



Autonomous PnP Flight Software

Kenneth Center
PnP Innovations

Don Fronterhouse
PnP Innovations



7th Responsive Space Conference
April 27–30, 2009
Los Angeles, CA

Autonomous PnP Flight Software

Kenneth B. Center
PnP Innovations, Inc.
2017 Yale Blvd. SE, Suite A, Albuquerque, NM 87106, USA (505) 503-1563
Ken.Center@PnpInnovations.com

Donald C. Fronterhouse
PnP Innovations, Inc.
2017 Yale Blvd. SE, Suite A, Albuquerque, NM 87106, USA (505) 503-1563
Don@PnpInnovations.com

ABSTRACT

Most space markets could benefit immensely from satellites that not only start their service lives as soon as possible, but can perform their on-orbit roles with minimal intervention from ground operations. The development of flight software techniques that addresses both of these issues has been ongoing at the Air Force Research Laboratories (AFRL) Responsive Space Testbed (RST) for a number of years. The Space Plug&play Avionics (SPA) standards define the means by which software (and hardware) components installed on a networked spacecraft bus deliver their data interfaces and self-organize into a coherent, functional system. A collection of modular, pre-validated software applications are maintained in a “virtual store room” and are called upon to configure tailored mission capabilities based upon operational needs. Mission “Activity Agents” encapsulate the logic and sequences necessary to accomplish satellite operations autonomously. These Activity Agents coordinate their execution using on-orbit planning resources and a priority-based scheduler. The PnP Sat mission was the first formal application of these techniques. In the course of the program a spectrum of software tools were authored to support the rapid design, prototyping, and testing of flight software modules for SPA-based satellites. AFRL is now populating their Operationally Responsive Space (ORS) assembly depot with an assortment of hardware and software components sufficient to build a reasonably diverse array of tactical small satellites. The tools being used, and the philosophies employed in their application, are ready for exposure to the larger space community.

KEYWORDS: Plug and Play (PnP), Space Plug&play Avionics (SPA), spacecraft autonomy, flight software, PnP Sat, Spacewire, rapid assembly, Responsive Space Testbed (RST), Operationally Responsive Space (ORS)

INTRODUCTION

PnP technology has revolutionized the consumer market for terrestrial computing. It allows even the least technology savvy user to add features and peripherals to desktop work environments to maximize their productivity without the need for a detailed understanding of the inner-working of the platform. PnP for space applications aims to accomplish similar goals, effectively allowing satellites built for specific mission applications to be rapidly assembled and tested by technicians from depot stock. Facilitating this goal requires a combination of well-crafted standards to assure interoperability of PnP components, and a set of tools to guide the selection of those parts. In this paper we discuss the former item, specifically with regards to the development of PnP software strategies that enables the creation of space systems from a collection of modules that form required data relationships based on query and subscription mechanisms. Most importantly, these modules need no modifications from mission to

mission. So long as the descriptions formalized in PnP standards documentation are heeded, interoperability is assured.

Historical Synopsis of PnP at AFRL

The Air Force Research Laboratory’s (AFRL) Space Vehicles Directorate, Kirtland Air Force Base, N.M., began an initiative to develop PnP standards for spacecraft in 1994. The effort commenced with an objective look at the areas of the spacecraft development process that inhibit rapid deployment of assets. Responsive access to space (e.g. the launch vehicle) is one of the greatest barriers, but a look at the process reveals that the lack of widely accepted standards for hardware and software interfaces is the cause of much of the customization required for development of a new mission. Even though the same hardware may be utilized on many generational missions, a particular implementation typically uses a specific collection of hardware - and those hardware

elements will be supported by avionics (typically interface cards in a chassis) that require the authoring of specific drivers to facilitate use of that hardware.

Space data systems and flight-side software in traditional missions adhere to no particular common implementation approach. As such, a particular mission will employ solutions that are optimized for the particular use case being considered. While a consistent approach may be applied within a family of satellites at a particular vendor's discretion, the combination of processing platform, data transport medium, and software interfacing method are not such that mission heritage is easily transitioned or leveraged for reuse on markedly dissimilar programs.

The lack of a common standard for testing also results in the need to formulate customized harnesses for every mission to gain the integrated test perspective (component-level and day-in-the-life) necessary to instill confidence in the performance of the spacecraft to launch. This is a very large portion of the development cycle for spacecraft. By extension, this also applies to the test support environment and ground station interfaces, which must generally be customized to the particular spacecraft.

AFRL sought to focus primarily on the interoperability and tools aspects of the problem when they embarked on the Space Plug&play Avionics (SPA) initiative. SPA defines a common standard for both hardware and software elements of a system. The physical SPA connection to a hardware device in the system consists of a defined data exchange mechanism, supply of power, and the distribution of timing. SPA also defines a common mechanical mounting approach for physical components that facilitates unambiguous assembly of a satellite from PnP parts on a modular structure (a 5x5cm bolt pattern). Incorporating the SPA infrastructure components within these panels allows much of the labor intensive integration of spacecraft support devices (processors, sensors, actuators, and payloads) to be reduced to a no-nonsense attachment and single pig-tail harness connection to the system.

Perhaps most important feature of the standard (particularly with respect to the primary subject matter of this paper) is the use of a descriptive electronic data sheet to relay the capabilities of a PnP component to the system. This Extensible Transduced Electronic Data Sheet (xTEDS) is effectively the Interface Control Document (ICD) describing the identity, role, characteristics, and data exchange features of the component. Keep in mind throughout this paper that one of the goals of SPA is to abstract hardware and software such that in the network-based system there is

no distinction between the two. The rationale for this will be discussed later.

The SPA standards were substantially matured during an initial Operations Concept (IOC) hosted by AFRL. In this exercise, the first generation core infrastructure components required to enable SPA were developed. This included Appliqué Sensor Interface Modules (ASIMs) to interface legacy components to the SPA standard, data routers to facilitate the transport of data between devices and processor-resident applications, and power hubs to distribute and monitor the delivery of 28V power to all SPA devices. This development phase also included the delivery of the initial version of the Satellite Data Model (SDM), the "sideware" that enforces the data aspects of the SPA standards. All of this work was integrated at the Responsive Space Testbed (RST) at Kirtland AFB.

A successful demonstration of the key SPA technology elements at the conclusion of the IOC led to the creation of a technology demonstration initiative. The PnP Sat program was funded by AFRL with the objective of leveraging the SPA technologies to build a satellite (see Figure 1 below) composed exclusively of SPA-based PnP technology elements.

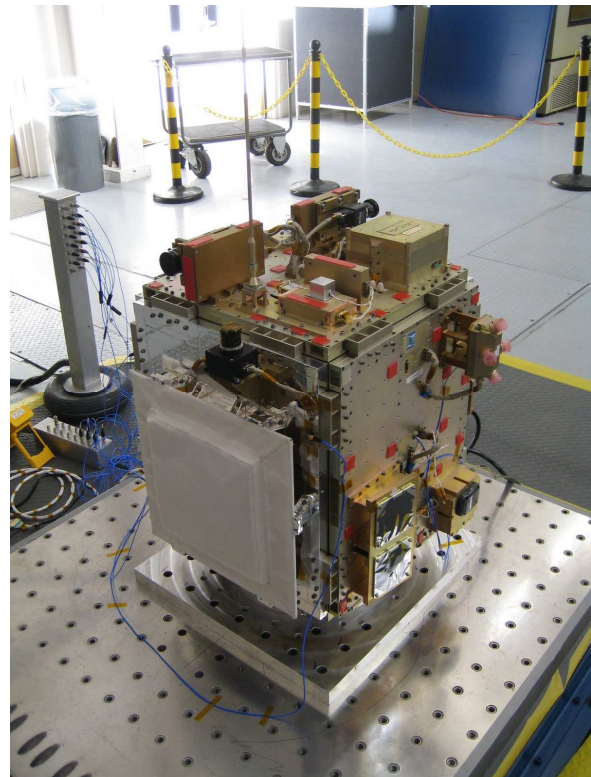


Figure 1. PnP Sat – The first satellite based completely on SPA technology.

PNP SOFTWARE FUNDAMENTALS

Characteristics of Software for Responsive Space

Software has traditionally been one of the greatest barriers to rapid spacecraft development turns. The reason stems both from the fact that the software implementation and test cycle is time consuming and that that final testing of flight software is contingent upon delivery and assembly of hardware (or at least a responsibly robust mock-up of the ultimate hardware system).

SPA provides mitigation for both of these issues by offering a software system that encourages the construction of modular elements that can be developed, thoroughly tested, and ultimately certified for use in a SPA-based system without modification between missions. The model presumes the development of a collection of these modules that become part of a comprehensive PnP software library that can be drawn from to assemble functional capability given a set of top level mission objectives.

The evolving Operationally Responsive Space (ORS) office at Kirtland AFB is beginning to use and extend the processes involved in the rapid assembly and testing of satellites using the SPA standards. In large part, they are leveraging from the successes of the PnP Sat program. This includes the ability of front-end design tools to specify a collection of PnP components (both hardware and software) suitable to accomplish a specific set of mission objectives. Once this set of components is assembled (PnP devices placed on a hosting collection of SPA modular panels and flight software modules loaded onto SPA processors) the system is largely self-organizing.

This self-organizing capability is a critical characteristic of PnP. It allows a system to be built from modular parts without customization. This is necessary to meet the ambitious capability of collecting theater support needs and completing the deployment of space assets to address them within the timeframe of a week – a goal that AFRL and ORS hope to realize.

One premise that SPA espouses is that in order to achieve the level of interoperability required for rapid assembly and system robustness, discrete functional modules must not present as hardware or software, but simply be abstracted as “PnP Components.” Honestly it is irrelevant whether a component is physical or virtual, so long as its data is consistent with the ICD it presents. This philosophy also enables highly enabling test support tools which will be discussed later in following sections.

A networked solution is highly desirable for a PnP-based system. Doing so provides the opportunity for processing scalability, the implementation of data robustness features, and integrated test support. Figure 2 shows the way in which a SPA-based computing implementation can be composed of many networked processing nodes, all endpoints in a system capable of adapting to the software installed on the system and able to redistribute processing load if certain assets go offline (temporarily or permanently). AFRL has pursued a variety of options for computing nodes in their programs, including lightweight FPGA-based processors running Linux, “General Purpose” computing nodes based on heritage space-qualified CPUs, and forward-looking Digital Signal Processing (DSP) style solutions such as Rome Lab’s Floating Point Application Specific Processor chip (FPASP). Under the SPA standards, all processors network together transparently via a Spacewire-based network as illustrated in Figure 2.

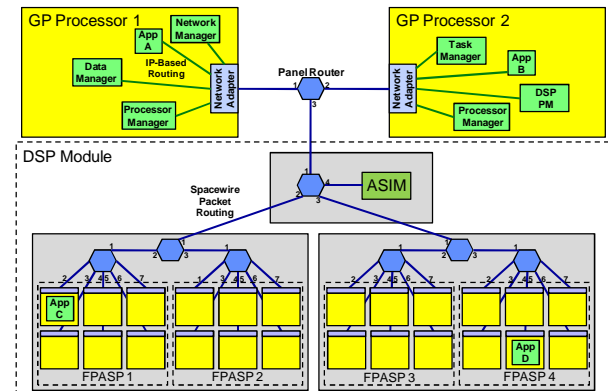


Figure 2. Two distinct processor technologies networked via the SPA standards

The Satellite Data Model

The cornerstone of being able to self-organize a SPA-based system relies on the ability to register the capabilities that exist in the system and facilitate the interconnection of data providers and consumers. The Satellite Data Model (SDM) provides this service. SDM consists of a Data manager (DM), Task Manager (TM), Processor Manager (PM), and Network Manager (NM). We call SDM “sideware” because it provides a “brokering” between SPA components – facilitating data relationships and then getting out of the way so that lightweight, efficient point-to-point IP message-based communication can occur between them.

The DM is the SDM component that maintains a registry of all system capability. When a SPA component joins the system, it delivers its xTEDS to the DM. Each xTEDS is an XML-based text stream that

states the identity of the component, its high level characteristics, and then lists a description of the various data interface that the component supports. This includes commands and data messages, each of which may contain references to variables that are named, typed, and tagged with supporting metadata. Upon receipt, the DM parses each xTEDS, building a database structure that relates aspects of the registered data to the component that can provide it.

Any component that joins the system may satisfy its own consumer data needs by querying the DM. Querying can be based on any of the descriptions or metadata characteristics acquired during the registration activity. The nominal means of discovering data to use is to query for a data “kind”. Data kinds are maintained in a SPA Common Data Dictionary (CDD) which formalizes the content to be included when authoring an xTEDS for a new PnP component. An accompanying xTEDS XML Schema enforces syntax. If discipline is applied when creating xTEDS (a component of the certification process), the self-organizing features of SPA are assured. The DM will return all query matches to the component making the request. Matches consist of a list of messages that satisfy the query and the location of the data item within each message. The characteristics associated with the matched data item and its carrier message are accessible as well, so metadata such as the type, format, and frequency of delivery of the message are readily available to support use of the interface. Once suitable data sources have been identified, it is then the responsibility of that component to manage the responses. The typical response is to subscribe to the messages returned in the query (an action also facilitated by the DM). Subsequent communications between the producer and consumer are then point-to-point. Handling of multiple matches is the responsibility of the consumer. A typical approach involves determining the “ideal” source using metadata embedded in the xTEDS, then using the remaining data streams as “backups” to facilitate instantaneous fallback if the primary data source is interrupted. The DM dynamically manages de-registration of components as well, notifying all subscribers if a component exits service.

The SDM TM handles the distribution and oversight of applications across all processors discovered in the PnP system. The TM receives processor status from all Processor Managers that have registered in the system, including available memory and processing cycles. The TM uses this feedback to determine where to load applications in the networked data system. Application binaries are loaded across the network to a receiving processor through its Processor Manager. The TM initially works from a list of applications to load at

system initialization, but once the default configuration has been brought up, applications may be started or terminated using the TM control interface. The TM also collects heartbeats from the Processor Managers in the system. If a PM becomes unresponsive, the TM will redistribute the applications that were running on that processor to other computing nodes in the system.

The PM brokers the loading of applications on each of the computing resources within the PnP data network. The PM facilitates the local loading and execution of application binaries as directed by the SDM Task Manager. The operational state of each application is monitored, both individually and collectively, so that a view of memory and processing load can be reported to the Task Manager for overall system data resource balancing. A heartbeat is expected from each locally running application. If that heartbeat ceases to arrive (or if a signal fault is detected) the PM will report the anomaly to the Task Manager for executive resolution.

The NM is an application that performs the discovery of PnP endpoints on subnets attached to a processing node. This discovery leads to the formulation of a network topology for the subnet that can then be used by a SPA Network Adapter to bridge IP messages to destination endpoints on the subnet. The NM delivers subnet routing information to all nodes upon initial discovery, and dynamically responds to subsequent registration and deregistration of components.

Built in Robustness

It has already been mentioned in previous sections that there is an element of dynamic behavior inherent in the SDM. This is primarily in place to facilitate system robustness. SPA facilitates the ability of the system to adapt to short-term lapses in data or self-heal when sources disappear altogether. Figure 3 illustrates two cases that showcase these characteristics.

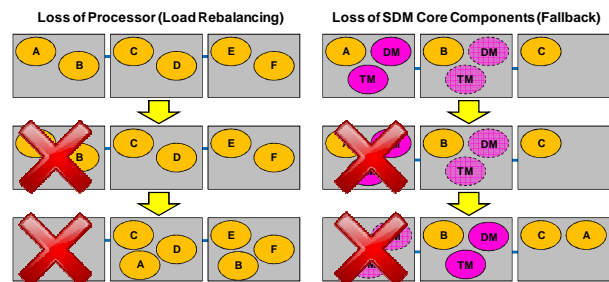


Figure 3. SDM Data Robustness Features

The leftmost figure illustrates the role that SDM is designed to play in compensating for the loss of a processing component. If a node becomes unresponsive (determined by failure of that node’s PM to exchange a

heartbeat with the TM), the TM immediately loads the applications that were hosted by the defunct node onto the remaining processors. Although the general case involving an unresponsive processor must be handled for mission flight software modules, the same fault tolerance must be demonstrated for the SDM core applications themselves. In the event that a processing node hosting the DM or TM becomes unresponsive, the responsibilities of these core applications must be transitioned to fallback copies (as indicated in the rightmost graphic).

ENABLING BY-PRODUCTS OF SPA

Some of the most compelling benefits afforded by the SPA PnP data features have proved to be areas not originally conceived when the standards were set. Although rapid configuration of a satellite system and the tools to support integrated testing were originally envisioned, several of the capabilities that have evolved through the research program have yielded benefits worthy of mention.

Flight Software In the Loop

In the PnP Sat effort, many aspects of SPA development were being conducted in parallel. This included the creation of flight software at the same time that infrastructure hardware and endpoint support components were being designed and fabricated. In order to meet the schedule, a capability was assembled to allow the many software contributors (including Utah State University's Spacecraft Dynamics Laboratory, Design_Net Engineering, Microcosm, Interface and Control Systems, and AFRL staff) to test their flight software against a stand-in for the flight system. Fortunately, SPA facilitated an easy solution to this need. Because of the data network paradigm, Star Technologies Corporation was able to create an environment based on their Satellite Design Tool (SDT) to expose virtual behavioral models of spacecraft support devices. This virtual device bridge allows those models to be presented to the data system according to the same SPA data standard as the actual ASIM supported hardware. This capability, termed Flight Software In the Loop (FSWIL), was instrumental in jump starting software development for the program. Figure 4 illustrates the way in which the virtual devices derive their responses from the six-degree-of-freedom engine in the SDT environment and present (via SDM messaging) to applications running on either the local simulation workstation or over Ethernet to flight-like development processors.

Using a central configuration managed repository, developers were able to stay synchronized with the work of other parties, including the developers of the

device code being developed for use on the hardware ASIMs. The vast majority of flight code development was conducted within this infrastructure. Once hardware was available for test in the RST, the migration from FSWIL to PnP Sat was relatively seamless.

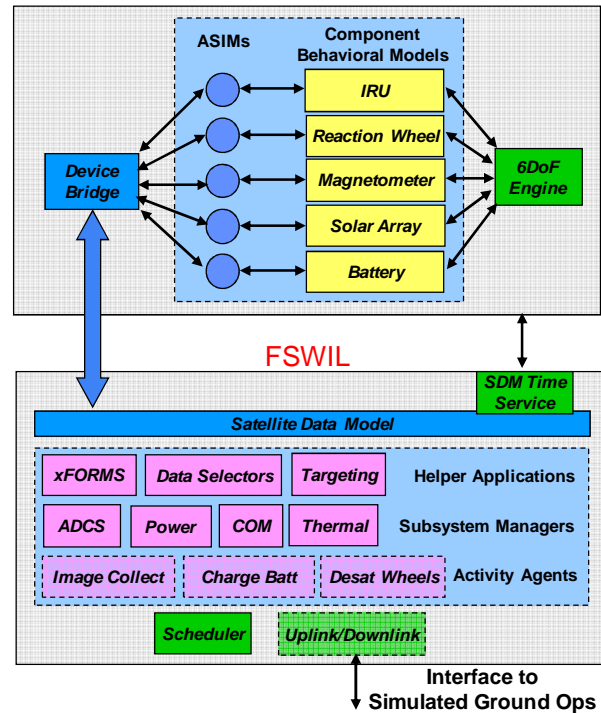


Figure 4. SPA-facilitated Flight Software in the Loop

FSWIL's benefit going forward in the Chileworks process is not only to support early development and wring-out of PnP code modules, but also to allow preliminary verification of flight software's ability to meet mission expectation when coupled with a collection of modeled PnP hardware elements. When reasonable confidence is gained in FSWIL testing, the logical continuation is to actually assemble a satellite configuration from SPA components and test again with the performance of the flight processors and network latencies in place.

Hardware In the Loop

The ability to rapidly assemble and test a satellite system composed of PnP technology was built in to the SPA standard when it was originally specified. The SPA interconnect standard includes a TestBypass connection that allows data from external coordinated simulation to be directly injected into the data variable registers maintained by each ASIM. When utilized, every endpoint device delivers data to the flight system consistent with the operational space environment

without any disruption of the primary data network. This is classical Hardware In the Loop (HWIL) testing, but using a standardized mechanism that is built into the system so that custom solutions need not be developed on a mission-to-mission basis. TestBypass utilizes RS-422 to deliver an addressed protocol to each panel that is routed to each ASIM-hosted endpoint via a router embedded in the SPA modular panel. The same device virtual models that are utilized for FSWIL provide the register-level data items for HWIL, therefore enforcing consistency (and traceability) from the early development phase to the flight verification phase.

Although the HWIL capability utilized for PnPSat relied upon an Ethernet to RS-422 rack-mounted breakout box, PnP Innovations and Star Technologies are jointly developing a PCIe card and breakout box solution that may be installed in a Windows-based workstation to provide benchtop HWIL capabilities at developer sites for early assessment of SPA compliant hardware in unit-level testing or against SPA modular software running on the workstation.

Test Harnesses and Ground Station Operations

Almost certainly the most overlooked benefit of the SPA PnP paradigm is to the test and operations environments. Because the assembled system of SPA components registers its capabilities with the SDM DM, any external data fixture may query for the complete inventory of components (or a subset) and formulate an understanding of the possible interactions without having any a priori knowledge of the satellite. Interface and Control Systems capitalized on this feature by developing auto-generation of command and telemetry screens to their Remote Information Management System (RIMS). Star Technologies has added mission visualization to the SPA palette by demonstrating the ability to subscribe to spacecraft state data in flight software and tie that data to their 3D viewer – rendering the spacecraft position and attitude and overlaying key status data items as vectors (such as the line of site to the sun in spacecraft body frame) or textual heads-up-display style information. The screenshot shown in Figure 5 depicts a PnPSat derivative concept satellite visualized in the SDT framework using Data Browser obtained state data from flighty software.

Use of the StarTech data Browser similarly supports a warfighter interface to spacecraft capabilities by querying for “tactical” interfaces and generating screens that can be used to task the satellite and scrutinize the resulting data products.

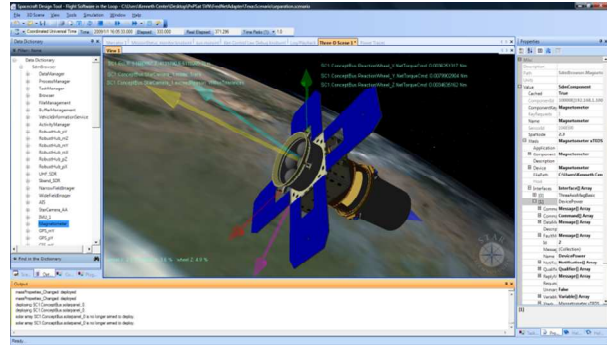


Figure 5. SDT Data Browser used to visualize state data from flight software

These development efforts are proof-positive that SPA facilitates significant advantages and flexibility in the area of support tools. The ORS Chileworks effort is capitalizing on these tools and soliciting ideas for others. Ultimately, an extremely robust process toolchain built around SPA is expected for support of design, production and operation of assets built from SPA technologies.

PNPSAT FLIGHT SOFTWARE

The PnPSat initiative provided a perfect opportunity to validate the SPA standards in a real-world scenario. Although much work had been accomplished toward refining the standards in strategic working groups during the IOC exercise, PnPSat provided a tangible use case that led to much advancement in the paradigm, and as mentioned in previous sections also resulted in the development of several critical process tools.

Architecture

Initially, SPA software was envisioned to be completely self-configuring. The presumption was that software modules could be written to be completely agnostic to spacecraft-centric issues and ultimately adaptable to any newly introduced module. There were no standard naming conventions – simply standards that enforced the organization and construction of the information contained within an xTEDS. This notion proved untenable as developers began to utilize the SDM code base to author the first-generation flight software baseline. It was soon discovered that there was a logical hierarchy to the application modules that were being developed. They clearly fell into distinct layers that map to roles reminiscent of traditional spacecraft data systems. These layers are visible in Figure 6. The properties of each, as well as the rationale for defining them as such, will be discussed in subsequent sections.

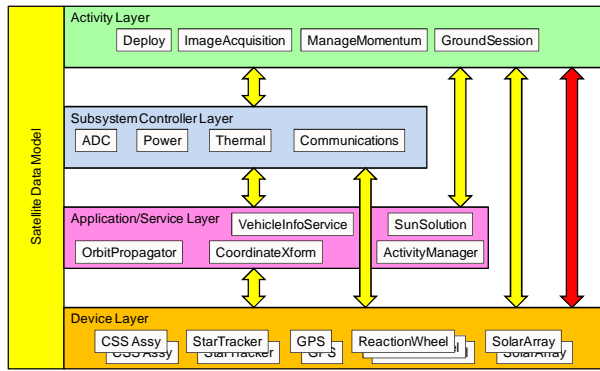


Figure 6. SPA software architecture layers

Service Classes

While there was not originally a defined “software architecture” for PnP space missions, one began to emerge through the course of the program. A distributed, network-based approach to software practically begs for the compartmentalization of system services - resources that are so commonly needed in the system that deserve to be encapsulated as stand-alone modules. As such, these modules came to constitute a “core software load” that is required to support any space mission of higher-level relevance.

The most essential of these applications is the *VehicleInformationService*, whose role is to provide a clearing-house for satellite-specific or mission-specific information that is otherwise unobtainable from the SPA components themselves. This extends to information that is not included within the xTEDS. xTEDS themselves are static and do not, for example, include the mounted location and orientation of a component on the spacecraft. This is of course critical information for interpreting sensor information, which must very frequently be resolved into other reference frames (most notably the spacecraft’s) to be useful. The *VehicleInformationService* application also addresses the need to assign a unique identity to installed components (that are identical according to their xTEDS). This is accomplished by reading a system configuration file that relates “Keynames” to an installed location. This is helpful not just for proper interpretation of data from similar devices, but for a ground operator who requires a friendly moniker to hold on to when understanding the interfaces presented on the ground.

Helper Applications

Occupying the same layer in the architecture are “helper applications,” which serve the purpose of converting device-level data to system-relevant data. An example from the architecture layer diagram of Figure 6, is the *SunSolution* application, which

subscribes to all instances of course sun sensor cosine angle data discovered in the system and calculates the sun line-of-sight vector in the spacecraft’s body reference frame. Many examples of helper applications exist in the PnP system. Most exist to bridge the gap between raw PnP devices and mission software. In the end-vision they are the “filler” applications identified in the satellite design process when desired capabilities must be mapped to a collection of physical hardware components.

Subsystem Controllers

There are common high-level functional constructs that exist in all satellite systems. It is not coincidence that satellite designers consider system trades along subsystem boundaries. Viewing the satellite system along the lines of these engineering domain areas (power, thermal, attitude determination and control (ADC), communications, computing) is not just logical, but well understood by engineers and operators. A vast array of analysis tools and an extensive knowledge exists to support systems organized along these boundaries. It was natural, then, to define “subsystem controllers” in the PnP architecture. Each subsystem controller acts to aggregate the installed SPA devices and available Helper Applications to present a uniform view of that subsystem through a standardized SPA interface. The intent is that regardless of the implementation, any ground operations center, tactical user, or autonomy implementation (discussed below) should be able to uniformly access status and issue directives. For PnP, the Subsystem Controller interfaces were taken one step further – by adding subsystem-oriented planning. In order to allow some level of on-orbit planning to be implemented, each controller implements a primitive means of determining whether or not operations (confined to reasonably short look-ahead windows) are viable given the current state of the spacecraft. As an example, the ADC Controller may be given a directive in “planning” mode. If the maneuver can be accomplished, the controller will respond with an indication of the time required to complete the request and the projected energy to be consumed by the ADC sensors and actuators. This information may then be used in a dialogue with the Power Subsystem Controller to determine whether the combination of collectable power and stored energy can support the maneuver. The interfaces are admittedly crude in the initial implementation, but they do facilitate the demonstration of a reactive satellite autonomy as described in the next section.

Activity Agents – SPA Autonomy

Autonomy has many benefits to tactical space operations, not the least of which is the ability to

transition a portion of the typical command and control aspects of satellite operation from central ground operations to the platform itself. As flight-worthy processors become more capable and it is possible to have greater degrees of high-power computing on-orbit, a natural course is to interact with the satellite along the lines of tactical tasking or objectives. This was the paradigm sought for PnP/Sat, in which the various mission roles of the satellite (both routine and tactical) were identified and assigned to distinct software application modules termed “Activity Agents.”

The strategy was to utilize the interfaces exposed by each of the Subsystem Controllers (as described in the section above) to provide a mechanism to accomplish on-orbit timing of these activities. Interface and Control Systems (ICS) developed the Activity Manager application, which serves as the timeline management function on the flight side.

Activity Agents subscribe to system data in the same manner as any other application, but tend to work through Subsystem Controller exposed interfaces except in special cases. Each activity asserts its own interests by monitoring system status, using the subsystem planning interfaces to assess the viability of execution, and then working with Activity Manager interfaces to schedule insertion in the system timeline using an expressed priority. As an example, a routine activity such as *ChargeBatteries* compares the state of charge of the batteries against set thresholds and escalates priority according to the severity of discharge. The requested insertion priority allows the activity to effectively “compete” with other activities – such that if a high priority tasking is asserted by a ground operator, the lower priority activity may be pre-empted and forced to reschedule. In the *ChargeBatteries* example, the execution phase of the activity involves performing a dedicated pointing of the spacecraft’s solar arrays toward the sun to collect maximum power to restore battery charge.

Much more complicated “tactically-oriented” activities have also been developed. One such activity is *MonitorShips*, which utilizes Automatic Identification system (AIS) sensor data to demonstrate several operational scenarios based on the determination of ship locations. A user may express the desire to monitor the activities of vessels within defined geographic boundaries. This can be merely to count the instances that satisfy certain expressed constraints (such as speed or direction of travel) so that trends in activity can be autonomously monitored by the spacecraft and reported to the requestor. It is also possible to request imaging of ships meeting certain criteria. This feature exercises cross-queuing features of the PnP system, allowing discovered imaging devices to be recruited to provide a

service that may not have been the original intent of that device’s installation on the satellite.

The baseline configuration of PnP/Sat autonomy consists only of routine activities designed to characterize SPA reliability and gain flight heritage for the assembled hardware. These activities include *Deploy*, *ChargeBatteries*, *ManageMomentum*, and *Idle*. *Deploy* handles the stabilization of the satellite after launch vehicle tip-off and the sequenced deployment of all arrays and antennas. *ChargeBatteries* forces dedicated sun-pointing of the satellite based upon battery state of charge. *ManageMomentum* asserts dedicated de-saturation of the reaction wheels if they approach critical limits. The *Idle* activity maintains a default spacecraft state when all no other activities are running. For PnP/Sat, *Idle* sun-points when in the sun, maintains inertial hold when in eclipse, and commands an aerodynamic attitude (PnP/Sat has one very large thin-film array that generates a significant moment when exposed to the velocity vector) when the satellite’s orbit drops below a threshold value.

A variety of additional Activity Agents were also developed in the course of the PnP/Sat program. All were tested using the FSWIL environment mentioned in earlier sections but were withheld from the mission to avoid complexity. They include *ShedLoads*, *TakeImage*, *TemperatureMitigation*, *Commissioning*, and *MonitorShips* (described in an earlier example). *ShedLoads* will conserve energy by progressively powering off components when state of charge approaches a critical threshold. *TakeImage* is a tactical activity that allows targets in multiple theaters to be indicated and satisfied relative to their expressed priority. *TemperatureMitigation* manipulates the attitude of the spacecraft if out-of-bounds hot or cold conditions are detected within the satellite system. *Commissioning* is essentially an on-orbit checkout executive that checks phasing of actuators, calibrates alignment of devices with external fields of view, and determines the platform mass moments of inertia.

CONCLUSIONS

PnP/Sat successfully demonstrated the primary objectives of rapid spacecraft assembly. At the time of writing of this paper, the ORS office was engaged in exercises to quantify the degree of assembly risk associated with the approaches.

The fundamental tenants of PnP software were realized in the PnP/Sat application. Although running on three very lightweight processors, the flight software baselined for day-in-the-life system tests and environmental characterization demonstrated the ability to self-organize the software, recover capability upon

intentional device service interruptions, and re-distribute the software load when a processor was prevented from booting at initialization time.

Benefits to the Space Community

The Space Plug&play Avionics standards and the PnP approaches implemented on AFRL's PnP Sat program have great promise for tactical small satellites. While the rapid assembly benefits are undeniable, many await formal studies that will quantify the extent to which PnP can reduce cost in a mass-production model. The ORS Chileworks efforts planned for the next few years should go a long way toward developing PnP supportive infrastructure and developing a rich collection of certified satellite configurations based on the PnP Sat legacy. NASA has recently begun an initiative to develop PnP technology for a similar size class of science missions and intends to leverage much of the technology and infrastructure from the AFRL effort to accomplish that end.

The TacSat5 program aims to engage industry by soliciting a spacecraft bus requiring the liberal use of SPA. The hope is that industry will begin to embrace the standards, making them prevalent in military, science, and commercial markets in the relatively near timeframe.

Logical Continuation of PnP Research

PnP Sat2, the heir-apparent to PnP Sat, is already in motion. PnP Sat2 will springboard off of the lessons learned in the PnP Sat exercise to develop a tactically relevant platform capable of hosting tactical payloads and providing the performance necessary to achieve associated mission objectives. Toward this end, a larger modular bus structure (derived from the PnP Sat panel design) is being developed. Design efforts are similarly underway to produce the next-generation SPA infrastructure hardware to reside within those panels, capable of higher link rate support and radiation hardened for more challenging orbits. More capable computing nodes are similarly under development to host significantly more complex and processing intensive applications than were possible using the PnP Sat infrastructure.

ACKNOWLEDGEMENTS

The authors would like to the Air Force Research Laboratory's Space Vehicles Directorate, Kirtland AFB, N.M., for its continued support of PnP research through a variety of historical and ongoing programs.