METADATA MODELING FOR AIRBORNE DATA ACQUISITION SYSTEMS

Benjamin Kupferschmidt
Software Development Engineer
Teletronics Technology Corporation
Newtown, PA USA

Eric Pesciotta
Development Manager, Network Systems
Teletronics Technology Corporation
Newtown, PA USA

ABSTRACT

Many engineers express frustration with the multitude of vendor specific tools required to describe measurements and configure data acquisition systems. In general, tools are incompatible between vendors, forcing the engineer to enter the same or similar data multiple times. With the emergence of XML technologies, user centric data modeling for the flight test community is now possible. With this new class of technology, a vendor neutral, standard language to define measurements and configure systems may finally be realized. However, the allure of such a universal language can easily become too abstract, making it untenable for hardware configuration and resulting in a low vendor adoption rate. Conversely, a language that caters too much to vendor specific configuration will defeat its purpose. Achieving this careful balance is not trivial, but is possible. Doing so will produce a useful standard without putting it out of the reach of equipment vendors.

This paper discusses the concept, merits, and possible solutions for a standard measurement metadata model. Practical solutions using XML and related technologies are discussed.

KEY WORDS

Data modeling, INET Metadata, XML, XML Schema, XSLT, HTML

INTRODUCTION

Modern data acquisition systems are now employing XML in their configuration process. Indeed, the extent to which the ITC Proceedings reference XML grows consistently year to year. Recognizing this trend and the need to simplify the configuration process for test articles, the integrated Network Enhanced Telemetry (iNET) project has proposed the creation of a metadata standard for flight test telemetry. The paper, “Metadata For Range Telemetry”, presented at ITC ’06 made a convincing argument for the adoption of a metadata model. A working group with both vendor and range participation has been formed and work on a standard is well under way. The group is focusing on the creation of a unified standard for measurement metadata with the hope that standard XML instance documents will provide all that is needed to configure a test article (Figure 1). Through this effort, the group is attempting to transform test article configuration from today's "box" perspective to tomorrow's user-centric view.

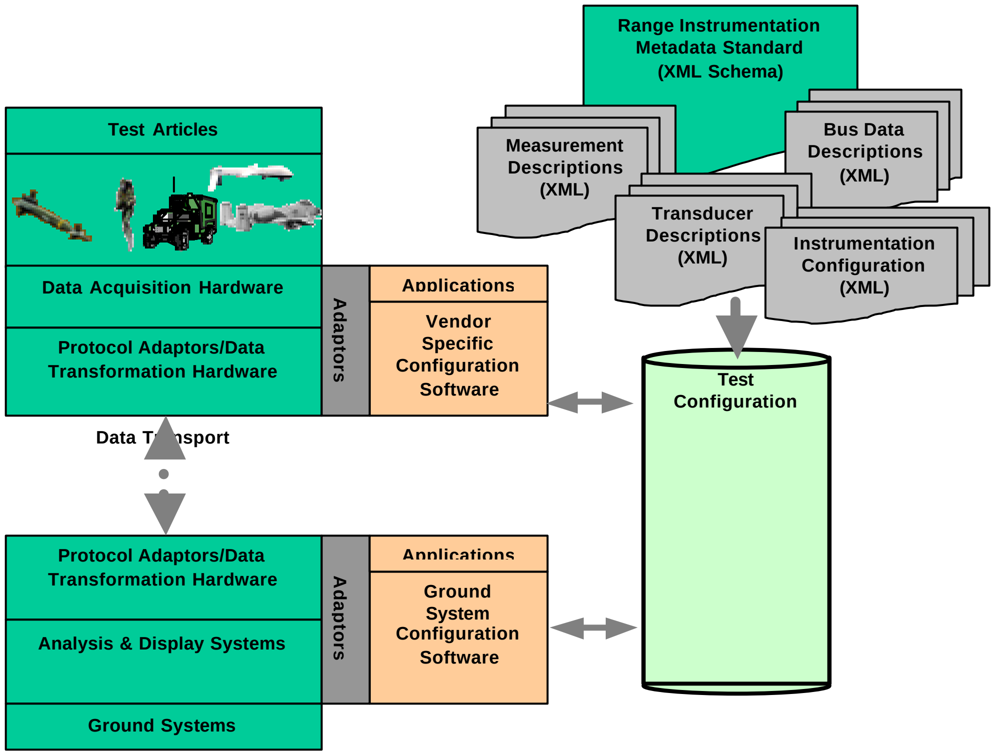

Figure 1. Proposed iNET Metadata Model

A test article with multiple vendor boxes, each with its own vendor specific configuration, clearly presents challenges and puts an unreasonable data entry burden on the flight test engineer. The flight test community will undoubtedly receive enormous benefit from the application of modern object oriented design techniques. However, the notion of describing a test article completely without box knowledge may prove to be too abstract. A fully automated process that translates abstract measurements into vendor specific box configurations will be difficult if not impossible to implement. For example, deciding on the capabilities needed to sample even a single analog sensor is very complicated due to significant overlap in capabilities across product lines. The complexity increases with the number of products considered for the task. As a result, choosing the best possible device for a given measurement may require user interaction.

This paper attempts to mitigate the ideals of the iNET metadata working group with the complexities inherent in test article configuration. First, a brief introduction to metadata, modeling, and XML are in order.

WHAT IS METADATA AND HOW IS IT USED?

Metadata is the term coined to refer to information that describes data. Simply put, data about data is metadata. Vast categories of data are possible targets for metadata. For example, in the context of a library where data resides in the content of a book, the metadata may include the title, author, publication date, and subject. In the context of a digital camera, a file containing a captured image is data, and the metadata might include the date and time of capture, camera settings (resolution and format), and perhaps a human description of the image content.

Metadata itself can be essential to the usefulness of the data it describes. As the number of books in a library grows, the data quickly becomes unmanageable and the books are of little use without a tool to find them (a library catalog). Similarly, files downloaded from a digital camera are useless without a tool that can render the original image (for example, a web browser). In each case, these tools rely on the associated metadata to do their jobs. In a library, metadata assists a human with the task of researching and locating a book. In the digital camera, metadata is used to assist a machine with the interpretation of data.

WHAT IS METADATA MODELING?

Modeling is the act of creating a model, which is a representation or imitation of something. It can be applied to many things in many ways. Modeling to hobbyists involves the miniaturization of a large and complex system (rockets, aircraft, automobiles, etc). To a hobbyist, the miniaturization process captures the essence of the larger system in a smaller and more manageable package. When applied to data, modeling techniques create packages of more manageable data. Metadata modeling is the process used to structure and organize metadata.

The output of the modeling process is a model theory describing the structure of data within a given domain. Representations for the entities within the domain, the attributes of those entities and the relationships between them are defined. Models may vary greatly in levels of detail and complexity. N modelers of the same domain may produce N models, each a byproduct of the modeler's perspective on the domain and their attention to detail.

MODELING TOOLS

The tools used in modeling vary greatly from domain to domain. For models that are machine oriented like those of data (and thus metadata), the language used to describe the model is critical. Standards such as Unified Modeling Language (UML), eXtensible Markup Language (XML) and eXtensible Stylesheet Language Transformations (XSTL) have been established to meet the needs of data modeling.

Unified Modeling Language (UML)

UML is a general purpose modeling language defined by the Object Management Group (OMG). It uses a graphical notation to create an abstract model of a system. The language provides a graphical representation of concepts such as classes, components, behaviors, aggregation and generalization. UML is the result of collaborative efforts by the pioneers in object technology: James Rumbaugh, Grady Booch, Ivar Jacobson, and others. With roots in object oriented analysis and design, UML is a proven technology commonly used by software engineers to describe software objects. Indeed, with the emergence of software code generators, the UML itself can be viewed as a development language.

eXtensible Markup Language (XML)

XML is a W3C recommended general-purpose markup language capable of describing many different kinds of data. It is a simplified form of the Standard Generalized Markup Language (SGML). Put simply, XML is a document format. It can be used to carry content for web pages, books, articles, manuals, etc. All manners of electronic files containing human readable content can be shared through the use of XML. In its more advanced application, XML itself is used to describe more specialized markup languages. Derivative languages such as, XHTML, XML Schema, XSL, and MathML are all based on XML. The flexible nature of XML has led it to become an extremely popular document format. XML parsers are readily available on most computing platforms.

eXtensible Stylesheet Language Transformations (XSTL)

The eXtensible Stylesheet Language (XSL) is a family of W3C languages allowing one to describe how XML files may be (re)formatted or transformed. XSL Transformations (XSLT) is the member of this family that translates XML files from one format to another. The inputs to an XSLT processor are always an XML file and XSLT stylesheet. The output can be any text file, not necessarily XML based. Many modern day operating systems have built-in XSLT processors. Both client (web browser) and server sides (web server) can take advantage of XSLT processing. Any XML file that references an XSLT stylesheet can be automatically transformed into an HTML document for display.

APPLYING METADATA MODELING TO FLIGHT TEST

The flight test instrumentation industry is well positioned to take advantage of the power of metadata modeling techniques. There are many vendors in the industry who produce similar, but incompatible products. Customers often want to use these products together but there is no easy way to do so. The primary reason for this incompatibility is that each vendor's setup software only supports the hardware that they manufacture. This makes it difficult to create a single file that contains all of the information needed to setup the entire system.

The principal goal of the flight test industry's metadata modeling effort is to create a universal language that can describe most flight test instrumentation systems. There are many potential benefits that can be realized by such a language. The metadata language will allow end-users to generate a single file that contains a list of all of the measurements that they want to collect from a data acquisition system. This makes it relatively easy for end-users to change the configuration of measurements. It also allows the end-users to reuse their measurement lists on future projects and to archive old configurations for reference purposes. In addition, since the metadata language is based on XML, it will be possible to use COTS tools to verify that a file complies with the metadata language's schema.

In order to use the metadata language, vendors will need to build translators into their software applications. This will allow end-users to create configurations directly in the metadata language and then import them into the vendor's software to compile the configuration and program the hardware. This ability will help users to save time by eliminating duplicate data entry in multiple software applications. Eventually, it might even be possible to directly program data acquisition systems with files written in the metadata language. The hardware would read the file and determine how to set itself up based on the required measurements.

A metadata language for data acquisition systems will help to facilitate interoperability between the various components of a telemetry system. It will also help systems that are manufactured by different vendors to work together. Another benefit is that the metadata language can be used to transfer a system configuration between different test ranges that use incompatible software. Ideally, the metadata language would be designed with extensibility in mind so that new items can be added without breaking any existing features. This will allow users to gradually upgrade their systems based on their needs without having to discard older but functional equipment simply because it is incompatible with the newer software.

Essentially the metadata language adds a layer of abstraction that isolates the end-users from having to deal with the configuration details of the data acquisition system. This frees them to focus all of their efforts on actually acquiring the data that is needed to analyze a flight test.

Despite the many benefits of applying metadata modeling to flight test instrumentation, there are several potential drawbacks that need to be considered. The most significant problem is that both vendors and end-users must agree to use the metadata language. Someone will have to implement translators between each vendor's native setup format and the metadata language. Translators will also be needed to convert from the internal measurement databases, which are used by many large flight test programs, to the metadata language.

The metadata language will need to be well documented in order to prevent vendors and users from misinterpreting the tags in the language. If different vendors implement the language in incompatible ways then the usefulness of the language is greatly reduced.

The use of XML validation tools can help to mitigate this problem. However, the best solution to this problem is the creation of an official metadata language validation tool. The group that is responsible for designing the schema for the language should create this validator.

Another potential difficulty relates to making changes to the metadata language. All changes must be made while keeping in mind both forward and backward compatibility. Structures that are being eliminated from the language should be deprecated rather than deleted from the standard. In addition, whenever possible, changes should be designed so that older interpreters can safely ignore them. A key feature that can be designed into the interpreters is the ability to save the contents of unsupported tags when importing a metadata language file. This will make it possible to regenerate these unsupported tags when the configuration is exported to the metadata language. The main reason why this feature is advantageous is that it makes it possible for older software to preserve newer content that it doesn't understand.

CURRENT DATA ACQUISITION SYSTEM MODELING LANGUAGES

There have been a number of previous attempts to create a generic language for representing the configuration of a data acquisition system. These languages have typically been designed to serve as an intermediate layer between two incompatible pieces of software. In order to pass a configuration between two software applications, a conversion tool must be written for both applications. This conversion tool must be able to bi-directionally convert the application's native format to the generic language. Most recent attempts to create generic languages for data interchange have leveraged non-proprietary XML technologies. Since XML is a mature general-purpose data storage language, there are many tools that can be used to work with XML files. It's also possible to easily validate XML files against a language schema and to transform them via XSLT transformations.

The most commonly used data exchange format in the flight test instrumentation industry is the Telemetry Attributes Transfer Standards (TMATS). TMATS is an industry standard that has been designed to facilitate the transfer of data acquisition system configurations between users and test ranges. TMATS is incorporated into the IRIG-106 standard as Chapter 9. The biggest advantage that TMATS has is its incumbency. It has been a standard for longer than the other current languages and most current data acquisition software can read and write TMATS files.

TMATS has also been incorporated in the IRIG-106 Chapter 10 recorder standard. The recorder standard requires that the first packet in each recording contain a TMATS file. This TMATS packet describes the contents of the Recording. This helps users to play the data without needing the original data acquisition setup files.

There are several issues that have prevented TMATS from fully realizing its goal as a data interchange format for any telemetry system. The most significant problem is that

TMATS is poorly documented. The valid range of values for many tags is not clearly defined. This results in each vendor implementing a slightly different interpretation of the standard, which limits the ability of end-users to actually exchange TMATS files. This problem is difficult to resolve because there is no universally accepted validation tool for TMATS files.

Another major problem with TMATS concerns the structure of the metadata files. TMATS is not based on XML. Instead, a TMATS file contains a list of tag-value pairs terminated by semi-colons. Each tag consists of a set of cryptic names that identify the property that is being set and a corresponding value. Properties that refer to the same object in the data acquisition system are linked via a sequentially assigned index number. This rather awkward system makes it very difficult for humans to interpret a TMATS file. It also makes it tricky to manually modify a TMATS file. Adding to this problem is the fact that values are stored in inconsistent formats throughout TMATS.

To address some of these issues, the RCC/TG's Data Multiplex committee recently created a new XML version of TMATS. This new standard maps all of the tags from the TMATS standard into an XML schema. It has several crucial improvements including the elimination of the counter indices and the extension of the tag names to improve readability. The hierarchical nature of data acquisition systems is expressed much more clearly in the TMATS XML format than in the classic TMATS format. All of the older PCM Format representations were also discarded in favor of the Word-Frame method.

The biggest problem with the TMATS XML format is that it is brand-new. It is currently unclear if the new standard will achieve widespread acceptance and use. Many vendors will not want to invest the time needed to add support for TMATS XML unless they know that end-users intend to use it.

In addition to the aforementioned industry standards, several vendors, including Teletronics, have defined proprietary XML based languages. These languages are mainly designed to help users build large projects without having to manually create thousands of measurements. The main advantage of these proprietary languages is that they are tightly coupled to the data acquisition hardware and software of the vendor that created them. This allows the languages to be designed in a much clearer and concise manner. The languages can also be much more flexible because the creator has complete knowledge of the systems that are going to be described by the languages. The problem is that these languages can only be used with the hardware and software of the vendor that created them. This makes it very difficult to use these languages for general-purpose data exchange.

PRACTICAL METADATA IN CURRENT APPLICATIONS

Large programs, both commercial and military, are already leveraging metadata modeling and translation techniques. One such system, the Boeing 787 has perhaps gone farther and faster than any other. The system devised for the 787 has modeled an entire network of avionics, measurements, and data acquisition equipment in XML. Their schemas are the intellectual property of Boeing and cannot be discussed here. However, it is important to mention since it proves that the lofty goals of the iNET metadata working group are nearly achievable today.

The Joint Strike Fighter has also made great strides in automating test article configuration through metadata (Figure 2). Fortunately, this system has been publicly documented. Rather than using XML instance documents, this system captures ICDs and measurement metadata in a database. This database, the Flight Test Data Center Database (FTDCDB), maintains the latest bus catalogs including those for 1553, 1394, and Fibre Channel. Interaction with the database is facilitated by Lockheed Martin proprietary software. This software can generate the Master Measurement List (MML), Test Configuration List (TCL), Test Measurement Request (TMR), Test Data Request (TDR), and others. All of which can be considered metadata.

In a joint effort between Lockheed Martin and Teletronics, software tools were created to export XML from the FTDCDB. One such XML file is used to directly configure the data acquisition hardware. This export tool can be likened to an adaptor application as envisioned by the iNET metadata working-group.

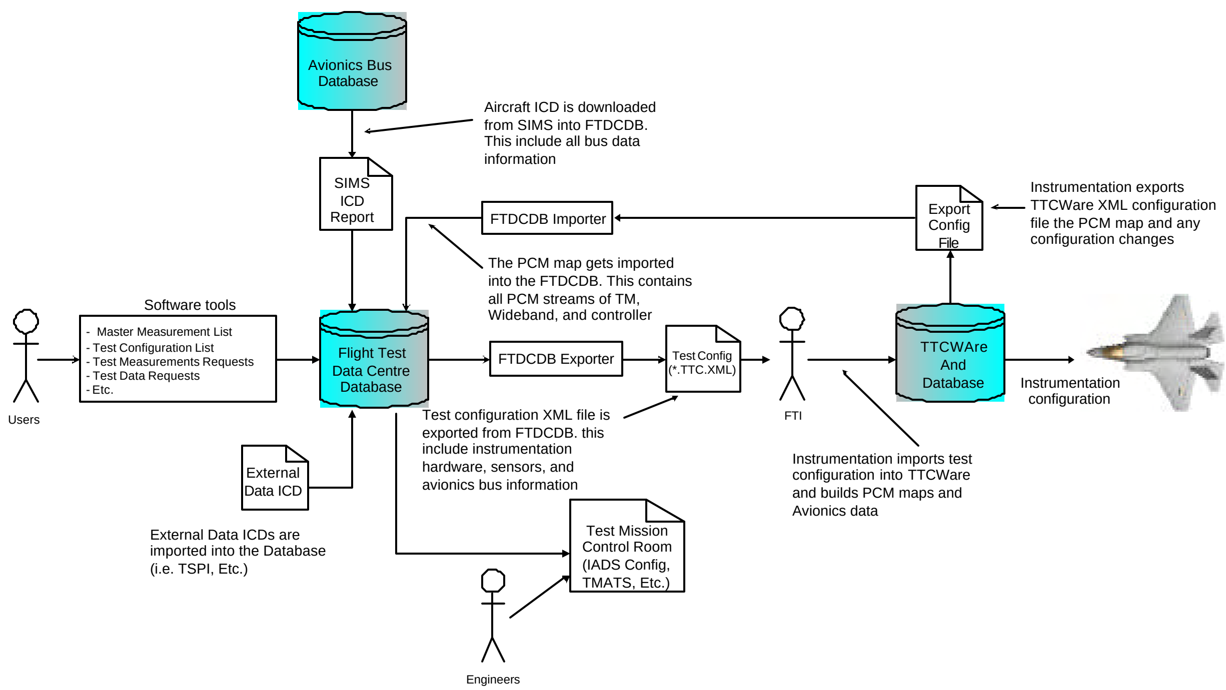

Figure 2. JSF Preflight Configuration Process

These modern systems demonstrate that large-scale automated test article configuration is possible through the use of metadata. Both systems discussed were designed and have been successfully implemented in a multi-vendor environment. This proves that the metadata working group should be encouraged to pursue their ideal. However, in each system, the modeling had been undertaken with significant domain knowledge of the test article and pre-selected data acquisition equipment. The subsequent metadata theories created by this process were not auto-generated. In fact, they were created with a great deal of human interaction.

A NEAR TERM METADATA SOLUTION

Attempts by the iNET metadata working group to apply work-in-progress model theories to actual data acquisition systems have been only partially successful. Much of the definition is incomplete and many assumptions must be made before a given vendor can truly auto-generate a configuration. Even with a fully defined standard, writing adaptors has proven to be a non-trivial task. In practice, adaptors will likely be XSLT translations (Figure 3). Teletronics attempted to write such an adaptor from a working group example using simple textual replacement and found it was insufficient. It was clear that a complex exemplar based translation was necessary. This type of translation requires skill and time. In addition, many assumptions about recording rules, port assignments, and telemetry formats were required. Investing in a complex adaptor at this stage of the standard was deemed premature. Regardless, the model proposed by the iNET working group is principally sound.

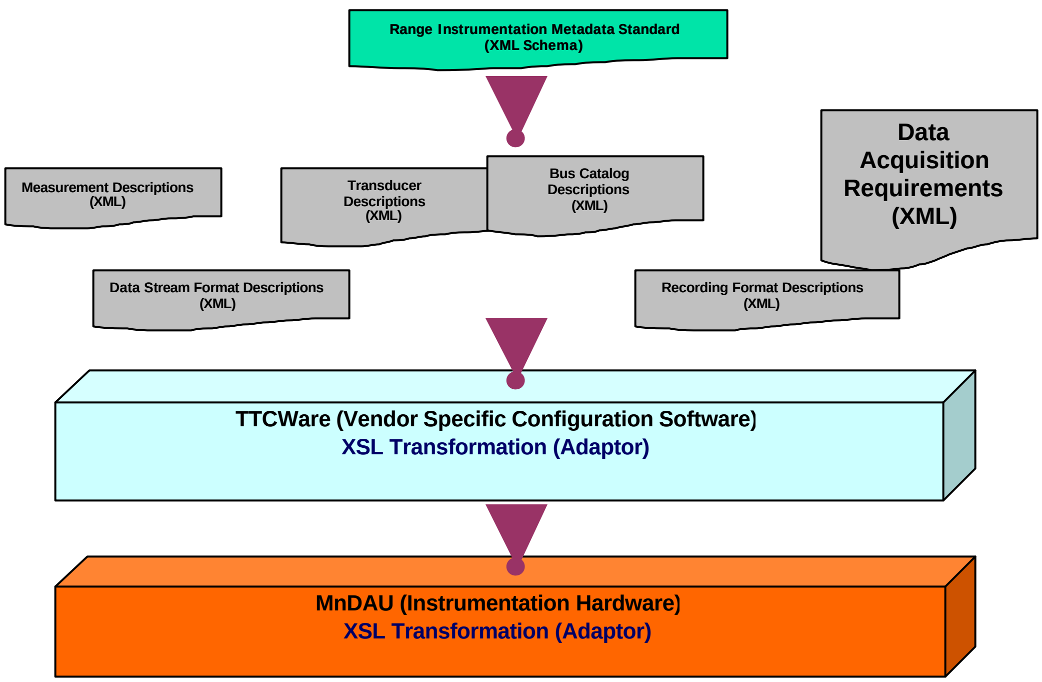

Figure 3. Applied use of the iNET Metadata Working Group Model

It is possible to bridge the gap between the program specific visions presented above and the proposed working group model. To do this a software tool is needed that combines measurement metadata with vendor specific XML instance documents to create configuration files that can truly configure a data acquisition system without an inordinate number of assumptions. With this tool, rather than instantly flipping the flight test paradigm from "box view" to "measurement view", a more sideways perspective is taken. When viewed from the side, measurements are described in a standard language by a user with domain knowledge of the test article. Measurements are then bound to the data acquisition equipment via software run by a user with domain knowledge of the vendor hardware. With this perspective, any ambiguity is removed from a measurement abstraction when the user binds it to a port or channel in the data acquisition system. Although user interaction is required, it does not need to be cumbersome or complex. In fact, vendor independent tools could be used to create or parse the standard measurement metadata and present a collection of measurements. If these tools support drag and drop features, measurements could simply be dropped into vendor specific configuration tools. This object exchange between applications would provide vendor tools with metadata in a standards-based format. Any additional information required to complete the binding would be entered directly into the vendor software. One such product under development at Teletronics, the Instrumentation Configuration and Management Server (ICMS) is taking this perspective. This viewpoint allows the software to be released ahead of the eventual metadata standard without sacrificing future compatibility.

CONCLUSION

The iNET metadata working group has taken up a challenging task that is well worth pursuing. However, its lofty goals are untenable in the near term. Taking a "side view" of the test article instead of a "box view" or a "measurement view" results in a half step towards a fully automated system configuration. This approach will produce results quicker and easier than a full implementation of the iNET metadata working group model theory. It will also achieve greater data portability and interoperability between vendor software without putting the task out of reach in the near term.

REFERENCES

1. Thomas Grace, and Clay Fink, "Metadata for Range Telemetry", Proceedings of the 42nd Annual International Telemetry Conference (ITC/USA 2006), USA, October 23-26, 2006
2. Doug Vu, and Albert Berdugo, "Overview of an Instrumentation Data System Used by the JSF (F-35) Test Flight Program", Proceedings of the 43rd Annual International Telemetry Conference (ITC/USA 2007), USA, October 23-25, 2007
3. John Roach, "Design of a Configuration and Management Tool for Instrumentation Networks", Proceedings of the 43rd Annual International Telemetry Conference (ITC/USA 2007), USA, October 23-25, 2007
4. W3C, World Wide Web Consortium, "XML Base", www.wc3.org, 27 June 2001
5. W3C, World Wide Web Consortium, "Extensible Stylesheet Language (XSL)", www.wc3.org, 15 October 2001
6. Object Management Group (OMG), "Unified Modeling Language", www.omg.org, 2007