

# "ACHIEVING INTEROPERABILITY THROUGH SEMANTIC TRANSPARENCY "

**Rodney Davis, Greg Hupf, & Chad Woolf**

Command and Control Technologies, 1425 Chaffee Drive, Titusville, FL 32780, 321-264-1193  
davisrd@cctcorp.com

## **ABSTRACT**

CCT is conducting research to provide a cross platform software capability that enables a common semantic for control and monitor of highly distributed systems-of-systems C<sup>2</sup> architectures by auto-generating semantic processing services from standardized metadata specifications. This new capability is significant because it will reduce development, operations, and support costs for legacy and future systems that are part of ground and space based distributed command and control systems. It will also establish a space systems information exchange model that can support future highly interoperable and mobile software systems.

## **KEY WORDS**

Interoperability, Semantics, XML, Telemetry, Command

## **INTRODUCTION**

The current lack of standardization requires custom ingestion of telemetry and commanding structural and semantic information across a variety of systems and organizations throughout a mission lifecycle. Additionally, many of these organizations must support multiple heterogeneous missions using a common ground segment infrastructure. This is made difficult and costly because standardized and automated methods are not used for communicating this information. The current lack of adequate standardization for general control and monitor data exchange results in a proliferation of custom ingestion and processing application programs. This

customization is inherently error-prone, resulting in the need to revalidate the data representation at each transition in the lifecycle. Mission operations would be more efficient if consistent telemetry and command definitions could be easily shared among all of the lifecycle phases, systems, and organizations. In addition, the existence of a standard description language could also enable software commonality across communicating nodes. The emerging OMG XML Telemetry and Command Exchange (XTCE<sup>1</sup>) standard is based on XML technology that has the potential to fill the standardization void; however, successful interoperability requires adaptation of a variety of legacy systems. Without supporting tools that ease the burden of adopting a new standard it may be cost prohibitive for the US space systems infrastructure to take advantage of it.

## **Background**

Future space systems (spaceport, spacecraft, launch vehicles, ranges) will need to be able to operate adaptively, with more autonomy than systems do today. In the future, concepts of ubiquitous communications infrastructure, dynamic service discovery, and mobile agents will enable loosely coupled systems to collaborate to establish objectives and achieve broad mission goals. A critical stepping-stone in realizing this future is establishing a vocabulary and mechanisms for exchange of a wide range of configuration, control, and instrumentation information. This potential stepping-stone to the future is a very real challenge for our space systems infrastructure today.

Spacecraft design is performed today through the use of a number of disparate tools and techniques. Interface design for space systems is manual and time consuming. Data design, both telemetry and commanding, is performed multiple times by multiple contractors during the vehicle lifecycle, well before the systems are deployed for mission operations.

Similarly, mission operations require the exchange of telemetry and command information across multiple heterogeneous missions using a common ground segment infrastructure. Information must be exchanged among all of the operational phases, systems and organizations. This is made difficult and costly because there is no standard method for exchanging this data definition information. The lack of standardization currently requires custom ingestion of the telemetry and commanding information. This customization is inherently error-prone, resulting in the need to compare and revalidate at each step in the lifecycle.

A typical example of this process is between spacecraft or vehicle and the operating organization. The manufacturer defines the telemetry and command data in a format that is much different than the one used in the ground, or spaceport, segment. This creates the need for database translation, increased testing, software customization, and increases probability of error. Standardization of the telemetry and command definition format will streamline the process allowing dissimilar systems to communicate without the need for the development of mission specific database import/export software.

---

<sup>1</sup> Object Management Group, XML Telemetry, and Telecommand Data Specification – Draft Adopted, <http://www.omg.org/docs/dtc/03-05-07.pdf>, May 2003

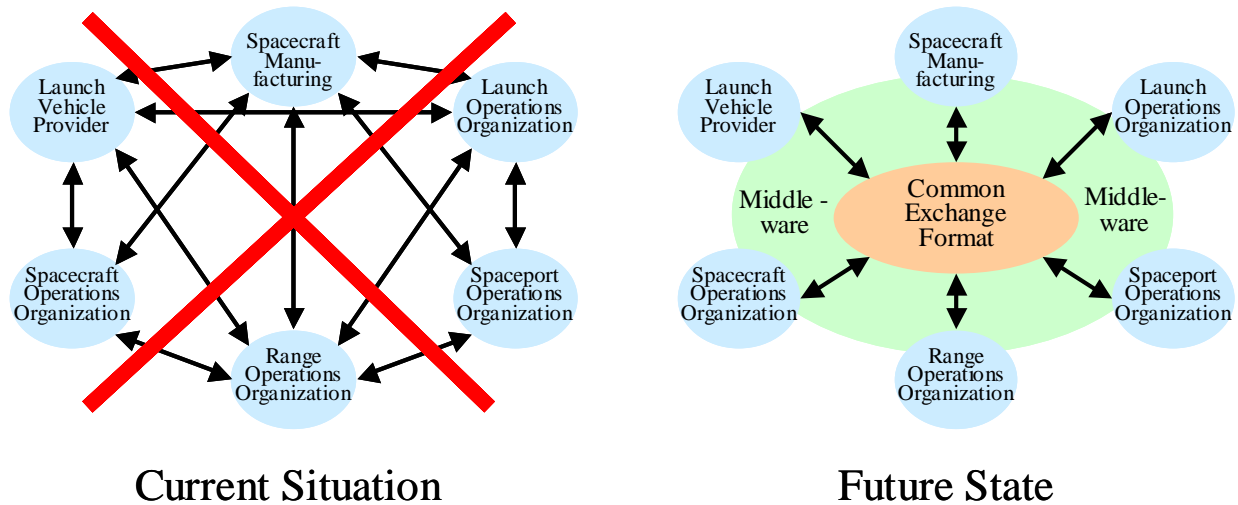


Figure 1 Research results

Ideally, a space system operator should be able to transition from one ground system to another by simply moving an already existing command and telemetry database that is compliant with the standard control and monitor specification. As illustrated in Figure 1, there currently is a need for  $N(N-1)$  number of interfaces between producers and consumers of telemetry and command description information. The proposed approach seeks to produce a more manageable strategy that requires  $2N$  potential interfaces. This means a smaller number of interfaces to develop and maintain, and increased flexibility and utility in between the producers and consumers.

A concrete example of the problems facing current space systems infrastructure can be seen through the NASA ELV telemetry operations at Cape Canaveral Air Force Station and Vandenberg Air Force Base. The NASA telemetry organization deals with a broad range of telemetry users and producers. Launch vehicle providers, payload organizations, the Eastern and Western Ranges all have a myriad of unique telemetry ground systems and unique strategies for describing telemetry format, and content. Exchange of format and content information is a standard part of everyday business. The NASA telemetry organization is faced with creating custom software for each unique organization and mission that supports exchange of information as well as binding mission databases to NASA legacy telemetry infrastructure. As a middleman in the exchange process, NASA would benefit tremendously if all of producers and consumers would standardize on how they exchange telemetry databases. However, adopting a standard is a non-trivial exercise, especially considering the various organizations involved, already have a significant investment in their current methods, and changing to a new strategy will require new software and processes. The problem, and possible price for a solution, is exacerbated by the diversity of systems, software, operating systems, etc. in use. It makes a single, monolithic solution/tool unpractical.

It is very easy to see how this same example scenario can play out in any number of possible current and future space systems infrastructures where heterogeneous systems need to interact. Hence, a fundamental requirement for the interoperability of current and future space systems is

to establish a common language for describing how systems communicate. An information model is needed that facilitates common understanding of structure and content for control and monitoring systems. Emerging standards, such as XTCE, offer a part of the potential solution by identifying common mechanisms for representing form and content. However, standards alone will not be sufficient to solve the problem. Software required to accommodate these new and complex standards will be expensive, especially if the standards are to be adopted by the myriad of legacy systems owned and operated by the government and industry. Adopters of these standards could be faced with spending hundreds of thousands of dollars per system, which will aggregate to many millions of dollars across the space operations enterprise. Robust, adaptive software tools and/or services are needed to reduce the cost and complexity of adopting the new standards.

## **BODY**

Widely distributed defense operations or even a sustained effort to explore the moon, Mars and beyond will involve simultaneous operation of a network of systems that must be continually managed via systems-of-systems command and control (C2). Creation of this C2 network without exploiting cross enterprise commonality will be cost prohibitive, as well as limit interoperability and mission effectiveness. C2 interoperability depends on standardization of protocols, semantics, and knowledge across disparate systems. Currently, little commonality exists between C2 nodes even though systems share a common domain ontology. CCT's research will use a formal cross enterprise ontology to auto-generate cross platform information processing software, suitable for modular integration in any weapon, spacecraft, ground system, or embedded C2 system. The resulting technology will significantly reduce development, and recurring costs for legacy and future C2 systems, as well as enhance interoperability.

The technology development approach is to implement a software capability for mapping Exploration C2 domain knowledge (protocols, messages, algorithms, sensors, stimuli, behaviors, etc) in to software technologies (requirements, models, classes, objects, processes, etc.) using a formal metadata specification and auto-generation methods. The auto-generation strategy will be implemented in conjunction with a standards based C2 domain reference architecture and Product Line strategy.

Based on the premise that interoperating systems must share a common ontology as described in Figure 2; using a rigorous metadata specification based on XML schema, such as the emerging XTCE standard, it is feasible to auto-generate stream, protocol, and semantic processing services directly from the metadata specification. The auto-generator, illustrated in Figure 3, will produce modular code or libraries that are integrated into C2 mission application software statically, or dynamically. This is useful because it enforces a common semantic, reduces amount of custom SW required to create new or adapt legacy systems, produces very compact and efficient SW for real-time systems, it is ideal for embedded applications, improves SW quality, and is ultra consistent. It is also supportive of Model Based Architecture & Product Line Practice.

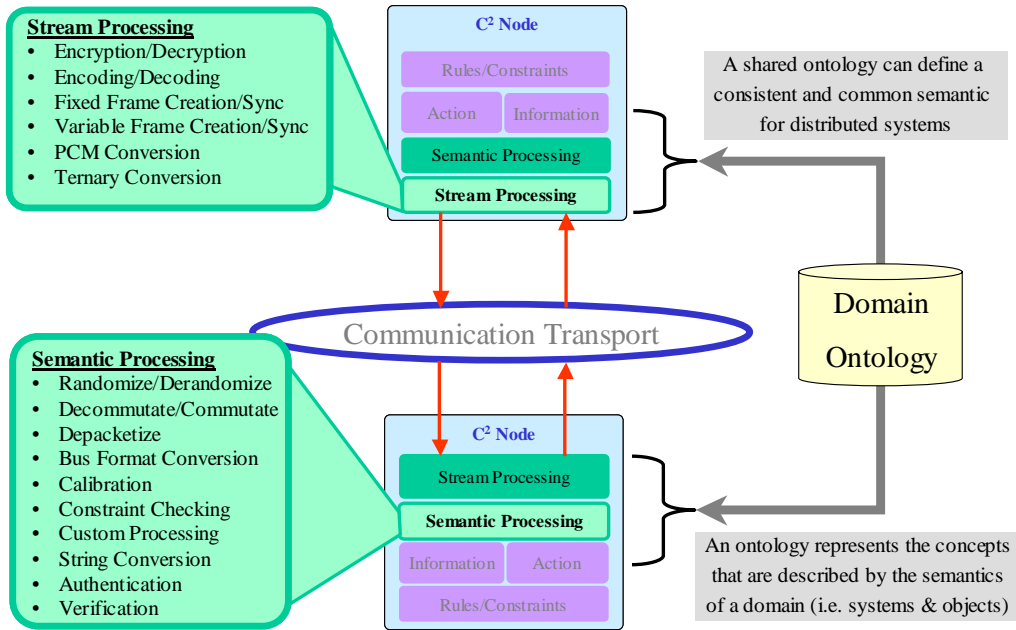


Figure 2 Common Domain Ontology for Stream & Semantic Processing

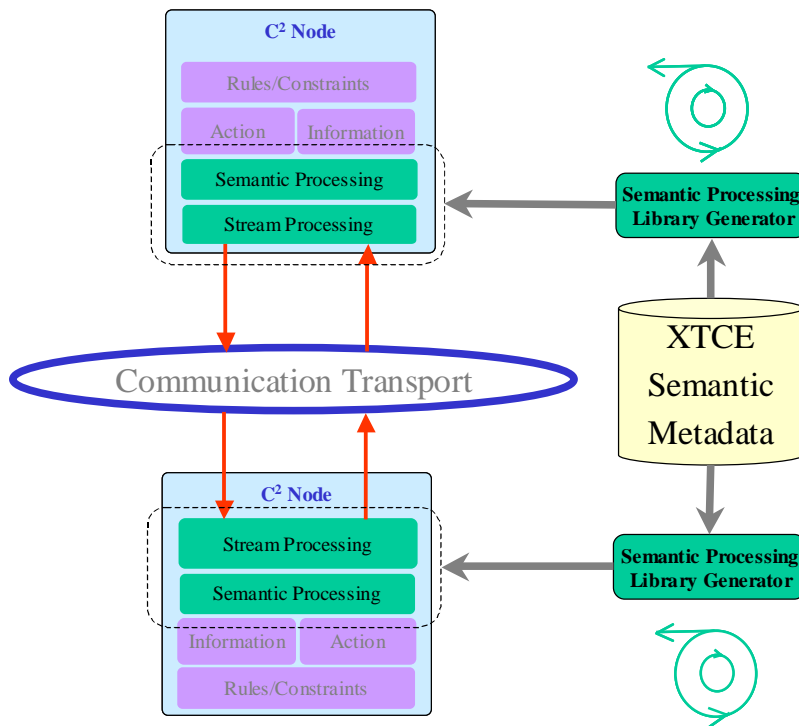


Figure 3 Stream & Semantic Processing Auto-generation Concept

Information processing service products/components will be created from a base of common assets created by a domain engineering process. Domain engineering and application engineering will serve as complementary, interacting, parallel processes that comprise a model-based, reuse-oriented software production system that produces stream, protocol, and semantic services for ground, flight, and embedded exploration architectures. Building a new instance will be more a matter of assembly or generation than creation, making a common cross program C2 information processing strategy practical and affordable.

The run-time library interface can be event driven, when a parameter in a message is found, the parent callback function is invoked for that parameter. It can also be structural. The parent can point the library at a message and ask for a given parameter.

**Static version** - the code is tailored to a message definition, access parameters by name, statically, in code. This is very fast for messages, which is perfect for embedded devices or vehicles.

**Dynamic version** - the library is built with knowledge that you can query at run-time though a consistent API. This is for code that must determine the messages structure, dynamically, during run-time, such as a highly parameterized ground processing system.

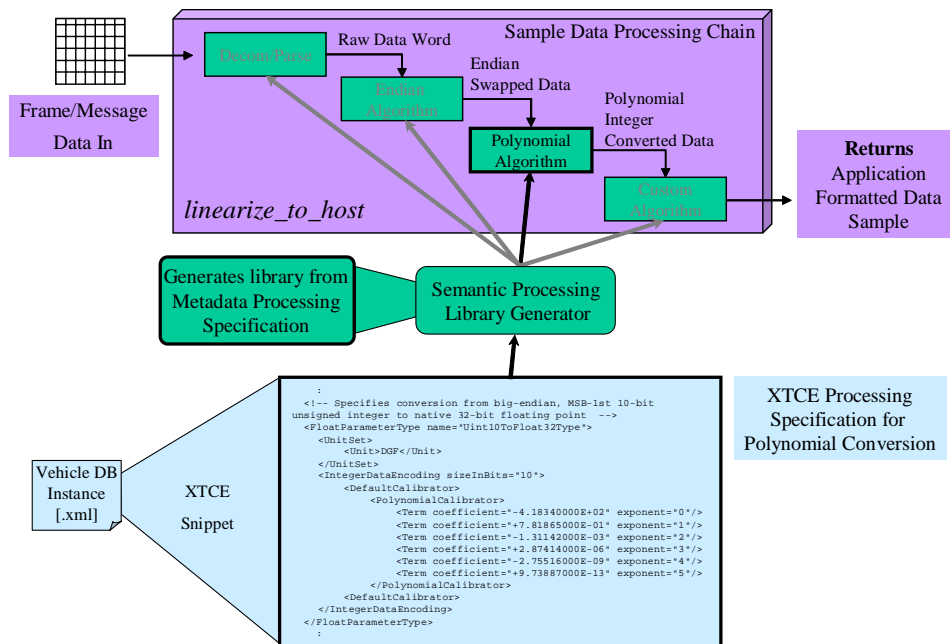


Figure 4 Example Auto-generation Scenario

A simple example use of the Service Library is illustrated in Figure 4. A buffer comes in from hardware as a telemetry frame, the ground system application points the library for reading the stream at the start of the message using the standard call “*parse\_message*”. The *parse\_message* routine invokes a callback functions set up for each parameter. If there is no callback function

for a parameter it finds in the message, the library ignores it. The callback function receives the raw or native data extracted from the message, the library functions “*convert\_to\_host*”, “*linearize\_to\_host*” return the value of the parameter in platforms native data types, or in Engineering Units respectively.

### **XML – An Enabling Technology**

XML is a key enabling technology for information exchange problems. It has existed since 1996, and was established as a standard by the World Wide Web Consortium in 1998. XML is a structural and semantic markup language. The power of XML lies in its simplicity, its support in the commercial community, and its relationship to the Internet. XML has its root in other markup languages (e.g., Hypertext Markup Language (HTML) and Standard Generalized Markup Language (SGML)) that deal with data format. XML allows a user, or community of users, to define a set of markup tags that capture the inherent structure of the data. The components of the structure are called elements, and these elements are constructed in a hierarchical form. Unlike HTML, which uses tags to define data presentation, XML uses tags to describe data content. This provides a mechanism for coupling the meaning of the data with the data itself, and makes that meaning available to software.

XML consists of an *extensible* set of rules for designing text formats to structure data. It is an incredibly simple, well-documented, straightforward data format. It allows users to define a new document format by combining and reusing other formats. XML elements can be embedded and layered in complicated patterns. Since XML is portable and license-free, there is no cost to use it, but you still have to build your own database and your own programs and procedures that manipulate it.

### **The Emerging XTCE Standard**

The Object Management Group and a number of major US and international industry and government aerospace organizations have collaborated to produce the XML Telemetry and Command Data (XTCE) Specification. The XTCE specification is intended as a way to describe telemetry and command “databases” as used in space and ground telemetry systems, packet, and TDM based systems. The XTCE is only a specification, not a service. The intent is to allow the easy interchange of these databases between systems and organizations.

OMG’s vision for the XTCE is that it will one day be the “native” format for ground systems. Until that time, companies and organizations can use converters to go from one system to another, or can convert an existing database into this format for exchange with other parties.

Currently in final draft release, the XTCE standard is projected for approval in 2004. The scope of the specification includes:

- Telemetry data definition including support for CCSDS packets as well as TDM frames
- Data manipulation algorithms to support packaging and unpacking of individual data items
- Commanding data definition including command identification, argument specification, and validation criteria
- Data representation definitions

- Data properties including such things as its default value, validity criteria, and data dependencies
- The definition of extensible formats such that blocks of information can be portrayed in this architecture

The XTCE uses *XML Schema* to describe TM/TC information. The XTCE schema is organized into seven separate schemas: 1) Space System, 2) Parameter, 3) Common Types, 4) Packaging, 5) Algorithm, 6) Stream Definition, and 7) Command Definition.

The XTCE schema is a hierarchical structure, mimicking the organization of space systems, which are typically systems within systems. The hierarchical approach is useful for minimizing name space collisions, more manageable organization, and implicit inheritance of features from higher levels to lower levels. It consists of a collection of space systems including space assets, ground assets, multi-satellite systems and subsystems.

### **Standardization Issues**

There is a proliferation of data representation standards, often making it difficult to determine exactly what is the ‘standard’ for a given domain. Instead of determining a path that ensures compatibility, organizations must either adopt a set of related standards or try to choose the one that seems to be more ‘preferred’ within their domain. This non-deterministic approach introduces inherent risk when investing in a new system built around a specific standard that may become obsolete or rarely used. Once a standard is adopted, it tends to become tightly integrated into systems and processes. Users and operators become familiar with specific keywords and syntax, and special tools are purchased. Eventually, organizations become financially committed to maintaining legacy compliance and tend to resist migrating their systems to adopt new standards.

Although standardization enables interoperability, the standards themselves continue to evolve. In the long-term, adaptability is the key to survival for large-scale systems that rely on standardization. These systems need an approach that abstracts external standardization details from internal components and operations. Systems and organizations that are insulated from, or loosely coupled to, the intricacies of standards compliance can take a more practical approach in the near-term, and have a cost effective means to adapt and evolve in the future.

### **Relevant CCT Experience**

Control and monitoring of spacecraft systems is CCT’s core competency. In addition, CCT has significant experience applying XML technology to solving typical control and monitoring data definition problems. We rely on XML for virtually all our data definition chores.

CCT’s flagship product, The Command and Control Toolkit™ (CCTK), incorporates an XML database for all stream and semantic definitions, including telemetry interfaces for PCM data acquisition, simulation and data transmission. However, when CCT decided to significantly leverage XML technology a few years ago, it had nothing to do with standards, and everything to do with using a great technology for describing and exchanging information that provided our product line with ease of reconfiguration, decoupled our core software from proprietary database management systems, and enhanced extensibility and flexibility to new applications. In fact, at

the time we transitioned to using XML, there were no standards like XTCE even on the drawing board.

### **Relevant CCT R/R&D**

CCT's primary interest in this research is the potential for incorporating the resulting technology into our command and control product line. CCT engineers have participated in many projects that required the use of data and command description languages similar to what has been discussed earlier in this proposal. In addition, as discussed in section 3, CCT's core product line heavily leverages technologies that are directly applicable to this proposed research. CCT has invested heavily in R&D in the last 3.5 years in information exchange technologies that directly support our products and our customers.

The Command and Control Toolkit™ (CCTK) is a commercial CCT product used to develop turnkey vehicle control systems. The CCTK provides a framework for command and control systems including real time messaging, archival and retrieval functions, and multiple plug and play data interface support. CCTK is easily customized with COTS tools and custom software to readily fit varying applications. The CCTK is based on a commercialized NASA technology developed for Space Station ground operations. This SBIR project is designed to augment ongoing design and development work in this area to the mutual benefit of NASA and CCT.

## **CONCLUSIONS**

The current lack of standardization for exchange of telemetry and commanding information is clearly a problem in search of a solution. The emerging OMG XTCE standard is a step in the right direction; however, successful interoperability requires adaptation of a variety of systems. Without supporting cross platform tools that ease the burden of adoption a new standard may not be affordable for most.

The proposed technology will reduce overall life cycle cost for future systems and be supportive of future autonomous software systems that rely heavily on discovery and adaptation. Systems that must operate with constrained performance margins, low power budget, or with high throughput efficiency will benefit most from this technology since auto-generation will produce very compact and efficient services (as opposed to generic, complex, highly parameterized services).

Challenges to successful implementation of this strategy include 1) use of auto-generated software will require new approaches to test and certification of mission-critical systems; 2) space domain metadata specifications are not mature and will require refinement and extension in order to fully address the full scope of NASA's Exploration Program and the Network Centric C2 needs of DOD; 3) cross-cutting technical solutions of this sort are culturally unfamiliar to most large organizations, significant emphasis will have to be placed on inclusion especially during domain analysis and scoping.

CCT's experience with XML in telemetry and command applications over the last four years, and during the 1<sup>st</sup> phase of this research project, has proven that there are tremendous benefits in using XML as a semantic description language both for exchange and native configuration description. Our experience has also taught us that adopting XML is not necessarily a simple undertaking. In fact, it can be tedious and software intensive requiring a significant engineering effort; however, we see this as an opportunity rather than a problem. We plan to leverage our in-depth XML skills, the new XTCE standard, and R&D support from organizations with a need to improve the state of semantic knowledge exchange, to create a generic cross platform suite of tools that significantly ease the adoption of XML and XTCE as a telemetry and command description language.

We recently completed the proof of proof-of-concept phase of this research and are seeking collaborators and contributors to this effort as we transition to full scale implementation. Success will require a broad cross section of input from the ground systems domain. We have received domain analysis support from the NRO, Air Force, and a number of NASA organizations. Other participants are welcome and desirable.

## REFERENCES

- W3C Recommendation - Extensible Markup Language (XML) 1.0 (Second Edition)", [www.w3.org/TR/REC-xml](http://www.w3.org/TR/REC-xml), Oct. 2000
- Object Management Group, XML Telemetry, and Telecommand Data Specification – Draft Adopted, <http://www.omg.org/docs/dtc/03-05-07.pdf>, May 2003
- Range Commanders Council (RCC), IRIG Standard 106-01, <http://jcs.mil/RCC>
- Abstract Syntax Notation One (ASN.1), <http://asn1.elibel.tm.fr/en/index.htm>
- Simon, Gerry; Lockheed Martin, AF-SMC/TEO, "Applying the Extensible Markup Language (XML) for Spacecraft Ground Systems Database Interchange", [www.classic.ccsds.org/meetings/xml2001summer/papers](http://www.classic.ccsds.org/meetings/xml2001summer/papers), 2001
- Simon, Gerry; Lockheed Martin, AF-SMC/TEO, "XML Command and Telemetry Standardization", GSAW 2002, <http://sunset.usc.edu/events/GSAW/gsaw2002/s3/simon.pdf>, 2002
- Peccia, Nestor, ESA, "XML Workshop – ESA Activities on XML", <http://www.classic.ccsds.org/meetings/xml2001summer/papers/CCSDS-WAS-DC-XML-Workshop.ppt>, 2001
- Dubuisson, Olivier, "ASN.1 - Communication between heterogeneous systems", Kaufmann, 2000
- Czarnecki, Krzysztof; Eisenecker, Ulrich W., "Generative Programming: Methods, Tools, and Applications", Addison-Wesley, Reading, MA, 2000
- J. C. Cleaveland, "Program Generators Using XML and Java", Prentice-Hall, XML Book Series, 2001, <http://craigc.com/pg/>