MANAGING INSTRUMENTATION NETWORKS

Eric Pesciotta
Director of Systems Software
Teletronics Technology Corporation
Newtown, PA USA

ABSTRACT

As traditional data acquisition systems give way to network-based data acquisition systems a new approach to instrumentation configuration, management and analysis is required. Today, most flight test programs are supported by traditional instrumentation systems and software. Pockets of network-based systems exist but are typically entirely new, closed systems. Relatively soon, test articles will emerge with a mixture of equipment. The merger of traditional and networked instrumentation is inevitable. Bridging the gap in software tools is a non-trivial task. Network-based data acquisition systems provide expanded flexibility and capabilities well beyond traditional systems. Yet pre-existing equipment requires traditional configuration and analysis tools. Traditional flight test software alone cannot fully exploit the added benefits gained from such mergers. The need exists for a new type of flight test software that handles existing instrumentation while also providing additional features to manage a network of devices. Network management is new to flight test software but a thoughtful implementation can facilitate easy transition to these modern systems.

This paper explores the technologies required to satisfy traditional system configuration as well as the less understood aspects of network management and analysis. Examples of software that meet or exceed these requirements are provided.

KEY WORDS

Network, Data Acquisition, Recorder, TCP/IP, SNMP, XML, XSL, Java, C++

INTRODUCTION

Traditional uni-directional serial (PCM) based data acquisition systems have been the standard of the flight test community for decades. Almost all major flight test programs active today use PCM based instrumentation systems. The technology to create, configure and manage these systems is well understood by most flight test engineers. Teletronics

Technology Corp. (TTC) has a comprehensive line of PCM based products and developed the TTCWare software specifically to work with these products.

While network-centric distributed data acquisition systems have become commonplace in other disciplines, airborne data acquisition systems have changed little in the last twenty years. Yet modern avionics and complex test articles are driving more demanding acquisition requirements. Requirements from both commercial and military programs are now exceeding the capabilities of PCM based acquisition systems. As a result, the network revolution has started to influence airborne instrumentation. Over the past few years, many papers have made the case for network based, distributed data acquisition products [1] and systems [2][3][4]. Papers are now appearing to document the successful deployment of such systems in both airborne and terrestrial programs [5]. Programs such as the Boeing 787 and US Army Future Combat Systems (FCS) have realized the benefits of network based acquisition systems.

Anticipating the demand for network based systems, the Network Products Division (NPD) at Teletronics has developed a complete line of rugged network devices for data acquisition, recording and switching. NPD products have been deployed in both airborne and ground based instrumentation systems. Since TTCWare was designed largely with PCM based systems in mind, it was not suitable for the network-centric product lines. Thus, NPD developed the Instrumentation, Configuration, and Management System (ICMS) to work with its network products.

With the development of both PCM and network based solutions minimal overlap between TTC product lines has occurred. Bulk data acquisition such as that characterized by bus monitoring fits well in a distributed, networked system. In such systems, multicasting makes efficient use of high bandwidth pipes (gigabit Ethernet). Data capture and recording devices with large amounts of random-access-memory (RAM) can sustain high traffic rates and large bursts. This is in sharp contrast to the comparatively low synchronized rates which characterize PCM based systems. The existence of both product lines and the uniqueness provided by each has led some customers to design hybrid systems. A hybrid solution can leverage existing inventories and provide modernization only where required. Regardless, with or without existing equipment, lower costs are associated with hybrid solutions versus an entirely new network based architecture. Hybrid systems create new and interesting requirements for configuration and management software. New tools are required for network design, configuration and management. Yet the desire for a well known and comfortable user interface is prevalent. The effort to consolidate and navigate between disparate software packages addressing different parts of a hybrid system is explored in this paper.

BACKGROUND

PCM Based Instrumentation Architecture

The components within PCM based systems are widely known to the flight test community and have well understood characteristics. These systems move data in a unidirectional, synchronous manner at rates up to 20 Mbps. Without the ability to designate classes of service, all data within the system is considered of equal priority. However, outside of telemetered data, the risk of data loss is slight. Time and control within these systems are not coincident with the data path. They are carried out of band on separate cabling to each component that needs them. Time is generally provided by IRIG and control provided via CAIS bus. Components within such a system are likely to include a PCM controller (CAIS master), data acquisition slaves (CAIS remotes), a telemetry transmitter, and a data recorder.

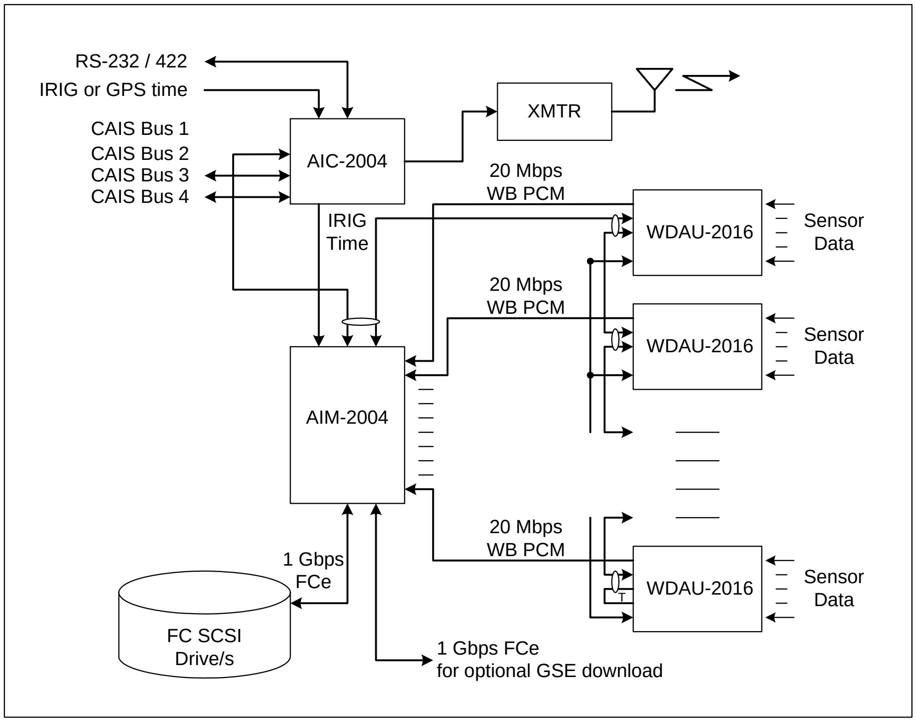

Figure 1: Sample PCM based instrumentation system

TTCWare - Software for traditional PCM based systems

Teletronics has developed Windows based desktop software, TTCWare, to configure traditional PCM based data acquisition systems. The functionality provided by TTCWare is extensive. Support for countless products and measurement formats is provided all through a standard Windows graphical user interface. The software can be used while connected to (online) or disconnected from (offline) the instrumentation. All user interaction occurs in the context of a TTCWare project. TTCWare projects are typically setup offline to mimic the hardware deployed on a test article. Projects define the hardware, sampled parameters, PCM formats, and more.

Beyond offline configuration, utilities are provided for programming and preflight checkout. TTCWare interacts with instrumentation systems via RS-232 or over CAIS bus (using an appropriate CAIS Bus PC adapter). When connected (online), TTCWare can query and report on installed inventory, program equipment, and monitor PCM data in real-time. Data portability is provided by importing or exporting XML files. However, XML is not the native format used for programming equipment. The programming process requires configuration compilation into files of proprietary format before being downloaded serially into each device.

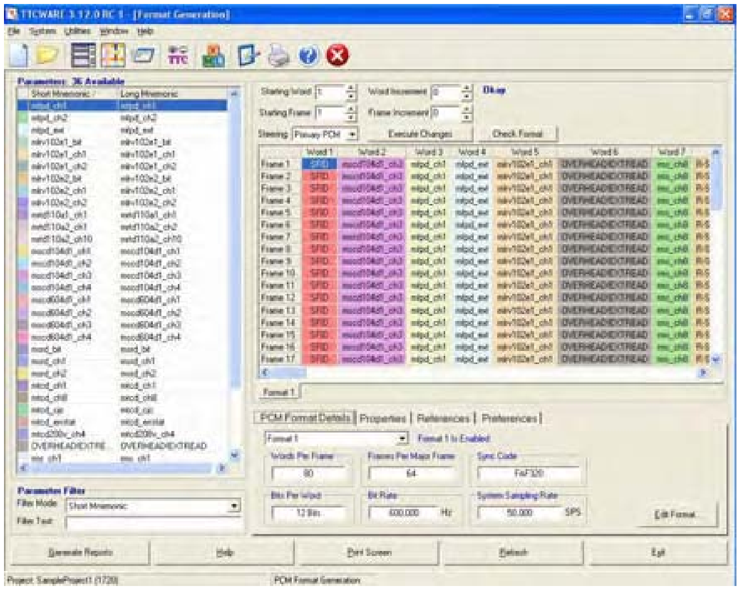

Figure 2: TTCWare – Building PCM Formats

Network Based Instrumentation Architecture

Network components are deployed around the globe and are widely understood. Their application in data acquisition networks is far less prevalent. Regardless, the flight test community can benefit greatly from the lessons learned by a world that has already created global networks. Network architectures can be characterized by their asynchronous, bidirectional, packetized data flow, potential for classifying levels of service, and transmission rates in the Gigabit range. Opaqueness of data is also a distinct network characteristic. Though a network may recognize different classes of service, it is entirely data type agnostic. All data is transported through these systems using standard internet protocols (IP). In a network based data acquisition system, time, control and acquired data are all carried across the same wiring. Components within such a system are likely to include; a network switch, distributed network data acquisition units, an IP recorder, and a data selection/telemetry system.

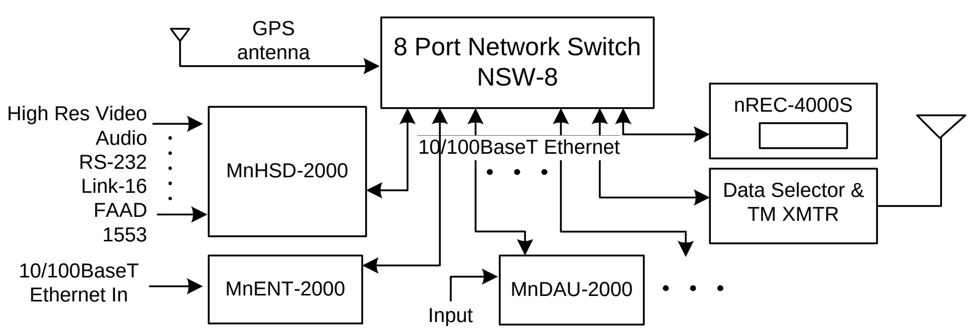

Figure 3: Sample network based system

ICMS - Software for network based systems

The Instrumentation Configuration and Management System (ICMS) is a relatively new development effort at Teletronics. ICMS is a platform independent solution for the configuration of network based acquisition systems. Both Windows and Linux operating systems are supported. The ICMS implementation has a flexible architecture based on proven web technologies, such as Java, Javascript, XML, CSS and XSLT. Like TTCWare, ICMS can be run online or offline. Again, all user interactions with ICMS occur in the context of a project and projects are used to mimic the hardware deployed on a test article. However, the similarities end there. ICMS projects work exclusively with network hardware and facilitate network design through a visual overview of the network. The visual overview provides a useful perspective on the instrumentation network currently not provided for PCM based products.

Beyond offline network design and configuration, ICMS provides online services such as programming, device browsing and network monitoring. ICMS interacts with the instrumentation network via direct connection to the Ethernet network. When connected, ICMS can query and report on installed inventory, program equipment, and monitor the network in real-time. Network protocols such as SNMP, FTP and HTTP are supported. ICMS projects are XML based and as such, data portability is native to the application.

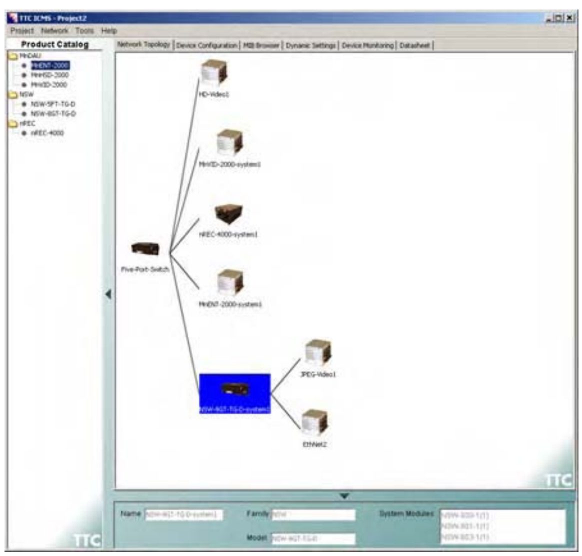

Figure 4: ICMS – Network Topology

Hybrid Instrumentation Architecture

Design proposals for hybrid instrumentation systems with components from both PCM and network based systems are starting to appear. The depiction of a large scale hybrid system may appear complex but is nothing more than the two disparate architectures joined at their borders by gateways. Gateways are network based devices that perform data format conversion. As data passes through a gateway, content and rates of data generally remain the same while the packaging changes to accommodate the destination transport. In the case of PCM, gateways can provide a path for PCM data to be packetized and transmitted across the IP network for recording or in-flight analysis. In the reverse direction, packetized network data can be transposed into a synchronous serial bitstream for telemetry.

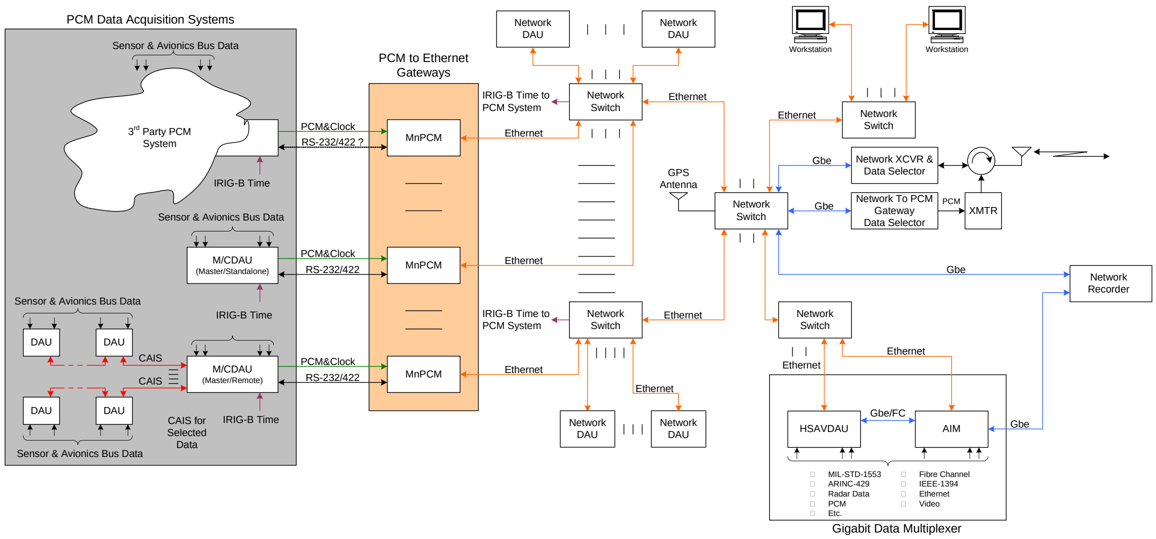

Figure 5: Large Scale Hybrid Instrumentation System

Software for hybrid systems

TTCWare manages PCM based devices up to and including the PCM gateway, while ICMS manages network based devices up to and including the PCM gateway. To date, a single software package that manages the equipment on both sides of a gateway does not exist. Nothing prevents a user from running each package independently to configure and manage their respective domains. However, Teletronics believes this is an unacceptable solution and is researching alternatives. This paper is in effect a status report on that effort.

THE CHALLENGES OF HYBRID SOFTWARE

The need for network management for instrumentation networks has been documented in the past [6]. The requirement for network management persists for hybrid systems as well. The need for PCM formatting, CAIS configuration, and PCM equipment programming also persists in the hybrid system. Thus, functionality provided by both TTCWare and ICMS is required to fully support a hybrid system. As can be observed in the screenshots above, each package has its own unique look and feel. Requiring two packages will complicate tasks and create a harder learning curve. The last thing a test engineer wants when dealing with an instrumentation system is additional complexity and doubt. Possible solutions to this problem include:

1) merge both ICMS and TTCWare into one package
2) replace TTCWare with ICMS

3) replace ICMS with TTCWare
4) replace both ICMS and TTCWare with a new package
5) exploit technologies for data and object sharing
6) abandon the notion of consolidation, run each package independent of the other

A Software Merger

It is quite natural to think about merging the two products. They are both designed to configure and manage instrumentation systems. Thus, it is easy to imagine commonality from a software development perspective. Unfortunately, the reality is not nearly that simple. As is apparent in the descriptions of each instrumentation architecture, the two systems are comprised of completely disparate technologies. Coincidentally, the same is true of TTCWare and ICMS. TTCWare is a mature product that has been evolving for over five years. Originally developed for Windows with Microsoft development tools, TTCWare has very little in common with the modern web development tools used for ICMS development. Although conceptually appealing, a merger of the two code bases is impractical if not impossible.

Replacement Strategies

Given the volatile and at times chaotic environment associated with flight test, the preference for familiar software is understandable. TTCWare has a large, faithful customer base built up over many years. Most TTCWare customers are fully accustomed to its interface and are highly proficient at building projects and configuring equipment. Giving up on this software in favor of a new package for networks would be disruptive to existing TTCWare users. This fact makes some favor TTCWare to replace ICMS. Unfortunately, as stated above, the technology used to create TTCWare is now dated and attempting to recreate ICMS functionality with older technology is not a task to be taken lightly. Even if the task could be completed within a reasonable amount of time, the result would be a software package already near its end of life.

ICMS was designed from the ground up with networks in mind. In fact, the development tools and technologies used are all deeply rooted in networking and the growth of the internet. As such, there is tremendous synergy between the ICMS software and the enhanced features required for network design and management. Quite naturally this synergy extends to the forthcoming systems proposed by iNET. Many of the iNET protocols are already supported by ICMS and it is fair to say that adoption of new iNET protocols will be quicker and easier in ICMS than in TTCWare. These arguments could lead some to suggest that ICMS might easily replace TTCWare. Based merely on the breadth of features encompassed by TTCWare, this suggestion should give any software engineer cause to pause. Adding the extensive hardware configurations supported by TTCWare into the equation makes the suggestion impractical at best.

Starting over

Refactoring is a popular term used in software development these days. To refactor is to reorganize the internal structure of software without changing its external behavior. To many, the challenges outlined above make the decision simple; start over by refactoring both applications into a single new development effort. Management tends to trivialize refactoring of this sort. At the same time the palpable desire to refactor is overwhelming to a software engineer. All too often this deadly combination leads to disaster. Countless mistakes can be made while refactoring, leaving the external behavior of the software changed for the worse. TTCWare supports an overwhelming array of features and equipment while ICMS supports somewhat complex protocols. Both TTCWare and ICMS products are shipping today. Both are far into their development life cycles. Replicating both development efforts again to support all products, features and protocols from both packages would likely be unreliable and counter-productive. In addition, the time to complete this effort would put delivery well beyond the timeframe required. Both products have a customer base depending on them to be reliable for critical test programs. Neither product can afford a delayed or buggy release.

Exploit technology

Application to application, or interprocess communication has been available for many years. Users of desktop office products exploit such technology routinely. In fact, as this document is authored, diagrams and figures are created via a drawing package that is more capable than the cryptic tools built into the word editor. When modification to a drawing is required, the word editor merely acts as a portal to the drawing software. The word editor appears as the main application, but merely proxies input/output to/from the drawing package. The drawing is actually modified by the drawing software and not the word editor. In Microsoft operating systems this is enabled via COM or .NET components. On open platforms, the very same underlying protocols, XML and SOAP, used by the .NET framework are available. Also available to Java applications is the Java Native Interface (JNI) which facilitates direct connections germane to a specific operating system. Thus, it is technically possible to open pipelines for communication between ICMS and TTCWare. These pipelines could provide tighter binding between the two applications, helping to alleviate the confusion caused by running both. TTCWare would remain responsible for all PCM configuration and management but would be ICMS aware. Conversely, ICMS would remain responsible for all network devices but would be TTCWare aware. When TTCWare required access to PCM equipment, it could proxy commands and responses through ICMS and across the network. Reversing roles, ICMS may encounter new hybrid equipment variants such as a network DAU that accepts standard PCM modules. In this case, ICMS would proxy configuration editing and compilation through TTCWare. Creating such an environment may require a fair amount of creativity but it is not out of the question and shows tremendous promise.

Last, to provide a seamless system programming solution, additional functionality is required in the PCM gateway. The gateway can act as a portal from the network side to the PCM side of a hybrid system. Accessing the PCM side of a hybrid system through the network is preferred for many reasons. Networks are faster and use standard, readily available protocols. Almost all laptops today (and the foreseeable future) have an

Ethernet port, while very few new laptops have RS-232 serial ports much less CAIS bus support. Fortunately, products such as TTCs MnPCM can act as a conduit providing a network portal into the PCM equipment.

Abandon consolidation

Naturally, this is the least desirable solution. Already chaotic, most test environments strive to reduce risk of mis-configuration and simplify processes wherever possible. Those goals directly oppose a requirement to run two disparate packages for configuration of a single system. In the past this may have been the only solution available. Fortunately, today's software development environments provide tremendous power and flexibility making this a very last resort solution.

CONCLUSION

The development of TTCware has been influenced over many years by its users and reflects a truly remarkable set of user-driven features. ICMS is a new and exciting addition to the Teletronics product line. Both packages offer unique qualities that make them invaluable within their respective domains. As such, leveraging the best of each to configure and manage hybrid systems remains highly desirable. Through the use of modern interprocess communication, web based technologies and new network elements, creating such a system is possible. In doing so, Teletronics can assure customers deploying hybrid systems that their test programs will proceed without risk in an efficient, stable environment.

REFERENCES

[1] John Hildin, Sergio Arias, "Airborne Network Switch With IEEE-1588 Support", Proceedings of the 42nd Annual International Telemetry Conference (ITC 2006), USA.

[2] John Roach, John Hildin, " An IP-Based Recording System", Proceedings of the 42nd Annual International Telemetry Conference (ITC 2006), USA.

[3] Albert Berdugo, John Hildin, "A System Approach to a Network Centric Airborne Data Acquisition System", Proceedings of the 43rd Annual International Telemetry Conference (ITC 2007), USA.

[4] John Hildin, "Network-Based Distributed Data Acquisition and Recording for Small Systems", Proceedings of the 43rd Annual International Telemetry Conference (ITC 2007), USA.

[5] Eric Pesciotta, John Roach, Nathan Sadia, Hsueh-szu Yang, "Networked Data Acquisistion Systems for the Army FCS Progam", Proceedings of the 44th Annual International Telemetry Conference (ITC 2008), USA.

[6] John Roach, “Design of a Configuration and Management Tool for Instrumentation Networks”, Proceedings of the 43rd Annual International Telemetry Conference (ITC 2007), USA.