Reuse and Interoperability of Avionics for Space Systems

Robert F. Hodson[1]

NASA Langley Research Center, Hampton, Virginia 23681

The space environment presents unique challenges for avionics. Launch survivability, thermal management, radiation protection, and other factors are important for successful space designs. Many existing avionics designs use custom hardware and software to meet the requirements of space systems. Although some space vendors have moved more towards a standard product line approach to avionics, the space industry still lacks similar standards and common practices for avionics development. This lack of commonality manifests itself in limited reuse and a lack of interoperability. To address NASA's need for interoperable avionics that facilitate reuse, several hardware and software approaches are discussed. Experiences with existing space boards and the application of terrestrial standards is outlined. Enhancements and extensions to these standards are considered. A modular stack-based approach to space avionics is presented. Software and reconfigurable logic cores are considered for extending interoperability and reuse. Finally, some of the issues associated with the design of reusable interoperable avionics are discussed.

Nomenclature

AAP	=	Advanced Avionics Project
AMBA	=	Advanced Microcontroller Bus Architecture
API	=	Application Program Interface
cPCI	=	Compact Peripheral Component Interconnect
EDAC	=	Error Detection and Corrections
FPAA	=	Field Programmable Analog Array
FPGA	=	Field Programmable Gate Array
FPOA	=	Field Programmable Object Arrays
FPTA	=	Field Programmable Transistor Array
IP	=	Internet Protocol
Krad	=	Kilo rad
LRU	=	Line Replaceable Unit
Mbps	=	Mega bits per second
NASA	=	National Aeronautics and Space Administration
OS	=	Operating System
PCI	=	Peripheral Component Interconnect
RTOS	=	Real-Time Operating System
RSC	=	Reconfigurable Scalable Computing
SRAM	=	Static Random Access Memory
TMR	=	Triple Modular Redundancy
USB	=	Universal Serial Bus
VME	=	Virtual Memory Extension
VITA	=	VMEBUS International Trade Association
XML	=	Extensible Markup Language

[1] Chief Engineer, Electronic Systems Branch, NASA Langley Research Center, M/S 488

I. Introduction

AVIONICS reuse and interoperability have always been a challenge for the NASA and the space industry. There are many reason for this including procurement approaches, “not invented here” syndrome, design propriety, lack of space standards, and the lack of standard design approaches. This paper will present a NASA avionics example that attempts to deal with some of the issues that limit interoperability and reuse. It also gives an overview of some NASA technology development efforts and approaches for avionics’ reuse. Lastly, it points to some of the road blocks to reusability and interoperability in space systems.

II. A Current NASA Example (Advanced Avionics Project)

The Advanced Avionics Project (AAP), underway at NASA Langley Research Center is one example of an effort to develop common avionics for space systems. This project developed hardware and software that could be used on a variety of space systems. The project looked for common functionality across space systems and focused on implementation of these hardware and software functions with extensibility to handle the custom requirements of future systems. The design approach leveraged commercial processors and operating systems and augmented these design elements with additional hardware and software. The chassis and boards developed for AAP are shown in Figure 1.

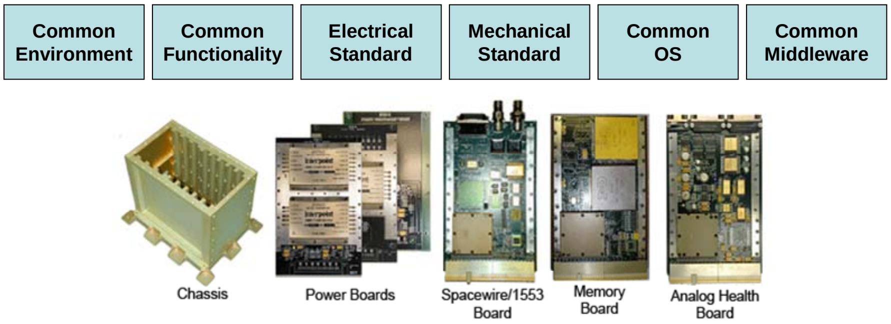

Figure 1. NASA’s Advanced Avionics Project (AAP) cards and chassis.

AAP integrates an extensible electro-mechanical design with modular functional units implemented in a 3U cPCI form factor. The system uses a commercial real-time operating system (VxWorks) and has implemented common process communication and command interpretation functionality in middleware. The intent is to build a flexible hardware and software platform with core functionality that can be easily extended and customized to meet specific mission requirements.

III. Enveloping Design Space

A key enabler of avionics reuse is enveloping the design space. In the case of AAP, multi-year missions in Earth orbits were the primary design driver. To accomplish this, the system is designed to withstand Delta and Atlas launch loads, reasonably high radiation (100krads(Si) total incident dose), is conduction cooled, and has many other features that enable reuse for this class of mission. Choosing the right class of missions to envelope can be challenging. Over designing can add cost to the system, and under designing limits reusability. Finding the “sweet spot” in this trade space requires looking at the availability and cost breakpoints in the marketplace. For example, building a 20krad system, as opposed to a 10krad system may only require a modest cost increase, but building a 50 to 100krad system may add significant costs due to the lack of qualified parts. Another example that can drive cost is the reliability class of the part (i.e. Grade 1, Grade 2, Class-S, etc). Thus the mission, and volume of missions, should be studied in conjunction with cost drivers.

IV. Standards, An Enabler for Design Reuse

To date, design standards have been generally inadequate for attaining reuse in space systems. Commercial standards like VME, PCI, and cPCI have helped but are commonly adapted or customized for space systems. In the AAP example, the cPCI with the VITA conduction cooling standards were used for the board designs. The designers did not take liberties with component height limitations, board-to-board spacing, or user defined pins, making the mechanical and electrical board interfaces well-defined and thus interoperable, reusable, and extensible. But certain aspects of AAP design were not addressed by these standards (i.e. the power interface). Input (box power) and board rail power levels, were selected and fixed at 28V and 3.3V, respectively. The power boards also had custom interfaces to the backplane, although a modular approach was used.

An example of a board level approach that addresses the short comings of other standards is a stackable circuit board approach that is backwards compatible with the PC/104-Plus standard. This approach, called SPACE-104, extends the electro-mechanical design of PC/104-Plus to include conduction cooling, venting (for ascent pressure), structural support (for launch loads), and power supply commonality[1]. The approach also provides for 64-bit PCI for high bandwidth applications, such as science instruments. The developers would like to evolve this approach to a next generation space standard, see Figure 2.

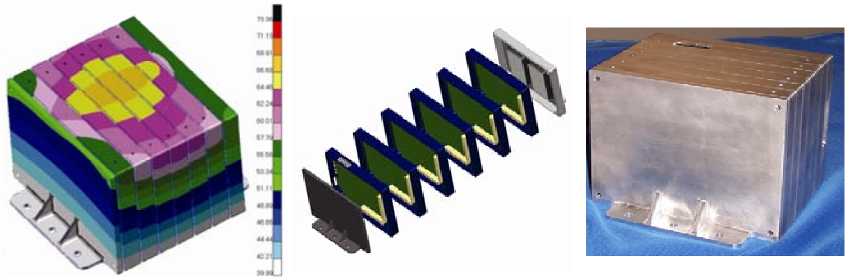

Figure 2. SPACE-104 stackable space system design approach.

In addition to board level interconnects, such as PCI, most space systems have data buses to interface with sensor and actuators. Example data bus technologies are 1553B, 1773, Spacewire, 1394a/b, and USB. For many years Mil-Std-1553B has served as a popular data bus for space systems but has not kept pace with system bandwidth requirements. The European community and to some extent the NASA science community has adopted Spacewire as a data bus. The NASA human spaceflight community seems to be rapidly moving towards 400Mbps 1394b with and an accompanying protocol (SAE AS5643) for fixed-frame rate deterministic applications. There are far too many data buses to address in this paper, but it is fair to say that all standards have advantages as well as short comings, especially since many of them were not specifically designed for space applications. The area of space grade bus technologies requires further technology development. From a reuse perspective it is important to define your bus interconnect as a modular system (i.e. a card in a backplane). In the AAP example a 1553B/Spacewire card is underdevelopment. In this case the board level modularity allows the card to be removed when it is not needed and multiple boards can be added if additional bus interfaces are required. From an interoperability perspective, the bus protocol plays a significant role. Using protocols with sufficient definition to facilitate system interoperability is important. Bus standards may have several implementation options that require specification. Also, data buses may only define lower levels in a communications protocol stack (i.e. RS-422), therefore by themselves, interoperability is not guaranteed. Clear definition of all protocol stack layers will ensure interoperability between systems.

As just eluded to, standards go beyond hardware (electrical and mechanical). Many software standards exist (i.e. Portable Operating System Interface (POSIX), Internet Protocol (IP), XML, etc.). Space system designers are now comfortable with commercial real-time operating systems. Several commercial RTOS exist (i.e. VxWorks, Integrity). Military and commercial aviation have also accepted the use of the ARINC 653 software partitioning standard for running multiple applications of different criticality on the same physical platform. This standard provides time (schedule) and space (memory) partitioning with guarantees of application independence. The space community can leverage this standard to improve software reuse through modularity with well-defined APIs for communications. The partitioning standard then becomes an enabling technology for a middleware layers for common software functionality. Example functions are command interpretation, memory scrubbing, health management, and redundancy management. If partitioning is used and combined with common APIs, required functionality can be included during a software build phase, and then custom software functionality can be added to the common baseline.

V. Building in Variation

When designing for reuse it is important to consider variation in a design. A variation point in a design is a conscious design decision to add capability to a design so it can be used in more than one way. For example, the memory board in AAP is designed with flexibility so in can be used in multiple missions. One straight forward use is a solid state memory or recorder mapped to PCI address space. It is also designed to be a “ping-pong” memory for high-rate instrument data buffering. It provides for a flow control mechanism via user defined pins on the PCI bus, but uses zero-ohm jumpers to disconnect these signals when they are not needed. Lastly, its memories can be configured as TMR or EDAC protected based on mission data integrity requirements, storage requirements, and/or data rate requirements. The point is that features to facilitate reuse were thought out in advance and addressed in the design.

Another way to allow for design variation is through the uses of reconfigurable logic. Many reconfigurable devices exist today (i.e. FPGA, FPAA, FPTA, FPOA)[2]. Reconfigurable components can be a primary enabler for new classes of reconfigurable space system. One taxonomy of reconfigurable components defines: digital systems, analog systems, pathways, materials, and mechanisms[3].

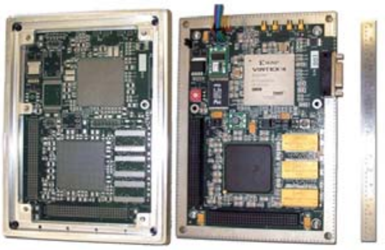

Figure 3. Reconfigurable board developed by NASA’s RSC project.

The FPGA are a good example of a reconfigurable component used in digital systems. In space system, one-time programmable anti-fuse FPGAs are frequently used. These devices allow for reconfiguration in the design phase but do little to allow for reconfiguration and reuse post-design or post-deployment. Flash and SRAM-based reconfigurable devices allow for both in situ reconfigurability and repeated reprogramming. These devices and systems, although clearly becoming more prevalent, are still more difficult to design with and less mature than other space-grade devices. These devices have found their way into some missions in low criticality systems and are actively being researched. Figure 3 shows is a prototype of a reconfigurable board in the SPACE-104 from factor.

Just like abstraction, layering, partitioning and common middleware can facilitate software reuse; similarly, reconfigurable logic modules, called IP (Intellectual Property), can be managed for reuse. One such mechanism is called a system-on-chip (SoC) bus. This is used to interconnect IP in a systematic way and allows for a common communication interfaces. Some examples include Wishbone, AMBA, and CoreConnect. Using a SoC facilitate modular design and reuse. An example IP repository built around the Wishbone SoC bus is OPENCORES (www.opencores.org). This site has Wishbone compatible microprocessor, memory controller, communication controller cores along with many others.

Reconfigurable devices with SoC buses and reusable IP are building blocks that can be used to build larger modules that can be reconfigured to perform multiple system functions and thus are reusable in multiple platforms. Unfortunately sometimes there are requirements for physical interfaces that cannot be accommodated by a reconfigurable device such as an FPGA. One approach to deal with physical differences is being taken by NASA’s Universal Reconfigurable Translation Module (URTM) project. This project proposes to develop Physical Interface Modules (PIMs) that translate the physical interface to a standard interface that can be used to connect to reconfigurable resources. Different PIMs can be attached based on the physical interconnect requirements of a given mission. The Figure 4 depicts the PIM concept.

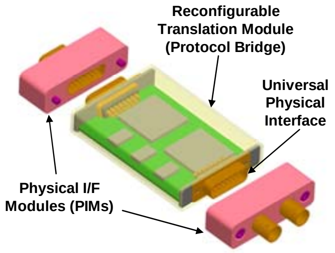
Reconfigurable Translation Module (Protocol Bridge)
Universal Physical Interface
Physical I/F Modules (PIMs)

Figure 4 . Reconfigurable physical interface modules currently under development by NASA UTRM project.

VI. Road Blocks

There are at least three types of road blocks on the road to reusable and interoperable systems. The first is technical. The technology needs to continue to evolve and mature. For the most part, maturing these technology for use in space systems is straight forward, not requiring radical break-throughs. The second class of road blocks is the engineering mind-set. There needs to be a recognition of value to meet requirements or to save costs before investment will be made in appropriate technologies. There also needs to be change in how designers attack a problem. Looking for commonality and reuse across a system, utilizing standards, and taking part in standard development are all important. Lastly, there are programmatic road blocks. Government often deals with contractors that develop proprietary solutions. This makes it difficult to share designs between projects. This leads to multiple independent solutions that stifle reuse and sometimes interoperability. Interestingly enough, the technical hurdles are often the easiest to overcome; changing the way people think and how large organizations do business is a much harder challenge.

VII. Conclusion

NASA's future missions will require both reusable and interoperable systems. It is unreasonable to assume that box-level line replaceable units (LRUs) will be available in an outpost on the Moon or Mars. NASA is beginning to take some steps to address this issue but technology investment has been limited so far. A few technology development efforts and reuse approaches have been briefly outlined in this paper. But in addition to technology development, there needs to be a change in the engineering mind-set and new programmatic approaches that facilitate and encourage reusable and interoperable design of space systems.

Acknowledgments

The author would like to acknowledge Willis Scott (AAP Project Manager), Kevin Somervill (RSC Lead Engineer) and Robert Jones (URTM Systems Engineer) for their contributions to this paper.

References

Periodicals

[1] Hodson, R.F., *SPACE-104: A stackable solution for space electronics, PC/104 and Small Form Factors, The Journal of Modular Embedded Design,* Winter 2006.

Reports

[2] Jones, R.L., Hodson, R.F., *A Roadmap for the Development of Reconfigurable Computing for Space*, Version 2.0, NASA, 22 March 2007.

Proceedings

[3] Lanza, D., Lyke, J., Zetocha, P., Fronterhouse, D., Mealanson, D., *Responsive Space - Through Adaptive Avionics*, AIAA 2nd Responsive Space Conference 2004.