3rd Responsive Space Conference
RS3-2005-5001

HOME OF THE 1903
WRIGHT FLYER PROJECT

Space Plug-and-Play Avionics

Jim Lyke, Air Force Research Laboratory Space Vehicles Directorate, Albuquerque, NM

Don Fronterhouse, Scientific Simulation, Inc., League City, TX

Scott Cannon, Utah State University, Logan, UT

Denise Lanza, Science Applications International Corp., Albuquerque, NM

Wheaton (Tony) Byers, Science Applications International Corp., Albuquerque, NM

3rd Responsive Space Conference
April 25–28, 2005
Los Angeles, CA

AIAA 3rd Responsive Space Conference 2005-**5001**

SPACE PLUG-AND-PLAY AVIONICS

Jim Lyke, Air Force Research Laboratory Space Vehicles Directorate, Albuquerque NM
Don Fronterhouse, Scientific Simulation, Inc., League City TX
Scott Cannon, Utah State University, Logan UT
Denise Lanza, Science Applications International Corp., Albuquerque NM
Wheaton (Tony) Byers, Science Applications International Corp., Albuquerque NM

ABSTRACT

The Air Force Research Laboratory is developing a system for rapidly building spacecraft based on adapting "plug-and-play" (PnP) approaches for use in space. This space plug-and-play avionics (SPA) system is based on an interface-driven set of standards intended to promote the rapid development of spacecraft busses (platforms) and payloads. As such, SPA is an open systems framework, combining commercial standards (such as USB) with carefully chosen hardware and software extensions necessary for modern real-time embedded systems (e.g. fault tolerance, higher power delivery, self-description). This paper will review the status of SPA and the efforts being made to standardize SPA through the AIAA.

INTRODUCTION

Operationally Responsive Space (ORS) is concerned with, once a mission need has been asserted, the dramatic reduction in timescale for fielding an operationally useful capability. ORS has become an important goal in DoD space systems for obvious reasons, since the pedantic pace of typical space systems development makes it difficult to exploit new technologies while they are "still new". To achieve ORS, a number of technology and culture barriers must be addressed to effect the desired systematic reductions in time for designing, building, integrating, launching, and bringing online a given space system. Much attention has been given to the creation of low-cost launch capability, as exemplified by the DARPA Falcon program[1]. However, even the availability of instant launch-on-demand will require the ability to create the spacecraft and its payload rapidly in order to make ORS decisive as a methodology for bringing capabilities to the warfighter more swiftly[2].

Creating a spacecraft rapidly is made difficult by a significant number of technical challenges, even when extensive standardization is enforced. Of the major subsystems of a typical spacecraft (avionics, software, attitude determination and control, thermal management, structure, and power generation), the development of avionics and software represent many of these challenges. Avionics and software also have significant hope for improvement, due to breakthroughs in "plug-and-play" (PnP) technology in terrestrial systems. PnP, as implemented in personal computing platforms, has resulted in an impressive proliferation of low-cost, rapidly integrable devices, and serves as the inspiration for more ambitious applications of its underlying principles.

The need for ORS, coupled with the possibilities of using PnP to accelerate the development and integration of electronics, has provided impetus for the pursuit of a concept for space plug-and-play avionics (SPA). This paper describes an initiative, spearheaded by Air Force Research Laboratory (AFRL), to create both the technology infrastructure for a meaningful PnP, as well as national standards for SPA. The mission statement for SPA is to enable the "six-day spacecraft", the idea that it may be possible within the next decade to reduce the time from "mission call-up" to operation on-orbit to less than one week. The paper is organized as follows. Two brief sections discuss relevant background and outline the requirements for a SPA framework. In the following section, the evolution and present definitions relating to the core elements of SPA are described. Finally, the current status of the SPA development as of this writing is addressed.

BACKGROUND

One of the simplest definitions of reconfigurability is the ability to demonstrate different non-trivial behaviors or physical states through software-only commands. This interpretation is appealing to space systems, as this gives new meaning to the principle of "action at a distance", particularly in the case of altering a system's characteristics (even its mission) by remote control. More practically, from the standpoint of responsive space, the concept of reconfigurability plays an important role in accelerating the time necessary to assemble and integrate components. Interfaces can be morphed, wiring harnesses configured, and, eventually, code auto-generated in direct response to the act of bringing together the components of a payload, spacecraft, and/or launch vehicle.

A brief historical perspective of both responsive space and terrestrial plug-and-play is provided as background leading to the initiation of the SPA concept.

Operationally Responsive Space. As part of recent US Air Force Space Command (AFSPC) study ("Analysis of Alternative"), a preliminary military utility analysis (MUA) concluded that significant benefits existed at the campaign level when ORS capabilities were applied. AFRL conducted its own responsive space study in 2004, which further amplified the need to emphasize "responsive satellites" as well as responsive launch. This Responsive Space Advanced Technology Study (RSATS) also led to the identification of PnP technologies, especially in avionics, as providing enabling benefit to the cause of ORS. A vision for "responsivonics" was described, based on previous work on a proposed Adaptive Avionics Experiment (AAE)[3]. The AAE concept identified four distinct elements leading to a modular, PnP avionics capability: appliqué sensor network, adaptive wiring manifold, high-performance computing on-orbit, and software definable radio. In the AAE concept, arbitrary compositions of these elements could be rapidly formed as needs dictated for a given mission scenario. Once on orbit, the elements could be reprogrammed or adaptively changed in response to faults and evolving mission needs. Of these four elements, the appliqué sensor network provided a particularly impactful contribution to the vision outlined in RSATS, one in which the spacecraft became essentially a self-organizing network of objects, easily assembled and integrated as if toy building blocks.

This appliqué sensor network concept was later refined, to become a new discipline involving machine-negotiated interfaces. The use of automation was believed to be important in speeding integration, as it could lead to reducing or eliminating error-prone human interpretations of interface control documents (ICDs). Automating the process of electronic self-configuration / self-organization could allow for rapid space vehicle construction.

Terrestrial / Commercial PnP. The idea of applying machine intelligence, the combination of hardware and software, to simplify the replacement / addition of components to personal computers and networks, has evolved for well over the decade. The work followed two basic tracts, that pertaining to personal computers (PCs) / networks and that pertaining to industrial, large-scale sensor networks. Most famously, the several generations of the unified serial bus (USB) standard[4] has resulted in a dramatic proliferation of commodity peripherals and storage devices. USB devices, when they work correctly, represent an almost ideal embodiment of the PnP concept. It is not necessary, for example, for users to consult manuals for configuring software interrupts or hardware jumper settings or to engage in painstaking deconflicting exercises. Often, the simple action of plugging a device into an available port will accomplish identification, resource configuration, registration, power distribution, and rapid availability of the device for immediate use by compatible applications. More recently, univeral plug-and-play (UPnP) has been introduced[5] as a technology-independent PnP system involving a combination of web-like strategies (to include http post and get methods) and SOAP[6] calls to facilitate messaging between the objects in a PnP network. Jini[7], a Sun-invented PnP system, exploits Java bytecode containers in PnP objects to facilitate registration and management.

PnP sensor networks evolved in parallel with PC-based PnP under different pressures, such as maintaining industrial sensor monitoring networks. These networks are often comprised of large numbers of simple, scalar sensors (e.g., thermometers) used in factory (for example) environments where rapid replacement and vendor independence are fundamentally important. NIST led the development of the 1451 series of IEEE standards to form smart sensor concepts, including network-capable application processors and transducer electronic data sheets (TEDS)[8]. Another competing concept for industrial PnP has been introduced by Echelon for embedding intelligence in devices to facilitate PnP. Though Echelon has created a family of proprietary building blocks to facilitate development of PnP networks (similar to the appliqué sensor interface modules described later in this paper), the underlying "LonTalk" protocol has been standardized (as EIA/CEA-709.1-A-1999)[9].

Why PnP? A common mistake in attempts to design modular frameworks is the misguided emphasis on standards and legacy components, as suggested in the depiction of a contemporary spacecraft avionics system shown in Figure 2. Emphasis is usually placed on forming a command and data handling (C&DH) system that exploits a standard backplane, such as VME or PCI (3U or 6U form factor). This standard permits the interchange of cards, in addition to a central processor and memory storage card set, used to connect the C&DH to components. Usually, even when standard interfaces are used for signaling (such as RS-422), it is often necessary to customize both hardware and software for each spacecraft component. Custom wiring harnesses, key to the specific combinations of

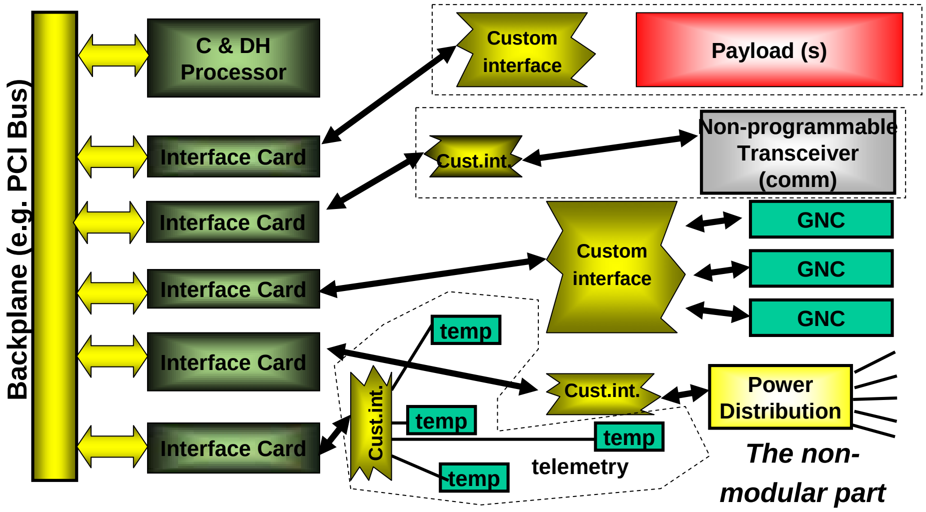

Figure 2. Contemporary avionics design.

these "standard" elements are required. In this case, even pervasive attempts to standardize do not solve the intrinsic problem, since custom software can require many months to write, harnesses can take months to construct, and customized electrical interfaces create additional obvious burdens in development.

The "responsivonics" vision for modular electronics, as originally outlined in the AAE concept development[3], also relies on standardization, but augments existing standards with other concepts to facilitate rapid composition of networks based on modular components. The responsivonics network employs a number of innovations, including: reconfigurable processors referred to as "malleable signal processors" (MSPs), grid-like processor networks with individual nodes referred to as fusion processors (FP), PnP networks, switched fabric (e.g. Spacewire), and an adaptive wiring manifold. Components would be to some degree designed to themselves be reconfigurable. For example, a fixed waveform communications transceiver could be replaced with a software-definable radio (SDR).

Under this depiction, several obvious advantages are conveyed in modular arrangements of the components shown in Figure 1. For example, pre-built SDRs with sufficient flexibility can be configured upon integration to support the waveform sets needed for a given mission. Reconfigurable processors can be used singly or in combinations to provide an amount of computational resource pool on orbit dictated by mission need. Switch fabric connectivity provides a convenient messaging infrastructure for aggregating

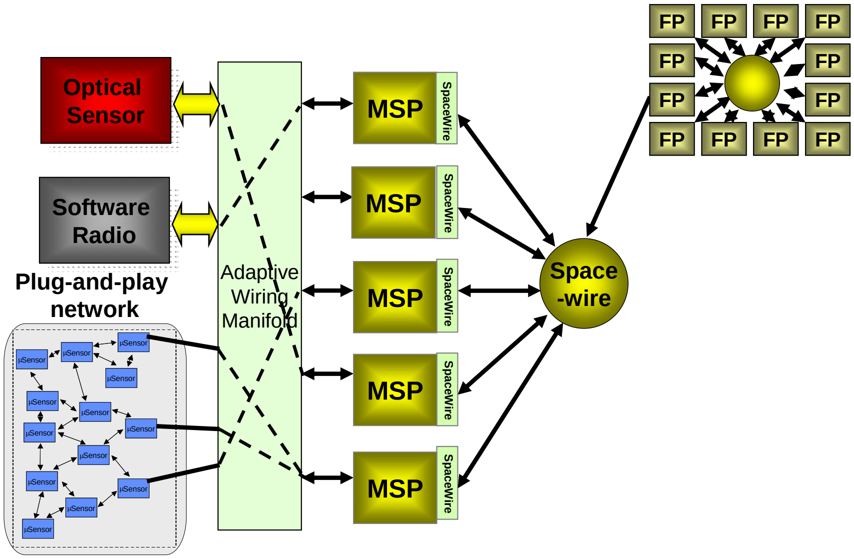

Figure 1. The "responsivonics" view of modular avionics as the basis of SPA

MSPs in combination with FP nodes, the latter resource providing a more power-efficient method for performing floating-point calculations. Possibly the most novel concept in this diagram is the notion of a programmable wiring system, enabling the possibility of *pre-built* wiring harnesses, configured at the time when components are integrated.

Even this simplistic depiction does not convey several key concepts fundamental to SPA. The PnP network, for example contain mechanisms similar to those in terrestrial PnP networks that make automatic the "join-and-discovery" processes that allow rapid integration to take place. Somehow, all of the components in Figure 1 must link into this infrastructure so that automatic configuration can take place. Self-description is an integral part of the Figure 1 concept, and some means of managing the descriptions of hundreds of components must be devised. These "invisible" parts of the Figure 1 concept, combined with a number of the explicit elements, became the basis of what is now referred to as the SPA concept.

THE SPA INITIATIVE

AFRL, in collaboration with other government, industry, and academic organization, conducted a series of workshops focused on the definition of standards for "SPA".

As a result of the early SPA discussions, a multi-generational scheme was proposed, as suggested in Figure 3. The need for the dispersion of SPA development was made clear by the realization that some of the Figure 1 would be more easily mobilized and qualified for spaceflight within a short time, whereas more involved concepts (such as the adaptive wiring manifold) would require significantly more time to develop and integrate into a SPA framework. Ideally, this multi-generational approach would be executed in concert with the emerging Joint Warfighting Space Demonstration (JWSD) series of spacecraft, so that SPA components could be readily integrated into particular JWSD missions. Mature SPA components could be integrated as baseline or primary elements of a particular JWSD mission, while less mature SPA components could be included as experiments themselves in JWSD missions, to be "promoted" for primary mission role in later JWSD mission.

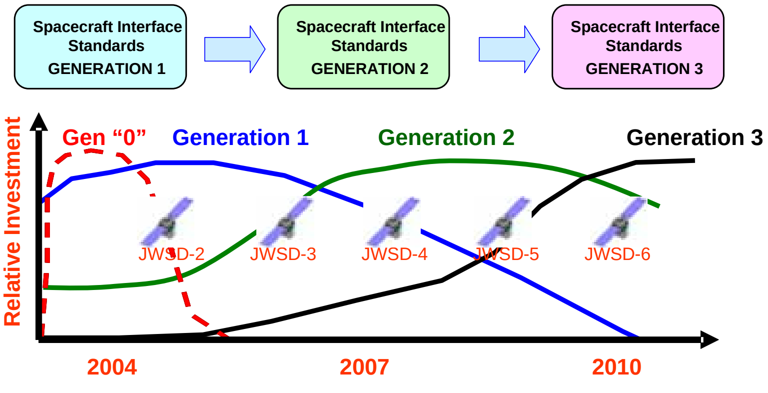

Figure 3. A multiple "spiraled" approach to SPA.

The specific technologies and standards to be developed for each "spiral" of the multi-generational SPA approach are briefly summarized as follows:

Generation "zero" (Gen 0). Gen 0 represents the accelerated implementation of the Gen 1 technology concepts. Since the aspiration of the SPA concept is to create standards and promote implementations of space-qualified PnP hardware, the use of radiation-hardened technologies are indicated, which are not compatible with accelerated implementations. "Gen 0" was defined to provide the opportunity to develop near-term reference implementations, in part short-circuiting this delay. As such, Gen 0 implementations may not be suitable for the harshest radiation environments, but would provide useful implementations for early brassboard development. The timeframe focus of Gen0 is from 2004 – 2006.

Gen 1. The goals of the Gen 1 development include: (1) develop PnP interconnect, hardware, and software to support centralized PnP in spacecraft; (2) establish rad-hard components for these interconnect systems; and (3) establish an open systems approach to unite the standards and concepts of appliqué sensor network. Gen 1 focuses on the development of USB-based, USB+Spacewire-based, and Ethernet-based PnP technologies, combined with the supporting software concepts. USB provides command, control, and configuration support for all SPA components (except those based on Ethernet). For low-speed devices, such as thermometers and gyros, the USB-based version of SPA ("SPA-U") also supports data transport, whereas the USB+Spacewire-based version of SPA ("SPA-S") dedicates a Spacewire-link (625Mbps including overhead) to data transport. The timeframe focus of Gen1 is 2004 – 2008.

Gen 2. The goals of Gen 2 include: (1) movement from centralized to distributed architecture; (2) maintain backwards compatibility with Gen 1 developments; (3) develop a *reactive interface* approach to improve flexibility of certain complex components; (3) improve "ontological" support through more generic application programming interfaces and driverless registration of nodes into the network at large; and (4) seek the development of suitable > 1Gbps interconnect transport standards (e.g. 10 Gigabit Ethernet, RapidIO). The timeframe focus of Gen1 is 2004 – 2010.

Gen 3. The goals of Gen 3 include: (1) Implementation of "geographic awareness" of components; (2) Adaptive wiring manifold; (3) Expand support of interconnect transport to include wireless; (4) Support of cognitive software for improved "self-awareness". The timeframe focus of Gen1 is 2004 – 2012.

SPA CONCEPT: ARCHITECTURAL OVERVIEW

The Space Plug-and-play Avionics (SPA) approach fully supports an à la carte method of constructing arbitrarily complex arrangements of virtually any sensor or actuator type. This behavior makes the network not only easy to expand and modify, but also makes it robust to component failure from either natural causes or from deliberate attack.

Space plug-and-play avionics (SPA) is defined as an interface-driven standard (or set of standards) intended to promote the rapid development of spacecraft busses (platforms) and payloads. The SPA standard comprises an open systems framework, which combines core commercial standards (such as USB) with carefully chosen hardware and software extensions necessary for modern real-time embedded systems. In the SPA concept (Figure 4), a spacecraft is a network of SPA components, in this case connected to a central command and data handling (C&DH) unit. In conventional (non-SPA) spacecraft, many of the spacecraft components (the "a" through "h" blocks, for example) would be directly connected to the C&DH system. In SPA, networks are more "Internet-like," meaning the topology is somewhat free-form (amorphous) in nature.

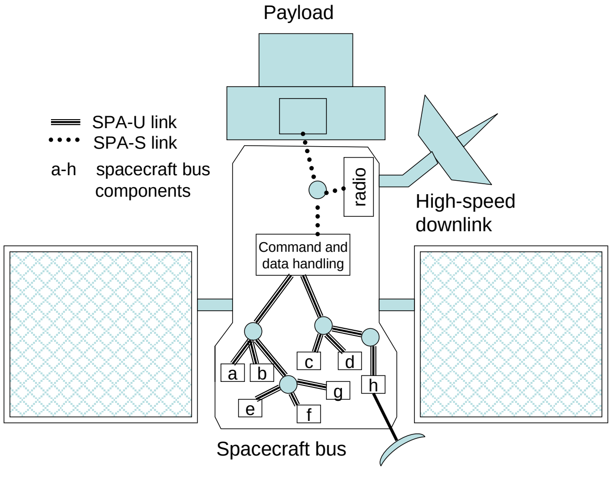

Figure 4. Spacecraft as a collection of Objects.

While the paradigm sought for SPA is similar to the ease-of-use model promoted by "plug-and-play" (PnP) in the PC industry (e.g., USB-based "thumb drives"), SPA is not simply the transplant/grafting of a "consumerized" PnP onto aerospace electrical components. Instead, while exploiting convenience standards for physical and transport layers (e.g. such as supplied by USB, Spacewire, and Ethernet), SPA represents a re-engineering of PnP to accommodate special constraints not typically faced by most high-volume commodity PnP products:

Environment / Fault tolerance. Even in initiatives for rapid-turn / low-cost spacecraft, reliability is an important consideration. The space radiation environment poses special challenges for electronics, which must be engineered to endure the accumulation of total ionizing dose and single event effects such as latch-up and disruption of charge storage configurations in integrated circuits (i.e., bit-flips in memories and flip-flops). Beyond this, it is desirable to provide explicit support for redundancy at a system level.

Synchronization. A unified notion of time must be coordinated across the components of a space system, even if they are loosely-connected.

Higher power delivery. Initiatives such as power on ethernet (PoE) recognize the value of binding power distribution to data transport. USB, which has very sophisticated power management facilities, does not deliver power levels consistent with spacecraft components, and PoE does not provide 28V delivery, which is commonly used in many spacecraft designs.

Driverless PnP. The PC version of PnP requires the use of drivers, which are developed for a limited number of platforms. In spacecraft, PnP driver design is problematic, owing to the need for each PnP component to support a wide variety of C&DH configurations. In driverless PnP, the information necessary to access PnP device services is bound in a universal way.

These considerations provided the impetus for the present direction of the SPA development, and the remainder of this section is dedicated to describing the fundamental Gen 1 SPA interface, namely the USB-based SPA (SPA-U) standard, which is the most mature of the three presently defined Gen 1 approaches (i.e., SPA-U, SPA-S, and SPA-E).

SPA-U (USB-Based SPA)

SPA-U is based on the existing USB (version 1.1) interface standard, which supports 12 Mbps data transport, suitable for interfacing with most spacecraft devices. In order not to alter the logical definition of the electrical part of the USB standard, it is necessary to supplement the four conductors that provide the USB connection with others to supply power (two pins), synchronization (two pins), and grounding support (one pin). This design approach allows the use of existing USB intellectual property for implementing radiation-hardened implementations, while providing the additional support needed for power (up to 3A at 28V) and synchronization (using 1 Hz sync pulse) of spacecraft components. Because this is not identical to USB, the USB-derived interface is coined “SPA-U”.

Like USB networks, SPA-U networks consist of three different types of components: (1) hosts, (2) endpoints; and (3) hubs.

SPA-U hosts are analogous to USB hosts, except for the distribution of 28V power, synchronization, and the availability of accessory grounding. As in the case of USB hosts, SPA-U hosts are the root of a tree-structured, dynamic network, and all communications are between host and endpoint.

SPA-U endpoints are “leaves” of the SPA-U tree network, and every SPA-U device contains at least one SPA-U endpoint. Ideally, simple SPA-U devices are connected to the rest of the spacecraft electrically through only the SPA-U endpoint interface, and for devices requiring less than 3A electrical power or having less than 10 Mbps, there is little reason to choose a high-power or higher-speed interface (e.g. SPA-S or SPA-E).

SPA-U hubs are similar to USB hubs, but there are two important differences. First, the SPA-U hubs are self-directing. Normally, USB networks are directional, with “upstream” referring to the direction towards the root (host) port and “downstream” referring to the direction away from the root port. In the robust hub, any endpoint or host can connect to any port, and directionality is discovered dynamically, permitting more complex, redundant topologies to be formed that would be normally illegal in a USB network. The second important different is that the SPA-U hub manages the switching / distribution of spacecraft power from a distributed power grid. The power switching is controlled by a “captive endpoint” within the SPA-U hub itself, which receives commands from the SPA-U host.

SPA-U networks, while having some advantages over USB networks, share in some of their disadvantages. For example, the chaining depth of hubs is limited to five levels, and only 127 endpoint devices are supported. To combat this problem in part, it is possible to employ multiple SPA-U host connections from a C&DH system, allowing the average chain depth to be reduced.

SPA-U Applique Sensor Interface Module (ASIM)

It became clear since the formation of the appliqué sensor network concept (the progenitor of the SPA concept) that a compact reference design of the standard would be useful if not essential for early adoption. Such a reference design would ideally provide a bridge between a compliant implementation of the SPA-*x* standard (e.g., SPA-U) and a user design and contain automatic support for useful services, including power management, synchronization, electronic datasheet (i.e., xTEDS). Especially if compact, low-power, and space-qualified, this reference design could be directly embedded in final user products, considerably reducing the burden of creating a SPA device.

The “preferred” embodiment of a SPA reference design is the *appliqué sensor interface module (ASIM).* One SPA-U ASIM concept is shown in Figure 5. Key features of the ASIM are briefly discussed.

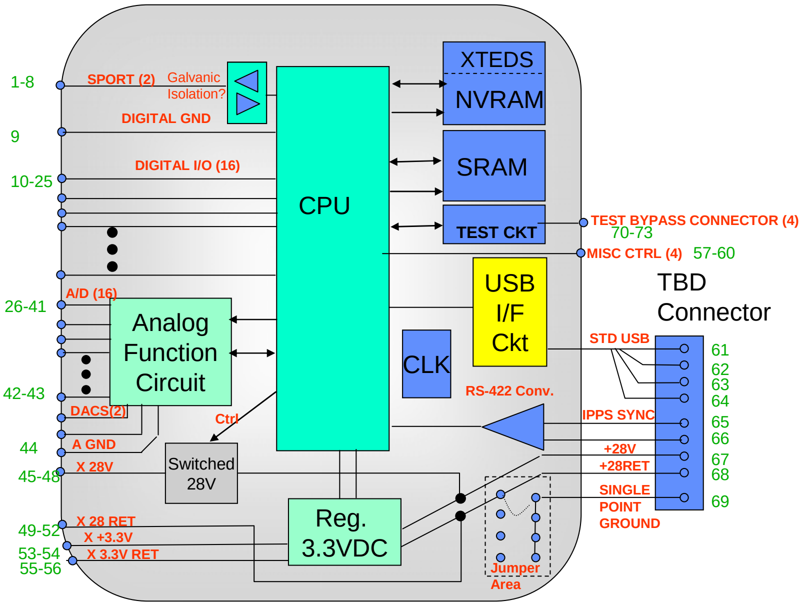

Figure 5. SPA-U applique sensor interface module (ASIM).

Central processing unit (CPU). Automated support of xTEDS and PnP requires machine intelligence, which could be handled minimally by an eight-bit microcontroller (e.g., 8051).

Non-volatile memory. A writeable, non-volatile store is required to maintain codespace, xTEDS, and writeable data structures.

USB interface. Supporting even the simplest USB endpoint requires significant functional overhead, and a dedicated USB interface circuit is necessary to support a SPA-U interface.

User facilities. A number of frequently-required user features include digital discretes, analog discretes, and serial ports. Providing native support of such features in the ASIM promotes close-coupling of user designs to the SPA-U interface, simplifies coding and reduces physical overhead.

Power management. The ASIM must receive power during the SPA-U device initialization process, however the primary user device cannot be powered until commanded. To meet these constraints, the ASIM can extract small amounts of power from the 28V bus. Device power is managed through switchable relay connections controlled by the ASIM CPU.

Clock management. The ASIM should manage the 1PPS synchronization pulse from the SPA-U interface and make timestamps available.

Test bypass interface. The ASIM should support connection through a small secondary connector to an emulation system, to provide an effective means of in-system test and verification.

SPA Device Software Support. The most important features affecting ASIM utility is the careful design of "client-side" API and auxiliary tools to reduce the burden of code and xTEDS development.

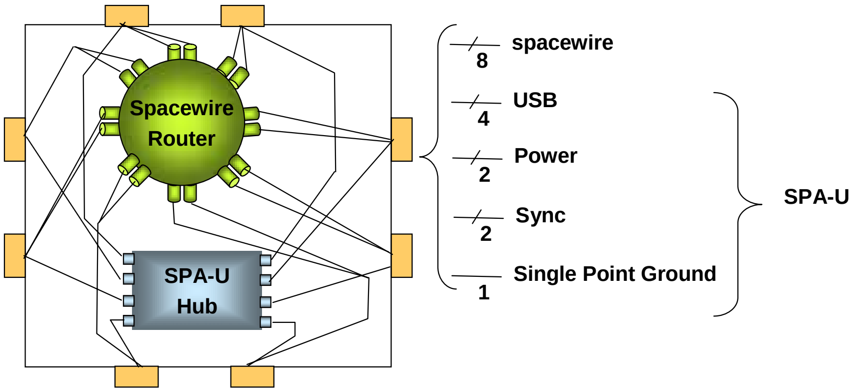

Figure 6. Simplified depiction of SPA-S router.

SPA-S (Spacewire based SPA)

Spacewire is a European Space Agency (ESA) standard[Error! Bookmark not defined.] that supports high data rate transport (up to 625 Mbps has been demonstrated) and routable interconnect using a *switched fabric* concept. SPA-S combines a Spacewire link with a SPA-U connection in a single connector in which the power conductors are "up-sized" from a 3A maximum to a 40A maximum. The decision to increase conductor size was based on the normal assumption that very high-speed links are usually associated with payload elements that require more physical power. SPA-U otherwise represents a convenient set of features (USB link for command and configuration support, 1PPS synchronization, and a single point grounding connection) to co-bundle with Spacewire. Hence, a SPA-S device is really a SPA-U device with higher speed data transport and power handling.

The SPA-S component family is more complex, but smaller than the SPA-U family. Unlike USB, Spacewire is a peer-to-peer networking approach, and as such, there is no SPA-S host. SPA-S endpoints (embedded in each SPA-S device) minimally contain a Spacewire link physical layer and protocol logic, along with a SPA-U ASIM that is modified for higher power handling. SPA-S routers, unlike the SPA-U hubs, are non-blocking crossbars capable of sustaining multiple simultaneous pairwise connections. The simplified depiction of a SPA-S router is shown in Figure 6.

Plug-And-Play Software: The Satellite Data Model

One abstraction of software engineering for PnP follows a vertically-layered model (Figure 7), reminiscent of the well-known seven-layer open system interconnect (OSI). At the bottom of this stack are the PnP components themselves, which interface physically using SPA-*x* but logically comprise a *component layer*. The component layer connects into a *"middleware" layer* referred to as the *satellite data model* (SDM). Above this middleware is the *application layer*. Applications access the PnP object-services through API calls to the SDM, which enforces an insular discipline in systems development. It is not, for example, necessary to write code to control specific thermometers, which might require modification when different thermometers are chosen. Rather, this layered approach encourages *device independence* in application design, which is one of the principles that permit more rapid integration of components. It is possible to define a final *mission layer*, potentially as a script-driven interface to the application set.

The key innovation in the PnP software architecture is the SDM. The goal of rapid satellite design, integration, and test requires that established, but time-consuming, concepts be rethought and revamped. For example, the interface control document (ICD) focuses on the device, but the SDM shifts the focus to the *data* provided or used by *processes*. In this context, a process can refer to an application, a PnP device (sensor, actuator), or any other user or producer of data on the satellite. The SDM does not focus on the electrical transport mechanism, so in principle any number of SPA-*x* interfaces could be devised. Rather, the SDM is based on the transport of data. In all other respects, SPA proposes specific, standard electrical interfaces (e.g. SPA-U, SPA-S, SPA-E) that reduce that physical part of the ICD *to* standards compliance. In the case of the SDM, the ICD is reduced to an electronic description of the data that can itself be embedded in the process and shared with other processes.

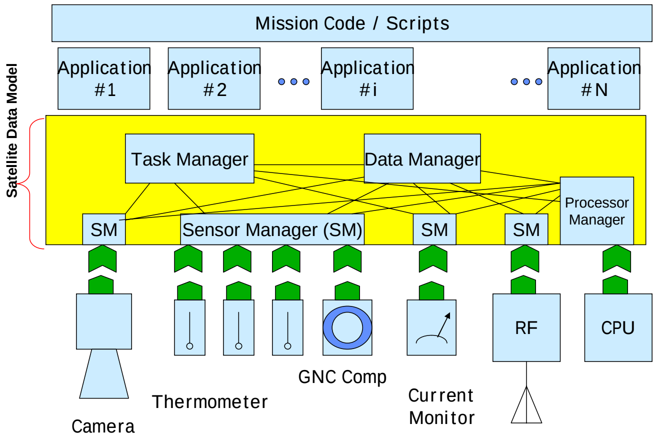

Figure 7. Vertically-layer software engineering model for PnP.

Ontology plays an important role in SDM. For the various aforementioned processes to understand each other, they must speak the same language. To do this, SDM requires a public Common Data Dictionary (CDD) whose contents are created by the community of process developers and managed as a public resource (for example, through a website). The CDD concept is key to a *data-oriented* model, and it enables disparate teams to develop processes in different places and times that are able to understand what data each produces or requires. It permits a community understanding for the development of the device xTEDS, as well as the applications that exploit them in the various SPA components. Done properly, these concepts support effective distributed development and reduce the many opportunities for misinterpretation in paper ICDs.

The SDM defines a series of interacting "function managers":

- *Processor Manager* – resident on each processor and is responsible for keeping that processor busy;
- *Data Manager* – keeps track of all data available at any given time and supports data queries;
- *Task Manager* – keeps track of active and pending tasks;
- *Sensor Manager* – provides the PnP interface to the processing network; and
- *Network Manager* – explores the network and maintains routing tables.

The managers are logically a single function even though they can have a multi-instantiated distributed implementation. These "managers" support data access, task management, and network discovery. *Data access* accumulates descriptions of what data is produced by system processes and how that data can be accessed. *Task management* keeps track of what processes are executing on what processors and their statuses along with what additional tasks are needed. *Network discovery* determines what components are connected to the network, their addresses, and associated routing tables.

The processor manager bears special mention. It is a special process resident on each processor (since SDM is intrinsically designed to be distributed onto networks) that handles task acquisition and execution along with providing basic support functions. These functions include messaging between processes, maintaining a real-time clock, and providing a periodic heartbeat to the system (i.e., the task manager). The special "per processor" process continuously monitors activity of the parent processor and periodically checks for the existence of pending tasks that can be executed by the parent. If any are found, the appropriate executables are loaded and run. While no operating system is required *per se*, the process can be multi-threaded, handle interrupts, and utilize an operating system as appropriate based upon the specific processor.

STATUS OF SPA DEVELOPMENT EFFORT

Since the first workshop (July 2004), the SPA development activity has been aggressively pursued by AFRL and other participating organizations. This section briefly highlights recent accomplishments as of the time of this writing.

SPA Technical Committee

As a by-product of the first two SPA workshops (July and September 2004), a SPA Technical Committee (TC) was formed (November 2004), and a Committee on Standards (CoS) was approved by AIAA as part of this TC. The TC has formed a number of working groups, whose focus is the development of SPA technologies into a documented form. The CoS works to convert these documents into AIAA-approved standards and guidelines for the US space industry. There are currently four working groups that have been defined by the TC to develop SPA technologies and documents:

Gen 0. The Gen 0 working group pursues the development of near-term hardware interface concepts based on commercial off-the-shelf (COTS) technologies

Gen 1. The focus of the Gen 1 working group is similar to Gen 0, but focuses on the creation of radiation-hardened reference implementation.

Software. The software working group emphasizes the development of PnP middleware, the definition of electronic datasheets, and creation of ontologies (vocabularies).

Advanced Technology. This working group explores trade studies and technologies and concepts beyond the Gen 1 horizon.

The first key document of the SPA TC, the *SPA Guidebook*, is nearing completion as a draft at the time of this writing. The intent of the guidebook is to provide a comprehensive overview of the SPA-U and SPA-S technologies, which would spawn a series of standards and guidelines under the auspices of the AIAA as the primary standard development organization.

SPA-U Component Development

The advent of design-hardening at aggressive feature sizes (e.g., 0.13 µm) and structured ASIC approaches, even as research projects, have allowed impressive progress to be realized in compressed timescales. For example, several digital-only SPA-U building blocks were released to fabrication in approximately eight months after identification of USB as an interconnect standard for SPA. Two of the more significant test chips are shown in Figure 8, shown in simplified block diagram and post layout forms. The first design, a USB endpoint (Figure 8a-b), was configured to interface directly to a 8031/51-class microcontroller. The second design, a five-port USB hub (Figure 8c-d), is a compliant USB 1.1 implementation. In both designs, no physical layer is included, but COTS components are available as an interim solution. These designs were made possible through the recent DARPA/MTO design-hardened initiative[11].

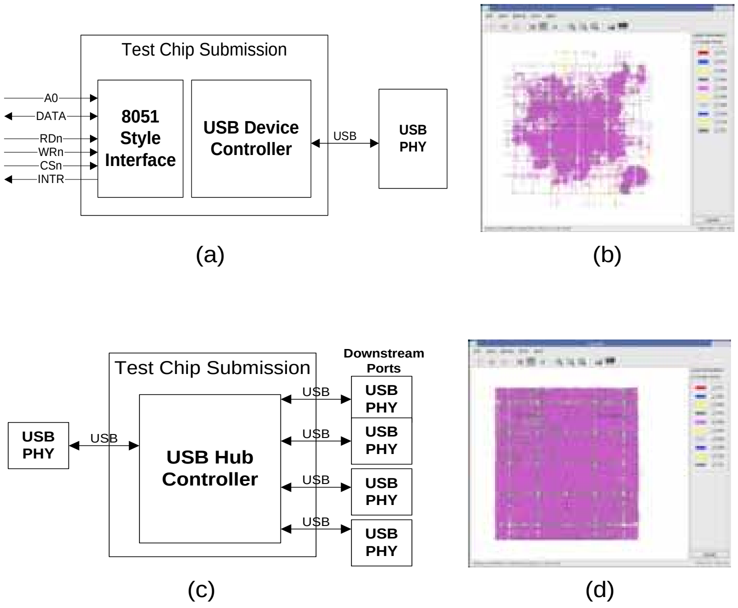

Figure 8. Recently released SPA-U components. (a) USB 1.1 endpoint. (b) Design-hardened structured ASIC layout. (c) USB 1.1 five-port hub. (d) Design-hardened structured ASIC layout.

A simplified, reduced form of Gen 0 ASIM (Figure 9) was also created and has been used in initial network demonstrations. The ASIM is designed around a Cygnal 8051F320 CPU with limited memory resources, but is nevertheless capable of managing simple scalar sensors with rudimentary XTEDS support. As a Gen 0 concept, it is not radiation-hardened, though radiation tests are planned to be completed by July 2005. The ASIM is not complete, as it does not currently support power management, synchronization, or test bypass. An updated implementation that will be functional complete is planned to be completed by October 2004.

A simplified, reduced form of Gen 0 ASIM (Figure 9) was also created and has been used in initial network demonstrations. The ASIM is designed around a Cygnal 8051F320 CPU with limited memory resources, but is nevertheless capable of managing simple scalar sensors with rudimentary xTEDS support. As a Gen 0 concept, it is not radiation-hardened, though radiation tests are planned to be completed by July 2005. The ASIM is not complete, as it does not currently support power management, synchronization, or test bypass. An updated implementation that will be functionally complete is planned to be completed by May 2005.

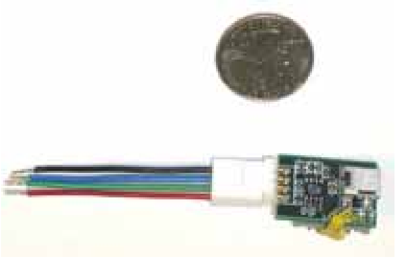

Figure 9. Compact reduced implementation of Gen 0 ASIM used in initial feasibility implementation.

A reduced three-port implementation of a SPA-U hub has been demonstrated building upon the same processor used in the Figure 9 ASIM and a commercial USB hub. The processor is carried within the hub as a captive endpoint, where it is able to accepts commands from a host to control power or to disconnect ports in the case of a fault. It is necessary to power the processor initially, since during initialization the commercial hub is not connected to any port, but implements a greedy search procedure to identify a valid upstream connection. Following the discovery of an upstream connection (i.e. a path to a USB host), the processor connects commercial hub ports to SPA-U external ports using CMOS analog switches. As in the previously described ASIM, this SPA-U hub is incomplete, lacking a complete power management facility, which will be implemented by the end of 2005 using a relay matrix under control of the captive microcontroller.

SPA-S components. Spacewire intellectual property (IP) has been developed through a previously existing SBIR project[12], and brassboards containing Spacewire routers (eight-port) and PCI-to-spacewire interfaces (implemented on Xilinx Virtex 2 field programmable gate arrays) are currently operating. Progress is being made toward a design-hardened standard cell ASIC (0.18 μm), largely complete except for the implementation of the analog physical layer. Since SPA-S links and hubs are built upon SPA-U components, little effort has been invested in creating SPA-S components, though it is expected that initial prototypes may be available by mid-2006.

In related work, a lightweight implementation of Spacewire (called "SpaceWireLite") has been developed that is interoperable with Spacewire links, but is not capable of the higher speed data transport. A testchip containing the digital core has been released to fabrication on the same run as the Figure 8 components. SpaceWireLite, as a peer-to-peer networking system, may have significant advantages over USB and as such may form an alternate Gen 2 SPA technology for simple devices. One challenge in introducing a USB replacement interconnect will be in building a sufficiently flexible hub/host system capable of backwards compatibility with the Gen 0/1 infrastructure described in this paper.

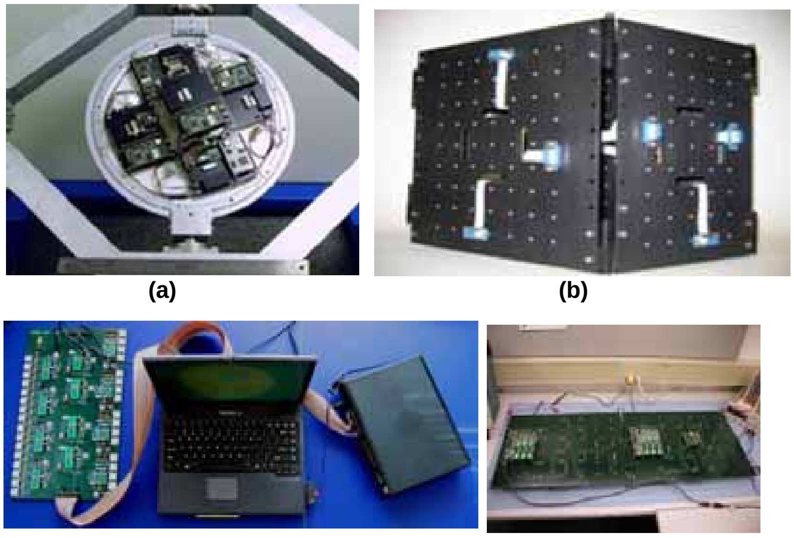
(a)
(b)
(c)
(d)

Figure 10. Demonstration / test SPA network configurations. (a) Gen 0 – class PnP demonstration. (b) SPA-U 0 brassboard panels. (c) SPA-U radiation test board. (d) Adaptive wiring manifold test panels.

SPA System-level demonstrations

Several recent test demonstration brassboards have been completed supporting feasibility assessments of the SPA-U architecture, shown in Figure 10. In AFRL-sponsored work, Microcosm recently demonstrated PnP concepts that pre-date the SPA effort but are very similar to the present Gen 0 approach (Figure 10a). In this work, a number of guidance, navigation, and control (GNC) components were hot-swapped on a small, three-axis moveable jig, demonstrating the automatic join, discovery, and recognition of PnP components by control loops implemented in application software that is aware of those components. One of the first demonstrations involving SPA concepts exploited the reduced implementations of SPA-U ASIMs and endpoints is shown in Figure 10b. In this demonstration, SPA-U hubs and ports were spatially arranged on mock spacecraft panels and a laptop computer was used to mimic a spacecraft C&DH system. This demonstration was successful in demonstrating the dynamic evolution of network topologies and the successful drop-off and addition of SPA-U components. Multiple laptops have also been used to demonstrate preliminary implementations of a distributed SDM. To support flight applications of Gen 0 components, a radiation test board containing a number of candidate COTS components (Figure 10c). Not all demonstrations are Gen 0 focussed. The first known adaptive wiring panel (a Gen 3 SPA technology) demonstration was recently completed. This demonstration, shown in Figure 10d, contains two panels, each supporting a number of switchboxes that contain a number of latching MEMS switches and a number of connection ports. The fully populated panels will contain over 200 MEMS switches, and operating configurations containing > 100 MEMS switches have been operated. In the demonstration, two connection ports were connected to Spacewire links, and in the beginning of the demonstration, no electrical pathways exist between the links. Following a computer-controlled configuration of the panels, wire pathways are formed between the eight individual conductors of each Spacewire link. The pathways are non-volatile, as represented by the elemental settings of each bistable switch. After configuration, streaming DVD video signals are transmitted to demonstrate a certain level of signal integrity can be supported using a programmable wiring system.

SUMMARY / CONCLUSIONS

This paper has introduced a new approach to achieving plug-and-play (PnP) in aerospace systems. Its primary motivation is provided by operationally responsive space (ORS), specifically the hope of making possible the construction of space systems in a dramatically reduced timeframe, from months to days. But PnP is a term that over time has inherited a number of meanings, some of which might be wishful thinking. The SPA effort sought to achieve a PnP technology capable of rapidly forming a system, even dynamically, exploiting machine-negotiated interfaces to, in effect, self-organize that system. Recognizing this as a goal, it is clear that the random citation of standards would not be enough. Rather, it was necessary for SPA to follow a different tact, one that drew from the considerable base of terrestrial standards in a way to enforce the vision of PnP needed to make ORS a reality. Though interconnect standards, including USB, Spacewire, and Ethernet have been chosen, they are themselves not sufficient to achieve PnP. It has been found necessary to supplement the commercial interconnections with other provisions for power and synchronization. More importantly, a software infrastructure was developed to make possible a deeper idea for PnP, one not just capable of supporting automatic component identification, but one capable of device independent interchange, robustness, and flexibility to meet the diverse needs of ORS mission concepts.

REFERENCES

[1] Walker, S. and J. Anttonen. “Falcon”, http://www.darpa.mil/tto/programs/falcon.html, updated 7 January 2005.

[2] Brown, Kendall K. "A Concept of Operations and Technology Implications for Operationally Responsive Space", Air and Space Power Chronicles, 01 June 04 (http://www.airpower.maxwell.af.mil/ airchronicles/cc/brown2.html).

[3] Lanza, D, Lyke, J., Zetocha, P, Fronterhouse, D, Melanson, D, "Responsive Space Through Adaptive Avionics", presented at AIAA 2nd Responsive Space Conference, 2004.

[4] Key documents for the USB standard are available at http://www.usb.org.

[5] Lyke, J. et al. "Plug-and-Play – An Enabling Capability for Responsive Space Missions" presented at the AIAA 2nd Responsive Space Conference, AIAA 04.

[6] Kulchenko, P, Snell, J, and Doug Tidwell. Programming Web Services with SOAP, O'Reilly Books, New York, ISBN 0-596-00095-2, December 2001.

[7] Key documents for Jini are available at http://www.jini.org.

[8] More information on the IEEE 1451standard series is available at http://ieee1451.nist.gov/.

[9] More information on the Lonworks protocol is available at http://www.echelon.com.

[10] Key information on Spacewire is available at http://www.estec.esa.nl/tech/spacewire/.

---

[11] Mayer D.C. and Ronald C. Lacoe, "Designing Integrated Circuits to Withstand Space Radiation", Crosslink, publication of the Aerospace Corporation, Summer 2003.

[12] "Multiprocessor and ASIC standard interconnect for space system", USAF contract F29601-01-C-0030, 14 September 2001.