AN OPEN SYSTEMS ARCHITECTURE FOR TELEMETRY RECEIVERS

Peter Parker, John Nelson, Mark Pippitt *
MIT Lincoln Laboratory

ABSTRACT

An open systems architecture (OSA) is one in which all of the interfaces are fully defined, available to the public, and maintained according to a group consensus. One approach to achieve this is to use modular hardware and software and to buy commercial, off-the-shelf and commodity hardware. Benefits of an OSA include providing easy access to the latest technological advances in both hardware and software, enabling net-centric operations, and allowing a flexible design that can easily change as the needs of customers may change. This paper will provide details of an OSA system designed for a telemetry receiver and list the benefits of OSA for the telemetry community.

KEY WORDS

Open systems architecture, software defined radio, telemetry receiver architecture, C-band transition.

INTRODUCTION

Throughout the history of wireless communications, innovation has always followed the demand for a more reliable link with the ever decreasing availability for spectrum. The evolution of the IEEE 802.11 standard [1] for wireless local area networking is a great example of this. This standard was originally designed to enable individuals to set up a low cost wireless network within the lightly regulated industrial, scientific, and medical (ISM) frequency band. Over the years, additions to the standard have provided for more bandwidth while at the same time being more robust to interference in the increasingly used ISM band. The challenging link budgets of deep space telemetry systems have also played an integral part of the innovation within wireless communications. The first major advances in coding theory were implemented in the Mariner deep space program. Continued development of coding theory was driven by additional deep space programs such as the Viking and Pioneer programs [2].

The basic challenges present for today's telemetry systems have not changed significantly from challenges faced in the past. First, demand for frequency spectrum continues to grow at the same time that available

*This work was sponsored by the Army under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

spectrum for telemetry purposes decreases. With increasingly complex test scenarios and an increasing number of sensors on test platforms, available spectrum is used up quickly. Second, more reliable and robust links are desired. A robust link operating in co-channel interference could go a long way toward alleviating the shortage of bandwidth. In addition, hypersonic and autonomous vehicles present challenges that can be mitigated through a more reliable link, particularly when the signal is experiencing multi-path fading. Lastly, with the looming spectrum reallocation, a system that is only minimally impacted by a future change in spectrum assignment could save time and money during the transition.

To be able to address each of the challenges listed above, innovation on both the transmit and receive sides of a telemetry system is required. Innovation is fostered in an open environment where multiple groups and companies with varied backgrounds can compete and showcase their ideas. This paper proposes an open systems architecture for telemetry systems which provides a flexible architecture with well defined interfaces that create an open environment for rapid innovation to meet current and future challenges. While this paper concentrates on the telemetry receive system, the general concepts of open systems also apply to the transmit system as well. The next section will describe the concept of open systems.

OPEN SYSTEMS

Acquisition reform thrusts and the proliferation of open systems (OS) and commercial off-the-shelf (COTS) technologies have prepared the way for major changes and cost reductions in the development process of defense-acquisition programs. However, OS are about more than limiting development costs. They speed up the development process and provide easy access to the latest technological advances. Further, OS facilitate the use of common architectures, alternate vendors, and a more competitive procurement model. A standard open architecture applied to radar systems has been shown to streamline the development process for these systems and greatly improve future technology-insertion opportunities [3].

An OS has several salient characteristics. It is generally a complex system that is made more manageable by breaking it down into subsystems, and then further into components. The smaller parts of the OS interact with each other in a predictable fashion that involves inter-component interfaces that are well defined and published without reservation. This approach allows individual parts, i.e., subsystems or components, to be replaced without affecting the remainder of the system as long as replacement pieces conform to the published interoperability behavior and interfaces. The decomposition of the problem described herein has major benefits. The sub-problems associated with the development of the parts become more manageable as fewer engineers and developers need to work on any given part. The parts are more easily tested, and multi-level testing (unit, component, integration, and validation) are easily carried out. Individual parts may be replaced by other like-function parts that share the proper behavior and interfaces. This factor allows the integrating entity of an OS to be a different entity than those that may have developed the individual subsystems or components, thus breaking down barriers to competition within a system development environment.

The openness of a system is determined largely by the level to which parts are described with respect to their interfaces. It is quite possible that an open system may contain some closed or proprietary parts as long as their function is well-known and understood and they obey the common interface definitions. An obvious example of this circumstance is the highly integrated electronic circuit – for a complex example, the CPU chip. The behavior of these chips is well understood publicly because the interface rules as well as the programming model are generally made public by the vendor. However, the details of the chip design under the hood are often held as a trade secret. This does not inhibit the use of the chip in an open system context in any way, and allows commercial entities to function in a particular and common OS business model.

When developing an OS it is desirable to ensure that the architecture can support the following important aspects. The architecture should be applicable to a wide variety of different device instantiations. These include, for example, ground-based telemetry systems, radars and optical sensors, dish and phased array radars, airborne sensors including Synthetic Aperture Radar (SAR) used for ground surveillance and other devices that use the notion of processing chains and open loop data collection or closed loop control approaches. Another important dimension for an open architecture to address is common support for diverse computing frameworks. There are several types of computing frameworks of interest in sensor and device development, including symmetric multi-processor computers, cluster computation, embedded device computation, and other specialized high performance computation including graphics processing units. An open architecture should support these different computing venues with minimal (if any) changes to software components as they are shifted from one compute platform to another.

Net-centricity is another equally important aspect of the OS. Generally the system must be able to accept commands, requests for behavior modification, as well as provide data and results to the outside world. Not only must the system be connected to networks, in many cases it should use the common technology being developed for the Web-centric world to enhance its usability and configurability. The Test and Training Enabling Architecture (TENA) [4] is beginning to be an enabling architecture for the net-centric aspects of an open system.

The other aspect of the OS that is very important is the availability of a library of component functionality that can be reused across projects and programs. This library availability is a key aspect of the OS because it will promote the reuse of the components that have generic functionality. For example, a bit synchronization module that maintains a clock synchronization on the received signal is a component with wide applicability. With a suitable library of components, system integration and testing is simplified as long as the components behave in a fashion consistent with the system engineering design. To allow this, it is important that the standards to which the components have been designed be robust. The component library can also serve as a baseline from which components can be taken and modified. This capability is enhanced if the components are based upon an object-oriented component model.

CURRENT TELEMETRY RECEIVE ARCHITECTURE

One example of a telemetry receive architecture block diagram is shown in Figure 1. The hardware may be classified in two groups: hardware located near the antenna, and hardware in a centralized location. The system receives and tracks an incoming signal, processes and records the data at various stages, and provides data to various consumers. We assume that any additional processing done on the data provided to the consumers is outside the scope of the telemetry receive system.

The antenna feed generally has two radio frequency (RF) outputs that provide right-hand circular polarization (RCP) and left-hand circular polarization (LCP) signals (or two equivalent polarizations). These signals are amplified by low noise amplifiers to ensure a good signal-to-noise ratio (SNR). A portion of the signal is fed to a tracking receiver which, along with the antenna control system (ACS), points the antenna in the direction of the signal.

In the case of a remote antenna site, the RF signal is up-converted to an optical frequency and transmitted across fiber to the processing center where it is down converted back to RF and sent through a multi-coupler to one of the receiver/combiners. If the processing hardware is in the same location as the antenna

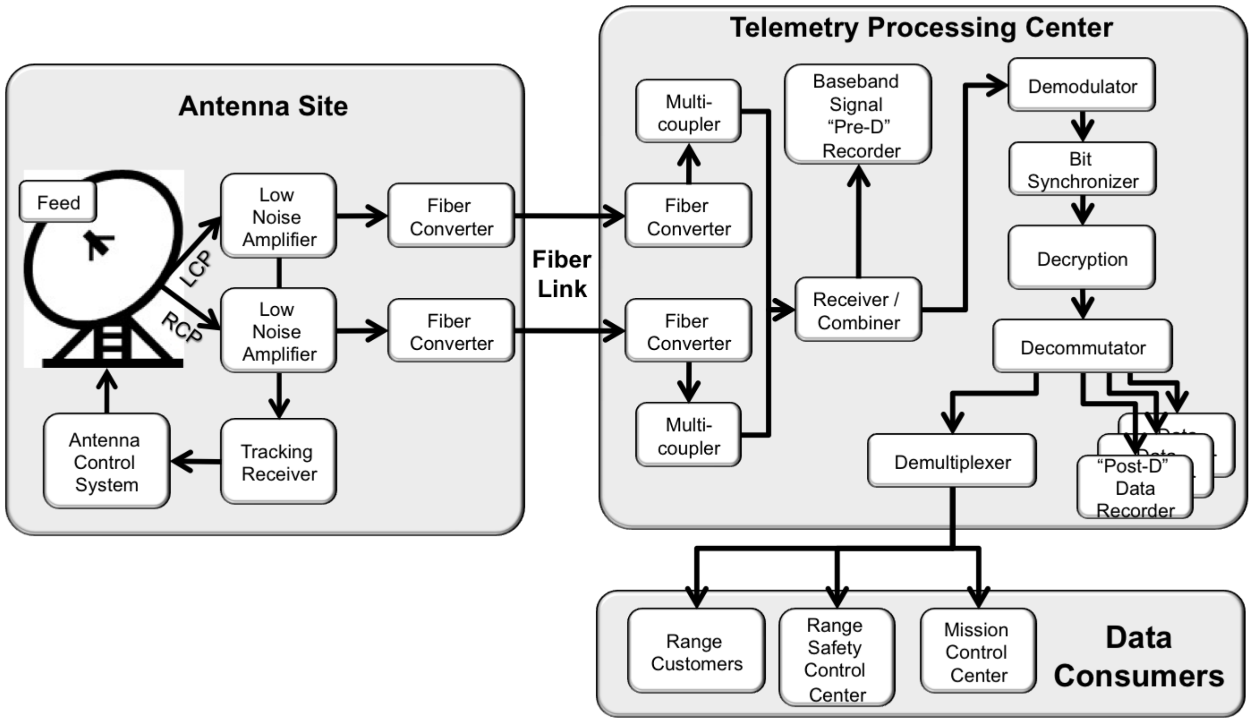

Figure 1: Block diagram of typical telemetry receiver system.

site, the fiber link and fiber converters are replaced with RF cabling straight into the multi-coupler. The receiver combiner then down-converts the RF signal to an intermediate frequency (IF), usually 70 MHz and this signal is sampled by an analog-to-digital converter (ADC). This sampled signal can be recorded by a baseband signal (“pre-D”) recorder if necessary. The digitized data are then demodulated, bit synchronized, decrypted (if necessary), decommutated, and demultiplexed. The data are then sent to the consumers. Most of the individual blocks in Figure 1 are implemented as dedicated hardware units.

In order to achieve a limited amount of flexibility in system configuration and to meet varying customer needs, there is a patch panel or digital switching matrix [5] which can connect any of the antennas to any of the receivers and include the necessary processing hardware into the processing chain.

PROPOSED OPEN SYSTEM ARCHITECTURE

Figure 2 shows a proposed telemetry receiver architecture based on the ideas of open systems architecture. The large block arrows represent the subsystem hardware interfaces and the small arrows within the backend processing subsystem represent the software interfaces. Each of these interfaces would be available to the public and maintained by a working group.

The antenna, feed, and ACS are structured in the same way as in the Figure 1 architecture. The rest of the system is significantly different primarily in how the functionality is split between subsystems. Breaking up the system into smaller subsystems (modules) enables greater innovation, faster upgrades, and a shortened procurement cycle.

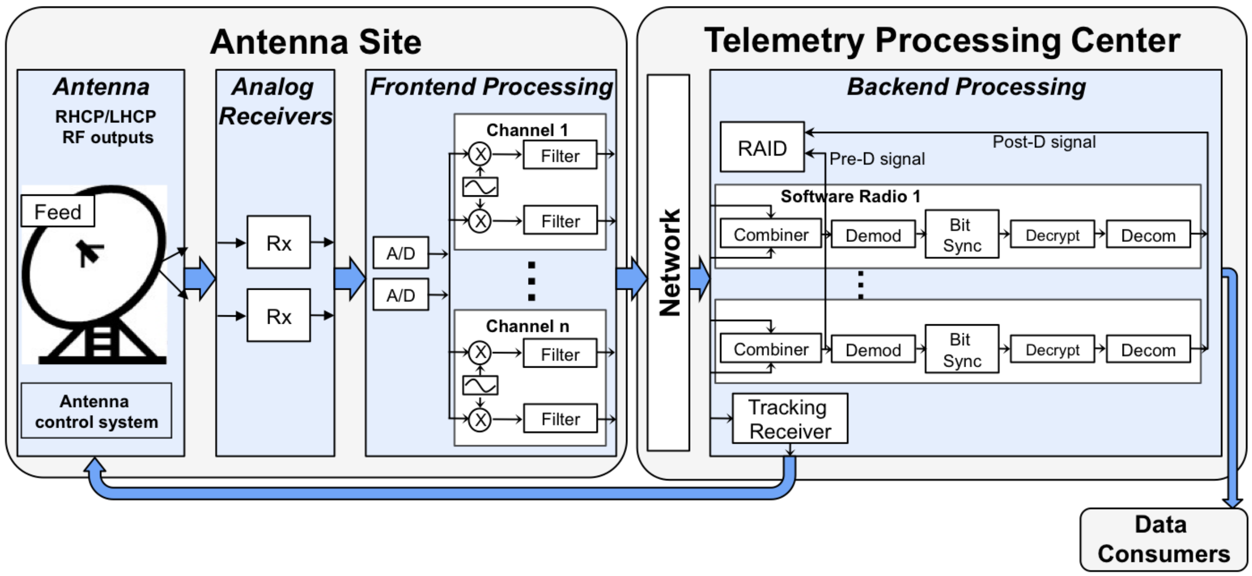

Figure 2: Block diagram of proposed telemetry receiver system based on an open architecture.

After the signal is received by the antenna, the analog receiver converts the incoming RF signal and block converts an entire band (approximately 100 MHz) to a common IF, *e.g.,* 300 MHz. Each of these signals is passed to the frontend processing subsystem where the 100 MHz band is sampled by an ADC (operating at greater than 400 MHz if the IF is 300 MHz). The digitized signal is sent to an FPGA which will down convert and filter around each individual telemetry signal within that band. Each of these signals is then sent across a data link to the telemetry processing center and made available on a network to any of a number of processing and recording servers. This network takes the place of the fiber converters, multi-couplers, and patch panels.

To maximize flexibility of the system, it is proposed that much of the radio functionality (demodulation, synchronization, etc.) be implemented in software.[1] The processing software would be written on top of an open systems software framework such as the ROSA II framework [6, 7]. Such a framework enables modularity in the code, and separation between the software and hardware. In addition to the data processing path implemented in the backend processing servers, a tracking receiver is also implemented in the backend. With access to the magnitude and phase of the data signals, a wide variety of tracking algorithms can be used to send control signals to the ACS. With the architecture of the network and software framework, the data are available at any point in the processing chain for recording. After the signal has passed through the entire processing chain, the data will be made available to the data consumers.

Comparing the telemetry open systems architecture of Figure 2 with the current architecture of Figure 1, the proposed architecture replaces many of the dedicated specialized hardware with commodity hardware and software and breaks the processing functions into smaller modules. The proposed architecture also reduces the amount of frequency specific hardware and eliminates frequency specific hardware in the telemetry center. This helps to reduce the amount of hardware impacted by any possible future spectrum realignments. In addition, this architecture simplifies the operations and maintenance by eliminating the patch panel and

[1] The software interfaces could be structured in a way to not preclude dedicated hardware to perform one or more of the processing blocks within the backend processing, *e.g.,* hardware decryptor.

enables the configuration of the system through software controls. By using commodity processing servers, much of the hardware maintenance personnel can be shared with the information technology group that each test range would have for maintaining other general computer equipment. Another benefit of commodity hardware is that networking and remoting software and protocols exist and would make the remote operation of a receive site a possibility.

Lastly, this architecture provides for the greatest amount of flexibility and options for including advanced techniques and algorithms. Due to the modular hardware and decoupled nature of the software and hardware, the development cycle for hardware and software are nearly independent. For example, when changing computing hardware, the application code remains nearly unchanged (in practice, about 1% to 5% of the code may need to be changed [8]). The fact that the application development team does not have to worry about hardware as much greatly decreases the algorithm development time and reduced time to technology insertion.

CONCLUSION

Open systems architectures are becoming widely used throughout the Department of Defense. These architectures have been used for radars, sonars, optics, and SIGINT systems. Innovation, and more importantly technology insertion, has increased for those systems with an open architecture. With the upcoming changes to the telemetry spectrum and the increasing demand for spectrum usage, a telemetry open systems architecture can provide the vehicle for the necessary changes to keep up with the increasing demands on telemetry.

REFERENCES

[1] IEEE, "IEEE std 802.11-1997," 1997. IEEE standard for information technology- telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements-part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications.

[2] D. J. Costello, J. Hagenauer, H. Imai, and S. B. Wicker, "Applications of error-control coding," *IEEE Trans. Inf. Theory*, vol. 44, pp. 2531–2560, Oct. 1998.

[3] S. Rejto, "Radar open systems architecture and applications," in *IEEE Radar Conference, 2000*, pp. 654–659, 2000.

[4] E. T. Powell and J. R. Noseworthy, "The test and training enabling architecture (TENA)." `www.tena-sda.org/download/attachments/6750/TENA-2012-Paper-Final.pdf`, 2012.

[5] M. Gilorma, "Patch panels vs. digital switching matrices," in *International Telemetry Conference*, 2011.

[6] J. A. Nelson, "Radar open system architecture provides net centricity," *IEEE Aero. and Elec. Systems Magazine*, vol. 25, pp. 17–20, Oct. 2010.

[7] S. Siegal and G. Schrader, "Parallel processing in ROSA II," in *Proceedings of the HPEC workshop*, 2009.

[8] A. Reuther, "Open architecture and middleware." private communication, 2011.