents implement the basic activities of the satellite such as charging batteries, maintaining thermal control, collecting imagery, and safe mode. Activity Agents provide the heavy lifting to get things done and are required whenever more than one subsystem must be coordinated. Utility support applications such as coordinate transforms, orbit propagators, and celestial almanac, provide general-purpose support. Finally, there are the general purpose applications, such as satellite protection, image processing, etc. that are not associated with any specific activity.

Perhaps a PnPSat separation timeline will help to illuminate how all the various controllers, agents, and managers work together to bring the satellite up from cold at launch to a fully functional system. First, the separation switch closes as the satellite leaves the launch vehicle allowing the battery ASIM to provide battery power to the bus at which point the robust hubs boot providing power to the endpoints. As each ASIM boots, it provides control to its attached device. The WSSP ASIM boots the WSSP processors and loads and executes SDM. After network discovery by the Network Manager, the Task Manager is started and retrieves the initial

Task List from the IDS. The list includes the Subsystem Controllers, Activity Agents, Activity Manager, utility applications, etc. The Solar Array Activity Agent will place a deployment activity in the schedule via the Activity Manager and when executed by the Activity Manager will reduce tip off rates, deploy the solar arrays, and request ADCS go to sun point mode. Then, the normal activity agents take over and the satellite is up and running.

4.6 Building PnPSat

PnPSat will be built in the Responsive Space Testbed at AFRL's Space Vehicles Directorate, Kirtland Air Force Base, N.M. The schedule is very aggressive. We held a CDR last month and will have an AI&T Review by the end of this year. At that time, we will not have a full complement of flight components, but we will have tested with the engineering models. After AI&T, we will be taking the satellite apart, updating components, updating software, and testing to demonstrate that assembling and testing a semi-custom satellite in two to three days is achievable.

5. Acknowledgments

The authors would like to thank the AFRL's Space Vehicles Directorate and the Responsive Space thrust for its strong support of this effort.

6. References

---

[1] Lyke, J., Fronterhouse, D., Cannon, S., Lanza, D., and Byers, W. "Space Plug-and-play Avionics", proceedings of the AIAA 3rd Responsive Space Conference, Long Beach, CA, April 25-28, 2005.

[2] Lanza, D.; Lyke, J. ; Zetocha, P.; Fronterhouse, D.; and Melanson, D., "Responsive Space Through Adaptive Avionics", presented at 2nd AIAA Responsive Space Conference, Los Angeles, CA, April 19-22, 2004.

[3] Lyke, J., Cannon, S., Fronterhouse, D., Lanza, D., and Byers, T. "A Plug-and-play System for Spacecraft Components Based on the USB Standard", proceedings of the 19th Annual AIAA/USU Conference on Small Satellites, Logan, UT, 8-11 August, 2005.

[4] Sundberg, K., Cannon, S., Hospodarsky, and Fronterhouse, D., "The Satellite Data Model", International Conference on Embedded Systems and Applications (ESA'06), Las Vegas, NV, June 2006 (ISBN 1-60132-017-5/CSREA, Editor H.R. Arabnia).

[5] Strunce, R., Eckert, F., and Eddy, C., "Responsive Space's Spacecraft Design Tool (SDT)", Proceedings of the 4th AIAA Responsive Space Conference, Los Angeles, CA, 24-27 April 2006.

[6] Snell, J., Tidwell, D. and Kulchenko, P., *Programming Web Services with SOAP*, O'Reilly publishers, Sebastopol, CA, 2001.

[7] Joint Test Action Group (JTAG), "IEEE standard test access port and boundary-scan architecture", *IEEE Standard 1149.1*, 1990 and supplement *1149.1-2001*, 2001

[8] Singer, J. "U.S. Air Force Sees Operations Role for Small Satellite Platforms", Space News Business Report [online publication], 1 March 2004 (http://www.space.com/spacenews/archive04/platformsarch_030104.html, [cited 4 Apr 2007]).