

XML Standards as the Basis for Data Interoperability Among Military C2 Systems and Beyond

Dr. Robert Beardsworth
Mr. David O'Neill
J87 Joint Data & Services Division
U.S. Joint Forces Command
Norfolk, VA. USA
robert.beardsworth@jfc.com.mil
david.oneill@jfc.com.mil

Dr. Philip Barry
Dr. Scott Renner
Information and Computing Technology Division
The MITRE Corporation
McLean, VA. USA
pbarry@mitre.org
sar@mitre.org

Mr. John Wandelt
Information Technology & Telecommunications Laboratory
Georgia Tech Research Institute
Atlanta, GA. USA
john.wandelt@gtri.gatech.edu

Abstract— Today's battlefield requires agility in a matter of minutes, not hours or days, but there is a critical data sharing gap at the tactical edge. Implementation of the DOD Net-Centric Data Strategy through XML-based data standards is essential to fixing this problem and providing an environment where interoperable data is available to the war-fighter wherever and whenever needed. To address these challenges, we are developing a collection of data components that provide semantics understood by all, and rules for composing them as needed into data exchange specifications. This effort is known as the C2 Core. Possibly the first major user of the C2 Core is the Tactical Edge Data Solutions (TEDS) Joint Capability Technology Demonstration (JCTD). The TEDS JCTD was initiated to work with the Services to overcome documented gaps in data sharing, and create a Joint approach to delivering data for use in a leader centric, net enabled future.

Keywords - Interoperability, data, XML, standards, tactical edge, UCore, C2 Core.

I. BACKGROUND

Net-centric warfare depends on seamless information exchange among the participants. This in turn depends on data interoperability, the ability to correctly interpret data exchanged across system or organizational boundaries. Data interoperability problems are caused by different definitions, of name, structure, and representation. Attempts to resolve differences pair-wise result in N-squared reconciliation efforts that will not scale.

Department of Defense (DoD) data standardization in the 1990s attempted to eliminate these differences. Efforts took the form of developing a single standard data model for the entire enterprise. While there were occasional pockets of success, this was widely judged an overall failure. The primary reason for failure is that a single data model for a

large community simply is not feasible in that it requires that too much knowledge be learned by too many people, too quickly. The challenge is not in the creating the model; the hard part is the community learning. The exact size limit of a feasible single model is not known; but 20 year experience with JC3IEDM and predecessors suggests that all of modeling the entire C2 domain is clearly too large. To address this challenge, the DOD has moved towards communities of interest (COIs). The COI concept in the DoD Net-Centric Data Strategy (NCDS) recognizes this infeasibility, and postulates that if you can't have one huge model for one huge community, then you should have several smaller models for individual COIs.

Since the introduction of NCDS there have been three realizations. First, COIs will always overlap and will develop different definitions for elements within the subject-area overlaps. This results in a return of the N-squared problem, albeit with a much smaller N. Second, experience with Cursor on Target (CoT) shows that a relatively small vocabulary for a large community can work very well. Third, DoD appears to have taken a cue from the National Information Exchange Model (NIEM) and embraced the concept of universal and common core vocabularies. Universal vocabularies capture data definitions common to many COIs, leaving each COI to define just the data that it alone needs. Core vocabularies may not be sufficient for any community, but contain definitions necessary to many or all. The DoD Universal Core (UCore) and C2 Core are the first attempts to realize the universal/common core vocabulary concept.

II. A TIERED STANDARDS APPROACH - UCore, C2 CORE AND COI STANDARDS

The concept of universal and common core data vocabularies was introduced to address the COI vocabulary overlap problem. Core vocabularies capture definitions that are common to many COIs, leaving each to define just the data that it alone needs. For example, the UCore provides a very small set of common definitions – Where, When and Who with some guidelines for What – needed in almost all communities. By design, UCore is applicable to a large number of domains besides C2. Consequently and by design UCore does not go deep enough for the C2 domain; that is, there are definitions needed by multiple C2 COIs that are only of interest to the C2 domain. These cross-C2 definitions do not belong in UCore, but they do not belong within any particular C2 COI, either.

The C2 Core is an emerging open standard built for the C2 community and by the community. C2 Core will consist of reusable XML metadata components that can be used as an extension of the UCore schema and taxonomy in a UCore Universal Lexical Exchange (ULEX) message framework. C2 Core is developing a core collection of XML metadata components needed by most if not all C2 COIs, providing semantics understood by all, and that are composed as needed into specifications for specific data exchanges. It is the intent that these C2 Core reusable metadata components be extended by COIs, Programs of Record (POR) and others, with controlled vocabularies, elements and attributes unique to specialized fields and information exchanges.

One of the key concepts for developing the C2 Core data exchange specifications involves a layered approach to data interoperability. This layered approach allows for multiple levels of understanding, which supports multiple levels of abstraction and precision. Parties to a data exchange may not be able to understand all aspects of the data they receive. At a minimum, layered interoperability provides machine-to-machine understanding to the level programmed by software developers as illustrated in Fig. 1 below.

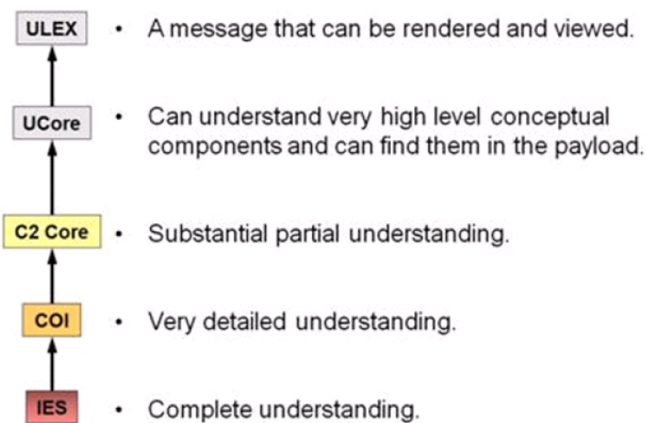


Figure 1. Tiered Interoperability and Federated Data Models

As a result, layered interoperability establishes a base level of understanding for exchange partners adopting C2 Core data exchange specifications. A single, authoritative, and federated

approach, parsing responsibility for data definitions to pre-determined realms of responsibility, provides a governance and work-breakdown structure that reduces ambiguity of design and responsibility. Further, layering reduces and tends to isolate the impact of change. It also helps reduce the time and costs necessary to develop and deploy new data exchanges.

C2 Core must support a spectrum of exchange requirements with a flexible information exchange framework, which is enabled by the afore-mentioned tiered approach to interoperability. C2 practitioners identify their data sharing problems and define their vocabulary needs. They first look for the necessary components in UCore, then C2 Core, and finally in C2-related COIs. New artifacts are created only when reuse is not practicable. This "reuse before reinvent" approach helps enable an optimal partitioning of the semantic landscape within C2.

Given the hierarchy of reuse by developers scheme, it is believed that COIs and PORs will naturally focus on mission specific artifacts, of which they will be the custodians. In addition to improvements in data sharing and interoperability, the development and implementation of the C2 Core is expected to provide cost savings based on reuse of XML metadata components in future product development, providing a positive contribution to the Return on Investment (ROI) obtained from adopting an enterprise data standard and ultimately from the Service Oriented Architecture (SOA) approach.

III. IMPLEMENTATION PROFILES AND PATTERNS

A. Operating Environment

When considering the best use of an interchange such as C2 Core, it is important to balance the mission context, and particularly the physical operating environment. The operating environment can itself be characterized by multiple factors, e.g. network, system capabilities, physical environment, operational factors, and security. However, because these factors are usually correlated, it is not necessary to consider them individually. Instead, we rely on the *Tactical Edge Framework* [1] to identify common patterns appearing in these factors. We focus on network and system characteristics; in particular:

- Network – connectivity, latency, bandwidth, and packet loss
- System – processing power and storage

As illustrated in Fig. 2, systems exist in operating environments ranging from those having few constraints (i.e., computationally well-equipped, with high levels of connectivity to networks that have low latency, low packet loss, and high bandwidth), to those that are severely constrained (i.e., limited computing capabilities, and mostly-disconnected networks suffering from high latency, high packet loss, and low bandwidth).

		Tactical Fixed Center	Tactical Mobile Center	Mobile Platform	Dismounted User
NETWORK	Connectivity	Well Connected	Well Connected	Intermittently Connected	Mostly Disconnected
	Latency	Low	Medium / Low	Medium / Low	Virtually Unlimited
	Bandwidth	High	Medium / Low	Medium / Low	Virtually None
	Packet Loss	Low	Medium / Low	Medium / Low	Medium / Low
SYSTEM	Processing	Servers or Workstations	Servers or Workstations	Single Workstation or Handheld	Single Workstation or Handheld
	Storage	Large Data Storage Devices	Large Data Storage Devices	Single Hard Drive	Single Hard Drive

"DIL" Threshold

Figure 2. Operating Environments with Network and System Characteristics

B. Usage Patterns

In addition to categorizing the environment, we can define a canonical set of usage patterns. These patterns do not define all possible combinations of the values in our trade space dimensions. However, we believe that the following four usage patterns account for most C2 data exchanges. These usage patterns are summarized in Fig. 3 below.

	Usage Pattern #1	Usage Pattern #2	Usage Pattern #3	Usage Pattern #4
Constraints of the operating environment	Minimal/No constraints	Few constraints	Moderate constraints	Severe constraints
Data exchange patterns	Request/Response Publish/Subscribe Search/Retrieve	Request/Response Publish/Subscribe Search/Retrieve	Request/Response Publish/Subscribe Push to known consumer	Request/Response Push to known consumer
Expected value of unanticipated user	High	High	Moderate	Low

Figure 3. Usage Patterns Based on Commonly Occurring C2 Data Exchanges

C. Implementation Patterns

Given that there is a wide variety of operational contexts and there is also a variety of usage patterns, it makes sense that there would be differing implementations to best employ C2 Core and UCore to meet the information exchange requirements. The *C2 Core – UCore Technical Implementation Patterns* [2] paper describes three implementation patterns. Each implementation pattern includes key features and issues, identified by the *C2 Core – UCore Technical Implementation Patterns* paper. These are:

- ULEX + UCore Digest + C2 Core Payload (Wrapped Payload Method)**
 This pattern is the original design for C2 Core messages. It consists of a ULEX/UCore message framework (providing a standard format for carrying metadata about a message), a UCore digest (providing a high-level view of the data), and a single C2 Core payload (including C2 Core as well as COI/POR extensions and IES extensions).
- C2 Core Structured Payload-only (Separate Payload Method)**

This pattern is composed of a single XML data block, defined entirely by a single IES, which is the C2 Core structured payload. It does not use the ULEX framework, thus there is no digest or attachments.

- Non-C2 Core Implementation**

It is perhaps stretching the term to call this an "implementation pattern". However, there are situations where C2 Core cannot be effectively applied. In particular, if the producers and/or consumers must continue to support an existing data exchange that cannot be changed, then none of the implementation patterns will work, because C2 Core always presumes implementing a *new* data exchange. Likewise, none of the implementation patterns will work in an environment where the message size must be as small as possible *and* computational resources will not support Efficient XML Interchange (EXI) compression.

D. Guidance

An analysis of operational contexts as well as the usage patterns provides insight into a set of recommended implementation patterns given the usage and operational contexts. These are summarized in Table 1 and Fig. 4.

TABLE 1

If you need to support unanticipated users who understand the UCore framework and digest	Use the Wrapped Payload pattern
If some exchanges use publish/subscribe message routing based on the UCore digest content	Use the Wrapped Payload pattern
If all exchanges are push to known consumer	Use the Separate Payload pattern
If you need a message that is as small as possible	Use the Separate Payload pattern
If the expected value of unanticipated use is very low	Use the Separate Payload pattern
If you must support producers or consumers that cannot implement a new data exchange	C2 Core is not applicable

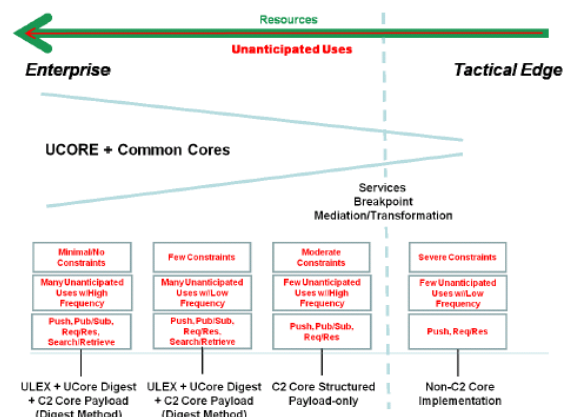


Figure 4. Summary of Trade Space with Usage and Implementation Patterns

IV. COMPRESSION, OR WHY DISCONNECTED, INTERMITTENT AND LIMITED BANDWIDTH (DIL) OPERATIONS ARE NO LONGER AN OBSTACLE TO THE USE OF XML

Size and compression characteristics are important in determining the ability of a pattern to support exchanges in bandwidth-constrained environments. Several methods are available for reducing size of XML documents and recently, a study has been completed that evaluates the effectiveness of compression for each of the implementation patterns described above as well as several additional permutations.

Fig. 5 outlines the process used to obtain the various versions of a compressed message. The message was first stripped of leading and trailing whitespace. Then GZip compression was performed on the message. Finally, the original message was compressed using EXI compression¹. Schema aware mode was chosen. Schema-less mode is also available, but was not chosen because schema-aware mode produces significantly better results.

At the time of testing, only two implementations of W3C EXI were found available for use: one open source and one commercial product. The open source product was known to be a pre-release version still under development. During testing, the open source product was proven to drop data during the compression cycle, producing non-conformant results. All metrics reported in the study were derived from conformant test results.

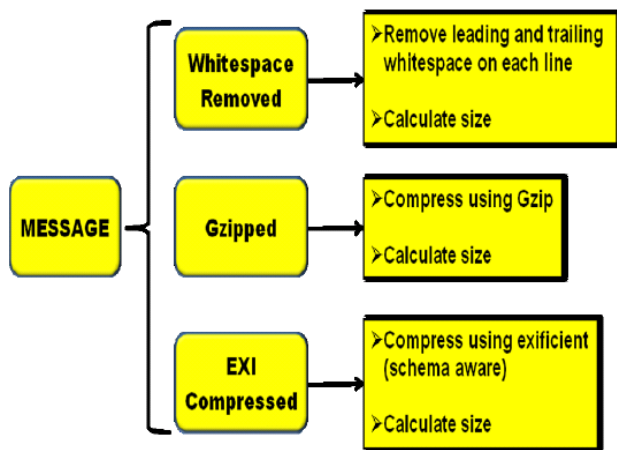


Figure 5. Compression Methods

When represented in EXI format, the structure of the exchange payload is converted into events [StartElement, EndElement, etc. similar to Simple API for XML (SAX) events]. Data is placed between these events. Further optimizations can be performed to increase compression performance. The result can no longer be viewed as text.

Fig. 6 below illustrates that while simple methods such as trimming whitespace and general purpose methods such as GZip provide some improvement, dramatic results can be achieved by employing methods specifically designed to compress XML such as EXI. In addition to the recommended

types described above, the compression study examined a number of other implementation patterns as indicated in the figures.

EXI is able to leverage the structure and language characteristics of XML in order to optimize the amount of bits necessary to represent the data. It is important to point out that these results represent schema-aware compression, i.e. schema is used to predict what data can appear in the instance documents.

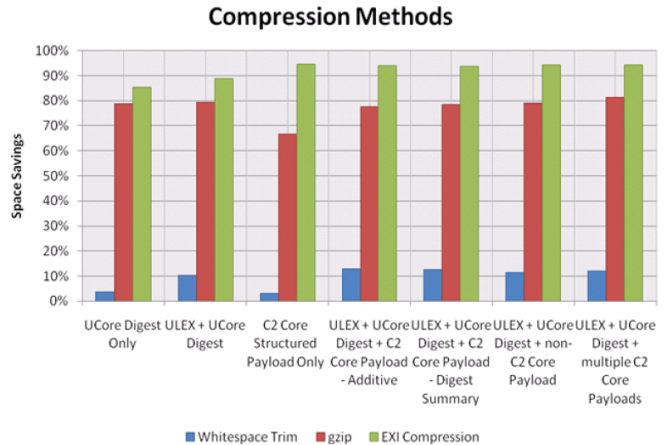


Figure 6. Compression Methods Performance

We have found that we can significantly enhance the compressibility of XML using schema documents to take advantage of schema-aware compression. More precise, better defined schemas give the encoder a better picture of what kind of data can be expected to appear in an XML document, and thus enable better space savings. Fig. 7 illustrates how well each pattern lent itself to schema-aware compression.

Among best practices that are recommended in schemas are:

- use xsd:sequence instead of xsd:all
- use minOccurs and maxOccurs
- use specific schema datatypes (don't make everything a string)
- use decimal (where possible) rather than float or double
- use enumerations if possible

Constructs that limit EXI effectiveness include:

- uses of xsd:any
- uses of xsd:union

¹ <http://www.w3.org/TR/exi/>

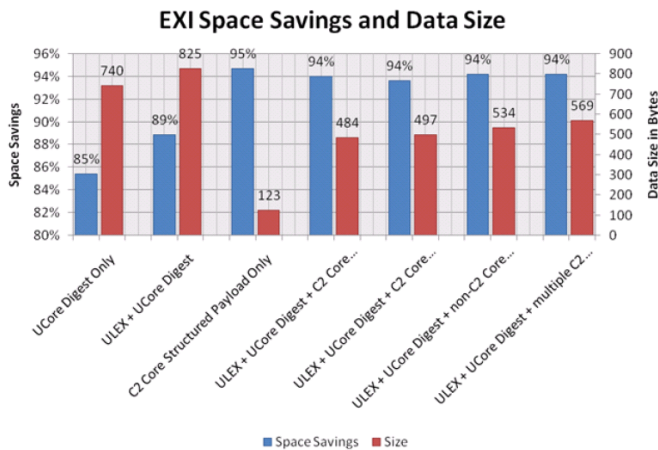


Figure 7. EXI Space Savings and Resulting Data Size

Several of the patterns have achieved space savings in the mid-90%. This indicates that their compressed version is only about 5% of the size of the original. While all of the patterns with a structured payload present in them achieve comparable space savings, the Structured Payload Only pattern clearly reduces to the smallest size.

Fig. 8 below illustrates the original message sizes for each pattern as well as the EXI compressed message sizes. EXI can offer further benefits such as performance improvements (due to the fact that it can be processed in encoded state). Performance depends on the implementation, and evaluating performance was out of scope for the purposes of this comparison.

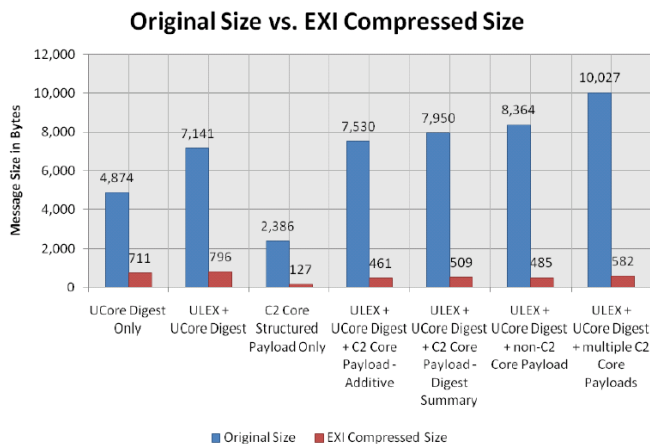


Figure 8. Original Size vs. EXI Compressed Size

From the study it is reasonable to conclude that XML is not the primary limiting factor in determining the final size of a particular data exchange. Other factors such as data content, data structure, environment variables, frequency of exchange, etc. have a greater impact on exchange size.

V. TEDS JCTD: AN EXAMPLE OF CURRENT EFFORTS TO IMPLEMENT C2 CORE

The objective of the TEDS JCTD is to demonstrate the capability of a Service Oriented Architecture solution, using C2 Core conformant data. In this demonstration, data is pulled from the authoritative data source, exposed and shared at the lowest practical level. Since this data is in a standardized format and shared via web services, it makes it extremely flexible and able to be shared to varying applications and unanticipated users.

The TEDS JCTD will demonstrate that C2 Core based data sharing can enable Service, Combat Support Agency (CSA) and Coalition/Allied ground forces and ISR assets to plan, coordinate and execute operations more effectively across disparate warfighting networks in a net-centric environment by sharing authoritative and reusable data. TEDS will demonstrate the process used for developing C2 Core Information Exchange Specifications (IES), meta-data tagging, standardized interfaces, and data sharing web services that can enhance data interoperability and provide useful decision making data to the tactical edge. The TEDS JCTD will conduct technical and operational demonstrations to assess and deliver visible, understandable, accessible, and reusable data solutions.

TEDS is an FY10 to FY12 JCTD that will be conducted in three increments. Increment I will demonstrate ground to ground information exchange between Army Battle Command System Capability Set 11/12 and Marine Corps Infantry Battalion level SOA enabled Combat Operations Center (COC). See Fig. 9.

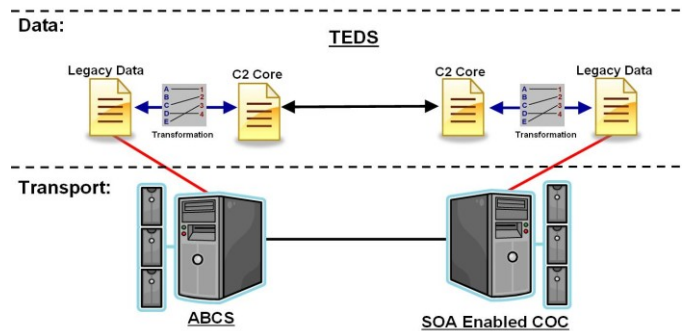


Figure 9. TEDS Architecture

Increment II will leverage the successes from Increment I and demonstrate ground to ground information exchange between a U.S. and Coalition/Allied battalion level elements by applying net-centric standards (e.g. W3C, web services), specifications, practices and procedures developed in Increment I at an appropriate level of Coalition/Allied information exchange.

Increment III will leverage the accomplishments of Increments I and II and demonstrate an information exchange between a U.S. Battalion level element and an ISR asset by:

- Integrating TEDS solutions with the Distributed Common Ground/Surface Systems (DCGS) information sharing in a SOA environment.
- Applying C2 Core in a relevant Intelligence, Surveillance, and Reconnaissance (ISR) mission thread.

Following each Increment and pending the Operational Utility Assessment (OUA), TEDS will determine the requirements for hardware, software and project demonstrations for transition to designated PORs. Process and development achievements will be recorded for each Increment as they evolve in order to document relevant standards and practices, lessons learned and a recommended way ahead for future C2 programs.

The potential of this new way of sharing data are vast, including sharing of data across differing platforms, applications, and networks. TEDS JCTD is taking a small set of Information Exchange Requirements (IERs), pulling the data from its authoritative data source, transforming it to a C2 Core compliant format, and making it available through web services.

The TEDS capability will demonstrate enhanced Joint and Multinational interoperability with improved data sharing by delivering web services, C2 Core compliant formatting, UCore compliant metadata tagging, common standards and interfaces. Ultimately, published data will be presented via web services for access and reuse by other C2 systems outside of the TEDS JCTD. Additionally, this JCTD will lead in the development of standards, practices, specifications and procedures to apply a common data strategy and enable Data Sharing objectives as defined in the *Data Sharing in a Net-Centric Department of Defense (DoDD 8320.02)* [3] dated 23 April 2007. This JCTD will explore the use of Extensible Markup Language (XML) compression methods with the intent of implementing a common data standard solution supporting C2 capabilities as close to the tactical edge as is technically feasible. Finally, the TEDS JCTD will enable C2 vocabulary discovery utilizing the Metadata Registry (MDR).

A. Service, Coalition/Allies, and Agency Operational Problem

Mission critical tactical edge command and control (C2) information systems and applications are unable to automatically share and reuse vital data with comparable Service, CSA, and Coalition/Allied systems at the same security levels. Organizations often must use manual solutions or expensive ancillary mediation products to share critical data. These interoperability solutions are expensive, time intensive, and often provide only point-to-point capabilities with limited or no reusability. Additionally, there are very few established standards, specifications, practices or procedures that unite Service and Agency approaches to overcome these technical challenges. Numerous recent lessons learned from OIF and OEF highlight these challenges. The lessons learned point to the need for net-enabled solutions that allow for sharing of critical information from the tactical edge, between disparate

and geographically separated Service, CSAs and Coalition Allies.

B. Approach

TEDS JCTD objectives include:

- Sharing Battalion level data to support C2.
- Using UCore and C2 Core metadata for interoperability.
- Developing web service interfaces for current information systems exposing data necessary for C2.
- Establishing data exchange standards, practices specifications, and procedures for future reuse C2 systems.

VI. SUMMARY

The C2 Core is being developed to facilitate the machine-to-machine exchange of command and control information. C2 Core is defined by an emerging standard and has the promise of significantly increasing the interoperability among C2 systems. The development of C2 Core has matured to the point where engineering guidance has been developed to ensure the best use of C2 Core to develop information exchanges in differing operational contexts and for several canonical usage patterns. We have further conducted an investigation for compression and have demonstrated that the use of XML is likely not an insurmountable barrier for message exchanges in bandwidth constrained environments. Finally, we have illustrated the use of C2 Core in an operational setting with the TEDS JCTD.

REFERENCES

1. Tactical Edge Characterization Framework – Volume 1: Common Vocabulary for Tactical Environments, http://www.mitre.org/work/tech_papers/tech_papers_08/08_0037/
2. C2 Core–UCore Technical Implementation Patterns, December 22, 2009, GTRI/JFCOM/C2 Core TFTSWG
3. Data Sharing in a Net-Centric Department of Defense (DoDD 8320.02), 23 April 2007