STAFF SUMMARY SHEET

	TO	ACTION	SIGNATURE (*Surname*), GRADE AND DATE		TO	ACTION	SIGNATURE (*Surname*), GRADE AND DATE
1	ESC/CXP Mr. Collens	COORD	5-13-02	5			
2	ESC/DI	COORD		6			
3	ESC/EN	COORD	5/24/02	7			
4	ESC/CX	SIGN	Brig, COL 6 JUN 02	8			

SURNAME OF ACTION OFFICER AND GRADE	SYMBOL	PHONE	TYPIST'S INITIALS	SUSPENSE DATE
TOMASI, GS-13	ESC/CXP	3-1557	cat/ah	

SUBJECT	DATE
Extensible Markup Language (XML) Implementation Guidance	16/19 April 02

SUMMARY

1. PURPOSE: To distribute the attached Extensible Markup Language (XML) Guidance document for implementation guidance to the C2 Enterprise community.

2. DISCUSSION: DAC Enterprise Directive (DED) 002, Web-Enabling Systems for the C2 Enterprise, 16 Jul 01, directed the implementation of four essential web-enabling components: Internet Protocol (IP), Extensible Markup Language (XML), Universal Resource Locator (URL), and Browser-based applications. Responses to the DED indicated that many programs have already implemented IP; URL and browser-based applications to a lesser degree. Implementation of XML lags behind the others, in part, due to its more complex nature. The purpose of this document is to provide implementation guidance for data transfers using XML.

3. This has been coordinated among the principal Chief Engineers at ESC and DISA

4. RECOMMENDATION: ESC/CX sign Letter of transmittal at Tab 1.

Patrick E. Dagle

PATRICK E. DAGLE, GS-14, USAF
Deputy Chief, C2 Planning & Integration

2 Tabs
1. XML Guidance Ltr of Transmittal w/2 attachment
   a. Distribution List
   b. XML Guidance
2. DAC Enterprise Directive 002, 16 Jul 01

DEPARTMENT OF THE AIR FORCE
HEADQUARTERS ELECTRONIC SYSTEMS CENTER (AFMC)
HANSCOM AIR FORCE BASE MASSACHUSETTS

14 June 2002

MEMORANDUM FOR: SEE DISTRIBUTION

FROM: ESC/CX
15 Eglin Street
Hanscom AFB, MA 01731-2100

SUBJECT: Extensible Markup Language (XML) Implementation Guidance

1. DAC Enterprise Directive 002, Web-Enabling Systems for the C2 Enterprise, directed the implementation of four essential Web-enabling components for AF Information Technology and National Security Systems within the C2 Enterprise: Internet Protocol (IP), Extensible Markup Language (XML), Browser-based Applications and Universal Resource Locator (URL). The attached document provides implementation guidance for data transfers using XML. A copy of the document is posted on the C2 Enterprise Portal. The C2 Enterprise Implementation Plan, also posted on the portal, describes the relationship between the DAC Enterprise Directives and the C2 Enterprise Guidance documents.

2. If you have any questions, please contact ESC/CXP, Mr. Tomasi, at DSN 478-1557, commercial 781-377-1557, or email: Carl.Tomasi@hanscom.af.mil.

BRADLEY W. BUTLER Colonel, USAF
Deputy Commander for C2 Enterprise Integration

2 Attachments:
1. Distribution List
2. XML Implementation Guidance

Distribution List

To:
ESC/AC, AW, DI, FA, FD, FM, GA, JS, MC, ND, SR, XP
ESC/BC, BP, EN, HO, IG, IN, PA, PK, SE
MSG/CD, ES
SSG/ED, PM
CPSG/CC

Cc:
AC2ISRC/A-3, A-5
AFSPC/DR
AMC/DO, SC, XP
ASC/AA, RA
SMC/AX, XR
AFRL/XP
OC-ALC/LA
OO-ALC/XP
WR-ALC/XP
AFOTEC/XP
38 EIG/CC
46TW/CC

C2 ENTERPRISE GUIDANCE
16 APRIL 2002

EXTENSIBLE MARKUP LANGUAGE (XML) IMPLEMENTATION GUIDANCE

OPR: ESC/DI (Mr. Jay Scarano)

Certified by: ESC/DI, Mr. Matt Mleziva
ESC/EN, Dr. Louis Metzger
ESC/CX, Col. Bradley Butler
Pages: 19

This C2 Enterprise Guidance document provides guidance on the implementation of Extensible Markup Language for Web-enabling of C2 Systems. In this document, C2 systems includes Command and Control (C2) Systems and Intelligence, Surveillance and Reconnaissance (ISR) Systems as well as those combat control and information systems that support the warfighter or provide safety or mission critical services.

Applicability

This guidance document applies to all ESC System Program Offices and Capstone Program Management Directive (CPMD) systems involved in acquisition of C2ISR systems as well as those C2 Nodes identified by the Designated Acquisition Commander for the C2 Enterprise

Administrative Comments and References

See Attachments 1 and 2 for references, related publications, and supporting information.

Releasability: This guidance document is approved for public release; distribution is unlimited. It is posted on the C2 Enterprise Portal at https://c2info.hanscom.af.mil. Access to the portal requires a user name and password.

Recommended Changes: Refer recommended changes, suggestions for improvement, or conflicts between this and other publications to ESC/DI, Hanscom AFB MA 01731

1. Purpose. DAC Enterprise Directive 002, Web-Enabling Systems for the C2 Enterprise, directed the implementation of four essential Web-enabling components for AF Information Technology and National Security Systems within the C2 Enterprise: Internet Protocol (IP), Extensible Markup Language (XML), Browser-based Applications and Universal Resource Locator (URL). This C2 Enterprise Guidance document provides implementation guidance for data transfers using XML.

2. Terms of Reference. In this section, the term "program" refers to System Program Offices (SPOs) and their contractors and the term "system" refers to those products (systems, subsystems, or components) delivered as a configuration controlled entity produced by programs. An "XML document" is a data object that is "well-formed" according to the rules of the XML specification.

3. Extensible Markup Language (XML) Description. In common usage, XML denotes both a specific standard and a family of standards. The XML standard is a language for defining data formats that are easy for software applications to generate and process. These data formats can also capture hints that make the data easier for people to understand. XML is therefore very useful for supplying data to people and machines across the web. The XML family of standards are aimed at making it easy for distributed applications to discover, obtain, and manipulate data – typically in XML format – across the web.

One key aspect of XML is that it permits separation of presentation from content; that is, separating the details of *how* information is to be arranged and displayed to people from the details of *what* information is to be conveyed. A single XML document may be presented differently to many people with different preferences and needs. The same document may be used for an application-level, machine-to-machine data exchange.

It is not the purpose of this document to explain every important point about XML. Refer to "World Wide Web Consortium (W3C)" [Reference b] for official definitions and background information on these standards. It is not the purpose of this document to set forth the best practices for employing XML. Suggestions of that nature may be found in the DON XML Developers Guide [Reference c].

4. XML Compliance with DAC Enterprise Directive 002. All information exchanged across system boundaries must be available in XML format. That is, programs must define and implement an XML version of each external interface in their system. An XML interface accepts input data, produces output data, or both, encoding this data in XML documents. An XML interface is defined by specifying the syntax of the XML documents it accepts and produces, and by providing documentation that allows programmers and users to understand the meaning of those documents. An XML interface is implemented by a runtime service that actually accepts and produces such documents.

Bottom line: If your system makes data available to external partners, that data must be available in the form of an XML document. If your system accepts data from external partners, it must be able to accept that data in the form of an XML document. This is required even if none of your current known partners want XML data or send XML data. You are free to implement other external data exchange mechanisms so long as an XML interface is also supported.

Programs are encouraged but not required to define and implement XML interfaces for internal use. Programs are likewise encouraged to use XML for the data they present to their own directly supported users.

The principles in this section apply to most data and almost all programs. Exceptions are discussed later in this paper.

5. Defining an XML Interface. Every interface has two sides. Question: who defines an interface, and who uses the definition provided by another? The guiding principle is that the information producer defines the data representation, and information consumers use those definitions. However, the provider of a web service may define the documents that comprise the input parameters of that service, even though the service partner technically produces those documents. Cooperation and collaboration over XML data definitions is highly desirable – especially when there are many producers and consumers of related information.

Specific requirements and guidance for programs that are defining an XML interface are:

a. Programs that define an XML interface must specify the syntax of the XML documents it accepts and produces. Programs must use the W3C XML Schema standard to express these specifications. Refer to "XML Schema Best Practices" [Reference d] for guidance on creating XML schemas. Refer to "COE Data Emporium" [Reference e] for guidance in choosing element, attribute, and type identifiers. See "XML Schema Namespace Management - Aerospace Operations Naming Conventions" [Attachment 1] for guidance in choosing namespace identifiers.

b. Programs that define an XML interface must also provide adequate documentation for the meaning of the documents it produces or accepts. An adequate definition is one that enables a programmer or user to understand the meaning of the data and determine whether it is suitable for his intended use. This documentation is to be expressed as annotations on the XML schema for the interface. Programs must supply a text definition for every element, attribute, and enumeration value defined in the schema. Refer to the XML Schema specification [available at the W3C site, Reference b] for more information on schema annotations.

Consider the following schema fragment as an example:

```
<simpleType name="CON_ACCESS">
 <annotation>
   <documentation>Indicates the access restrictions and/or the
   condition of the approach to the entity.  Restrictions may be
   natural or seasonal, such as ICE blocked ports, or may be man
   made such as fences and guards.
  </documentation>
 </annotation>
...
</simpleType>
```

c. Programs must publish the XML schema for every external XML interface they define. This is accomplished by entering the schema in the DoD XML Registry [Reference f]. This should be done as early as possible in the development process. Refer to the "COE Data Emporium" for guidance on the registration process.

d. Programs must search the DoD XML Registry for existing definitions that can be reused in the interfaces they define. When existing definitions are reused, this fact must be recorded in the DoD XML Registry according to the procedures given in the "COE Data Emporium".

e. Programs should look for government and industry consortia that produce XML definitions. Some of these may be suitable for reuse in the interfaces the program defines.

f. Programs that use an XML interface defined by some other program must record this fact in the DoD XML Registry.

g. Programs must not unilaterally define XML schemas for information that they do not produce. For example, if your program is not an authoritative source of the Air Tasking Order, then your program must not create and register XML definitions for the information in the Air Tasking Order.

h. Programs are strongly encouraged to define XML interfaces in collaboration with their known information exchange partners. Pairwise collaboration is only a little better than unilateral definitions. Refer to the "Community of Interest Approach to Data Interoperability" white paper [Attachment 2] for the principles of determining the best size and scope of a collaboration group.

6. **Implementing an XML interface**. This section contains guidance for programs that are implementing an XML interface:

a. Programs that produce XML documents must guarantee that the XML documents are valid according to the XML schema they have published in the XML Registry. Programs that receive XML documents should validate them against the schemas published by external parties.

b. Programs must implement one version of each XML interface that is accessible through a URL using HTTP. Programs may implement other versions of the interface using other transport mechanisms (e.g. FTP, SMTP) so long as the HTTP version is also supported.

c. Programs should examine the developing XML-related standards for web services: XML Protocol, UDDI, WSDL, etc. These standards will soon become finished recommendations. They are not yet ready for directed compliance, but they are appropriate to consider in a program's development plan.

7. Exceptions

a. Information that is customarily exchanged using a well-known open standard format does not have to be made available in XML. For example, programs may transfer image data in JPEG format, email messages may continue to use RFC822 headers, etc. There is no requirement to develop an equivalent XML interface for these. A list of the exception formats will be made available over time.

b. Information that can only be expressed using closed proprietary formats does not have to be made available in XML. In these cases it is impossible for programs to comply with sections 3 and 4. For example, programs may continue to exchange word processor files in Microsoft Word (DOC format). There is no requirement to develop an equivalent XML interface for this information. Note that the use of closed formats is contrary to the spirit of "web-enablement" and the DED-002 directive, and should be avoided in all but the small minority of cases where the benefits outweigh the interoperability costs.

c. Information intended for presentation that is currently held in Standard Generalized Markup Language (SGML) format does not have to be immediately converted into XML. However, programs should consider migrating such information from SGML to XML over time.

8. References

a. DAC Enterprise Directive (DED) 002, Web-Enabling Systems for the C2 Enterprise, 16 Jul 01, https://c2info.hanscom.af.mil. (C2 Enterprise Portal).

b. World Wide Web Consortium (W3C), www.w3c.org.

c. DON XML Developers Guide, https://quickplace.hq.navy.mil/navyxml.

d. XML Schema Best Practices, www.xfront.org.

e. COE Data Emporium, diides.ncr.disa.mil/shade.

f. COE XML Registry, diides.ncr.disa.mil/xmlreg/.

9. Abbreviations and Acronyms

C2	Command and Control
COE	Common Operating Environment
CPMD	Capstone Program Management Directive
DAC	Designated Acquisition Commander
DED	DAC Enterprise Directive
DON	Department Of The Navy
ESC	Electronic Systems Center
FTP	File Transfer Protocol
HTTP	Hyper-Text Transfer Protocol
ISR	Intelligence, Surveillance and Reconnaissance
JPEG	Joint Photographic Experts Group
SGML	Standard Generalized Markup Language
SMTP	Simple Mail Transfer Protocol
SPO	System Program Office
UDDI	Universal Description Discovery and Integration
URL	Uniform Resource Locator
W3C	World Wide Web Consortium
WSDL	Web Service Definition Language
XML	Extensible Markup Language

Attachment 1

XML Schema Namespace Management
Aerospace Operations Naming Convention

1. Purpose

This attachment describes the convention for defining namespaces within the Aerospace Operation namespace of the Defense Information Systems Agency (DISA) Common Operating Environment (COE) Extensible Markup Language (XML) Registry.

2. Goal

The goal is to provide an agreed upon naming convention for the Aerospace Operations namespace for registering schemas in this registry.

3. Assumptions

Namespaces are a powerful tool for data standardization and reuse.[1]

XML schemas registered in the Aerospace Operations (AOP) namespace of the DISA COE XML Registry will use this naming convention.

Where possible, a global perspective will be used (e.g., joint, not service specific; mission, not organization).

Conventions should specify the minimal guidance needed. Further subdivisions within a sub-domain should be done in the way most appropriate for that community of interest.

Schemas may be imported or included across namespaces at any level.

4. AOP Namespace Conventions

This section outlines the naming conventions. The motivation behind most of the conventions is consistency with the World Wide Web Consortium (W3C) namespace naming conventions and XML Schema specification.

---

[1] Namespaces can be used to develop XML schemas in well-defined sub-domains. This allows multiple domains to examine the same object from different perspectives without the risk of naming collisions. For example, one can examine a satellite from many perspectives (space surveillance, satellite control, etc.). One could develop an XML schema in each of these domains, with each containing an element called "Satellite." Since each "Satellite" element is in a different namespace, there would be no naming conflict. Thus, using namespaces allows one to avoid ambiguity and focus on characterizing elements in a way most appropriate in one's own domain. In this way, one can make progress in one's own sub-domain without waiting for a standard definition to be agreed upon across all relevant sub-domains. However, if these XML schemas are registered and exposed to others, this generates the opportunity for reuse and creates the potential for data standardization.

4.1 Use URL Syntax

An XML Schema namespace name is just a label (i.e., a unique value). There is no requirement (or expectation) to resolve the namespace to an online resource. The XML Schema specification [5] states that target namespaces enable us to distinguish between definitions and declarations from different vocabularies. The schema specification also requires that target namespace names must be a valid URI. A URI may be a Universal Resource Locator (URL), a Universal Resource Name (URN), or both [1].

Whether to use a URN or URL for an XML Schema namespace name is predominantly a matter of personal preference. However, this proposal uses URL syntax for the following reasons:

- URLs are integral to the World Wide Web (www). With a URL, there is *potentially* a resource as well. That resource could contain documentation (a schema, pointers to other schemas, etc.). If in the future the W3C decides to have namespace names point to resources, the appropriate syntax will already be in use and namespace names will not have to change.
- The URL syntax is familiar and memorable to www users.
- URL schema names are already managed [4]. Therefore, it would be easier to ensure namespace names are unique. In other words, with URLs it would be difficult to have two organizations with identical namespace names.
- This convention adopts a prefix of "xmlns://" to ensure that users realize that the target namespace name is a label, not an online resource.

4.2 Use Slash Character for Separating Hierarchical Components

RFC2396 [1] requires that URIs that are hierarchical in nature use the slash "/" character for separating hierarchical components. In this way, URIs are read from left-to-right with each slash denoting another level in the hierarchy.

4.3 Begin with lower case Letters and Use Only Unreserved Characters

Valid URLs in this namespace must use only unreserved characters as defined in [1]. These unreserved characters are a combination of upper and lower case letters, decimal digits and a limited set of punctuation marks and symbols shown in quotes below. These are the hyphen, underscore, period, exclamation point, tilde, asterisk, apostrophe, left parentheses, and right parentheses

"-" "_" "." "!" "~" "*" "'" "(" ")"

As a convention, use lower case letters for except where a capital letter delineates the start of a new word (e.g., spaceControl). Use of a capital letter to delineate the start of a new word is intended to improve readability.

4.4 Include an Enterprise Namespace in Each Sub-domain

Each sub-domain must have an enterprise namespace to contain common definitions that are applicable across that sub-domain. The enterprise namespace should be defined at the same level as the mission sub-domains across which it applies. For example, the Aerospace Operations enterprise namespace would be 'xmlns://us/mil/aop'.

4.5 Publish Acronyms and Abbreviations

Use acronyms and abbreviations where appropriate, but be sure their expansion is published in the enterprise sub-domain (see convention 4.4).

5. Syntax Summary

Note that this syntax summary applies only to the Aerospace Operations namespace and is not meant to be all-inclusive.

xmlns://*country/govermentBranch/disaNamespaceAbbrev/missionSubdomain*

Token	Meaning
xmlns://	Begin each namespace name with "xmlns://" to denote that the name is an XML namespace name using URI syntax.
country	The second field is an abbreviation for the relevant country (e.g., us) or group of nations (e.g., nato).
govermentBranch	For United States government namespaces, the third field designates the branch of the government (i.e., government or military as gov or mil, respectively).
disaNamespaceAbbrev	For the DISA namespace registry, the fourth field is the DISA XML Registry namespace abbreviation. For example, DISA's Aerospace Operations namespace abbreviaion is "aop". For a list of DISA XML Registry namespaces and their abbreviations, see: http://diides.ncr.disa.mil/xmlreg/user/namespace_list.cfm
missionSubdomain	Subdivisions within the Aerospace Operations namespace should be along mission lines. Therefore, the Aerospace Operations namespace will have a list of n mission sub-domains. An example, related to the four sub-domains which apply to the USCINCSPACE missions are: • Space Control, • Space Support, • Force Enhancement, and • Force Application. Other communities of interest will add to this linear list of mission namespaces as needed.

6. AOP Namespace Naming Examples

Community of Interest	Namespace Designation
National Aeronautics and Space Administration (NASA)	xmlns://us/gov/nasa
COE XML Registry Aerospace Operations namespace Enterprise level definitions	xmlns://us/mil/aop
COE XML Registry Aerospace Operations namespace USCINCSPACE missions	xmlns://us/mil/aop/spaceControl xmlns://us/mil/aop/spaceSupport xmlns://us/mil/aop/forceEnhancement xmlns://us/mil/aop/forceApplication
COE XML Registry Aerospace Operations namespace USCINCSPACE Space Control mission Space Surveillance portion of Space Control mission	xmlns://us/mil/aop/spaceControl/surveillance
COE XML Registry General Military Intelligence namespace Enterprise level definitions	xmlns://us/mil/gmi

7. References:

1. Berners-Lee, T., Fielding, R. and Masinter, L., *Uniform Resource Identifiers (URI): Generic Syntax,* RFC 2396, August 1998. Available online at: http://www.ietf.org/rfc/rfc2396.txt
2. COE XML Registry home page, http://diides.ncr.disa.mil/xmlreg/user/index.cfm
3. Masinter, L., Alvestrand, H., Zigmond, D., and Petke R., *Guidelines for new URL Schemes,* RFC 2718, November 1999. Available online at: http://www.ietf.org/rfc/rfc2718.txt
4. The World Wide Web Consortium (W3C) web page on *Naming and Addressing: URIs, URLs, . .,* available online at http://www.w3.org/Addressing/
5. W3C Recommendation, *XML Schema Part 0: Primer*, 2 May 2001. Available online at: http://www.w3.org/TR/xmlschema-0

6. AOP Namespace Naming Examples

Community of Interest	Namespace Designation
National Aeronautics and Space Administration (NASA)	xmlns://us/gov/nasa
COE XML Registry Aerospace Operations namespace Enterprise level definitions	xmlns://us/mil/aop
COE XML Registry Aerospace Operations namespace USCINCSPACE missions	xmlns://us/mil/aop/spaceControl xmlns://us/mil/aop/spaceSupport xmlns://us/mil/aop/forceEnhancement xmlns://us/mil/aop/forceApplication
COE XML Registry Aerospace Operations namespace USCINCSPACE Space Control mission Space Surveillance portion of Space Control mission	xmlns://us/mil/aop/spaceControl/surveillance
COE XML Registry General Military Intelligence namespace Enterprise level definitions	xmlns://us/mil/gmi

7. References:

1. Berners-Lee, T., Fielding, R. and Masinter, L., *Uniform Resource Identifiers (URI): Generic Syntax,* RFC 2396, August 1998. Available online at: http://www.ietf.org/rfc/rfc2396.txt
2. COE XML Registry home page, http://diides.ncr.disa.mil/xmlreg/user/index.cfm
3. Masinter, L., Alvestrand, H., Zigmond, D., and Petke R., *Guidelines for new URL Schemes,* RFC 2718, November 1999. Available online at: http://www.ietf.org/rfc/rfc2718.txt
4. The World Wide Web Consortium (W3C) web page on *Naming and Addressing: URIs, URLs, . .,* available online at http://www.w3.org/Addressing/
5. W3C Recommendation, *XML Schema Part 0: Primer*, 2 May 2001. Available online at: http://www.w3.org/TR/xmlschema-0

Attachment 2

A "Community of Interest" Approach to Data Interoperability

Scott A. Renner, Ph.D., Member, AFCEA
The MITRE Corporation
201 Burlington Ave., MS B265
Bedford, MA 01730, USA
(781) 271-8625
sar@mitre.org

Abstract

Existing data administration policy in the DoD has produced good results in some pockets, but has not delivered data interoperability for the enterprise as a whole. In this paper we present a concept of operations for a different approach to data interoperability, one based around the notion of "communities of interest". This approach corrects the shortcomings of the current regime in three ways: First, it does not assume a single monolithic data standard. Instead, we work toward data interoperability within separate communities of interest. These groups serve a knowledge management purpose: they extract the shared knowledge necessary for data interoperability from the larger community, then make this knowledge explicit, and finally help to transfer that knowledge back into the whole community. Second, our approach captures knowledge at more than one level of abstraction. Our approach will produce higher-level definitions that can support interoperability in architecture descriptions and information retrieval "semantic tags" in addition to the current implementation-level data standards. Finally, our approach can become part of the system acquisition process, giving both better incentives for program offices to participate, and better metrics for measuring their results.

Key words - Community of interest, data interoperability, knowledge management

1. Data interoperability is a knowledge management problem

Data interoperability is a key element in the *Joint Vision 2020* goal of information superiority [2]. But we still do not have a workable plan for achieving DoD-wide data interoperability. This paper is intended to describe the principles on which such a plan might be implemented.

We define data interoperability as the ability to correctly interpret data that crosses system or organizational boundaries. The key points are illustrated below in Figure 1. We assume that the people on the left have information needed by the people on the right, and that data in one system is accessible to the other. Information can flow (and we have data interoperability) if and only if the receiving system and users properly understand the data they receive. Clearly, data interoperability is only one aspect of the overall interoperability problem. The left-hand side might not ever have the information needed

by the right-hand side. Or it might be impossible to get data from one system to another (because of incompatible radios, physical connectors that don't fit, etc.) While those would be interoperability problems, they would not be data interoperability problems.

Two things must happen before two systems or organizations can understand each other's data. First, the people involved must identify a matching real-world thing of mutual interest. (All data describes some thing of interest. You can't meaningfully exchange data unless it describes the *same* thing.) We call this step *establishing a semantic match.* Second, they must arrange to eliminate or otherwise deal with the differences in the name, structure, and representation in the data models they use to describe this real-world thing. For example, if you measure a certain distance in miles, but I expect distances measured in kilometers, then the appropriate conversion must be applied to the data before I can properly use and understand it. This second step is *coping with representation mismatch.* (More details, with examples, can be found in [6].)

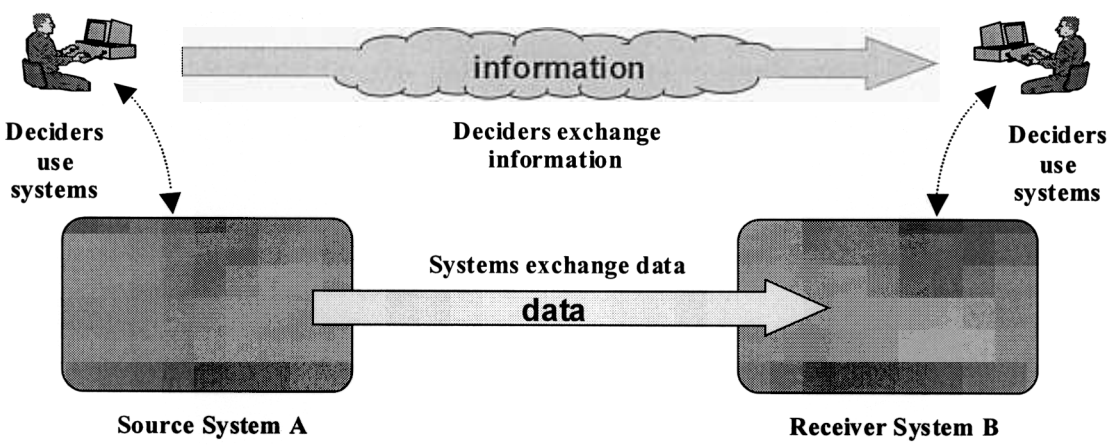

Figure 1: Data Interoperability

In this paper we focus on the semantic match problem. It is by far the most challenging of the two. Representation mismatch problems can have automated solutions, either through hand-written translation software or data mediation tools. But there is never going to be an automated approach to the problem of semantic matching, because the solution takes place inside the human mind. That is, for every data interoperability success, there are at least two people with a shared understanding of the exchanged data. These people may be users who look at the data presented to them, or they may be programmers who decide what systems will do with the data they process. Either way, these are the people who know what the data should mean. If the data they receive does not conform to the definitions they know and expect, the result will be an error, not interoperability.

Crucially, what must be shared is knowledge, which is inside human minds, and not merely written definitions, which can be captured in words and paragraphs. For example, you and I might both be satisfied with the definition "an aircraft is a vehicle that moves through the air." But if in your world all aircraft happen to have a fixed wing and at least

one engine, while my world includes free balloons and helicopters, then this difference in tacit knowledge is likely to cause some interoperability problems even though we are using the exact same explicit definition. Conversely, the developers on my software project might arrive at a common understanding of our data through conversation, without any written definitions at all. Shared knowledge, not shared documentation, is what counts. Common explicit definitions are of course useful – we are hardly arguing in favor of undocumented software – but they are neither necessary nor sufficient.

The semantic part of data interoperability is therefore a *knowledge management* problem: How do we arrange for the right people to have the right (shared) knowledge about what the data means? Automated tools can help solve this problem. They can never make it go away.

It is instructive to look at the places where shared semantic knowledge is important and at the roles of the people who need it. Here are the most important cases:

- Users need to understand the meaning of the data presented to them. Programmers need to understand the data their programs will manipulate. These people must have a shared understanding of the runtime instance data values. (This is the most common case, the one that everyone thinks of first.)
- Architects building products from the C4ISR Architecture Framework [1] need a common vocabulary for representing information exchanges. These architecture products include abstract descriptions of the instance data that will be exchanged between executing systems. That is, they need to describe *what* information will be exchanged, but not necessarily how it will be represented. Without a shared understanding of the information described in operational and system architectures, it will be impossible to relate architectures developed by separate teams, or to "roll them up" into an integrated view – or even to be confident that one product will convey the same meaning to any two readers.
- Information dissemination relies on commonly-understood metadata "tags" to distribute information products from producer to consumers. For example, one consumer might state in his profile that he wants information about "surface naval ships". He then receives documents that are so marked by their producers. This depends on a shared classification taxonomy – it won't work unless producers and consumers have the same understanding of "surface", "naval", and "ship".
- The *semantic web* [4] is a plan to extend the existing Web by adding machine-processable semantic metadata to web pages. Software agents can use this metadata to "understand" the information and/or services provided by a web resource and then to act on our behalf: reserve us a rental car, change a doctor's appointment, etc. But unless the people involved (the end user, the agent programmer, the resource provider) have a shared understanding of this semantic metadata, we shouldn't expect these smart agents to actually do what the user wants done.

These people do not all need precisely the *same* knowledge. For example, software programmers (of a web agent or of a traditional application) need more detailed knowledge than software architects. But people in these different roles do not need

completely *different* knowledge, either. Clearly the sort of high-level "what is an aircraft?" definition is useful in every role. The knowledge needed by architects (what information do these systems exchange?) should be an abstraction of the knowledge needed by the system developers (exactly what data are we passing to each other?). Figure 2 shows different roles and the level of detail and assurance they require. The determining factor is the location of the person who does the understanding: end users, who can examine instance data at runtime, need less than programmers, who cannot.

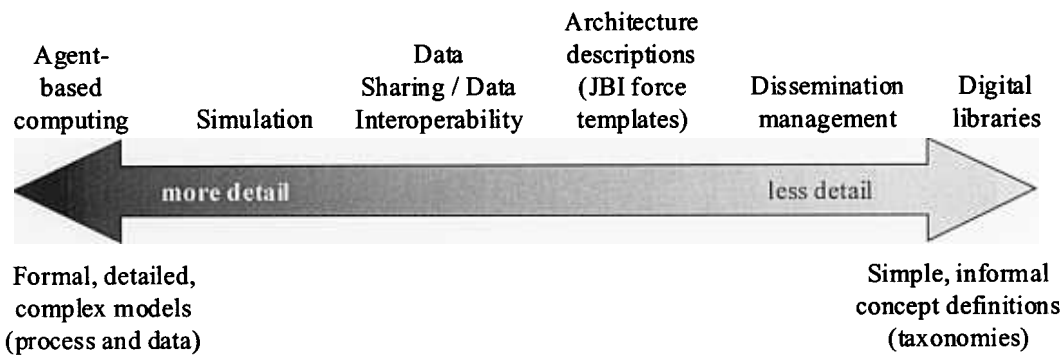

Figure 2: Level of detail and assurance needed for different purposes

How can we explicitly capture this shared knowledge in a way that maintains these relations without requiring people to learn details they do not need? In [7] we suggest organizing semantic metadata into the following layers:

- *Ontologies* are organized collections of definitions, including relations between the defined terms. In its simplest form an ontology is a dictionary of terms. Data models may specify the meaning of their entities and attributes by reference to the definitions in the ontology. (An ontology is not itself really a model; instead, it defines the universe of things that can be included in a data model.)
- *Abstract schemas* define information without specifying a particular data representation. They allow application developers to establish a semantic match with a database (or second application). Abstract models are sometimes called *conceptual data models*
- In a *concrete schema,* all of the attributes are fully-defined as implementable data types. A concrete schema defines everything the application developer needs to know about the data when interfacing with a database (or second application): a complete description of the names, structures, and representations of the data. A fully-attributed, normalized *logical data model* is a concrete model. Standard message formats (e.g. the Air Tasking Order) are also concrete data models.

Organizing semantic metadata in this fashion preserves the connections between related knowledge, because metadata in one layer points to metadata in a higher layer to specify its meaning. It helps with the level-of-detail problem, because people need not learn facts in the lower levels if they do not need them. We still have a big problem with the *scope* of knowledge required. It is impossible for *anyone* to know all of the definitions that are required for *everyone* to exchange data. That is, the shared knowledge required for the

union of all data exchanges is too large to fit inside any one person's head. We must partition semantic knowledge into smaller, more tractable packages. The mechanism for doing this is the subject of the next section.

2. Communities of interest are a knowledge management solution

Data interoperability poses a knowledge management problem: how do you get the needed semantic knowledge into the people that must understand the exchanged data? The community of interest concept is the knowledge management solution we propose. We define a *community of interest* (COI) as the collection of people that are concerned with the exchange of information in some subject area. The community is made up of the users/operators that actually participate in the information exchange; the system builders that develop computer systems for these users; and the functional proponents that define requirements and acquire systems on behalf of the users. The subject area is the *COI domain* – whatever the people in the COI need to communicate about. And in order for the systems and organizations to communicate – to have data interoperability – the people in the COI must all know and understand the consensus definitions for the data they will exchange.

Because we assert *a priori* that the people in a COI need to communicate with each other, we know that it must be possible for them to agree on consensus definitions for the things of mutual interest. But it takes hard work to establish what these things are, and what the agreed definitions will be. This work is the function of the *COI data panel*. The people on the data panel are drawn from the COI. Their job is to represent every part of the COI, determine where the data interests of these parts intersect, and then produce the shared definitions for the union of these intersections. In their job the data panel is performing a knowledge management task. First, the data panel extracts the shared knowledge necessary for data interoperability from the larger community, it then makes this knowledge explicit, and finally it works to transfer that knowledge back into the whole community. The data panel's job is done when everyone in the COI knows the definitions they must have in common in order to exchange information.

2.1 What do COI data panels do?

The main task of the data panel is to produce and maintain the *common data representation* (CDR) for the COI domain. The CDR is the explicit representation of the COI's shared semantic knowledge. It will be organized into ontology, abstract schema, and concrete schema as described above. We expect the data panel to produce reference data sets; that is, the values of coded fields and their meanings. They may also produce business rules (constraints on the data values) and process models (which describe how data is used).

The process of building the CDR is not vastly different from the collaborative data modeling sessions used in the present data administration regime. There are three important differences: The panel will produce more than standard data elements and models and other concrete schema artifacts; it will also build ontologies and abstract schemas. The panel will focus on exchanged data, not all data, and will consider

cost/benefit when choosing the exchanged data that will be modeled first. Finally, existing systems are not required to change their internal data representation to conform to the CDR. Instead the individual program managers will make that decision on a cost/benefit basis. [5] contains a description of a process that could be used to build CDRs, plus more discussion of the three differences above and the advantages they bring. Also, the Health Level Seven (HL7) consortium has developed a detailed modeling methodology that could be adapted to building CDR for COIs [3].

2.2 How are COI data panels formed?

The process always begins when someone identifies the need for a new data panel. This can happen bottom-up, when separate organizations realize that together they form a COI and need a data panel to construct their consensus CDR. It can also happen top-down, by direction, without waiting for organizations A, B, and C to realize (or admit) that they need to cooperate on a COI data panel. No matter how the process begins, the next step is always to prepare a draft charter for the COI data panel. This work will be led by the proposing organization. The following issues must be considered in preparing the draft charter:

- The proper scope of the new COI (the people) and of the COI domain (the subject matter). This is more an art than a science. The first principle is that data interoperability problems arise when data crosses system and organization boundaries. Therefore, useful COIs will also cross these boundaries. A COI completely contained within a single organization (or worse, within a single system) would not be worth the bother. Second, COIs become more valuable as they include more people and cover a larger domain. However, the costs of creating/maintaining a COI increases along with its size. Eventually the costs increase faster than the value. We do not have a hard objective rule for computing the optimum size of a COI – consensus among the participants (is this group getting too big to be effective?) is probably the best we can do, at least for now. As a rule of thumb, consider that everyone in the COI ought to understand all of the definitions in the CDR. When that seems too hard, consider partitioning the COI into smaller specialized domains.
- The organizations that should belong to the COI and be an active part of the data panel. (That is, try to distinguish those who want and need to build the CDR from those who will simply use it.) This will be an iterative process; adding an organization to the COI may change the domain, which may in turn change the organizations that should belong.
- The cost and benefit of chartering a data panel for the proposed COI. Both of these are difficult to quantify exactly. Costs include both the work of building the CDR, and of transferring the knowledge in the CDR to the whole COI. Some benefits are avoided costs elsewhere; e.g. the system interfaces become cheaper to build and maintain. Other benefits appear as increased capability.
- Does the proposed COI cover the intersection of two or more major existing COIs? Then it must be a substantial intersection, something worth the overhead of a data panel. Be sure there really is a distinct community of people who must

work in both domains A and B; be sure the necessary work can't be done within the separate A and B panels.

- Is the proposed COI a subset of an existing COI? This only makes sense if the smaller subgroup needs a specialized vocabulary and if it's too expensive to expect everyone in the larger group to know this vocabulary.

COI data panels are established when their draft charter is approved by the chartering authority. We suggest that the service Chief Information Officer (CIO) is the proper authority for those communities that do not cross a service boundary, and that the DoD CIO is the right authority for communities that include more than one service.
The data panel needs to include representatives from every part of the COI. When some systems or organizations are not represented, it is likely that some of their knowledge will not be captured by the data panel, or that some of the knowledge codified by the panel will not be transferred back into those groups. In either case, data interoperability with the missing groups will suffer.

Most of the people on the panel will be subject matter experts (SMEs) in the COI domain. These, after all, are the people who possess the knowledge the data panel is trying to capture. It is likely that most of the SMEs will be functional proponents. However, because some of the knowledge we are trying to capture is implementation-level detail, the data panel will need to include SMEs from the system builders. Finally, each data panel will typically need some people whose expertise is in capturing data definitions, rather than in the COI domain to be defined.

Most of the work in the data panel will be done by the SMEs drawn from the functional and builder communities, and so most of the costs will be carried by those organizations. Program offices will know the COIs they belong to, and will need to include costs of participation in their budget. A small part of the work should be centrally funded, and not directly paid for by the participating organizations. We suggest that responsibility for developing and maintaining the procedures and infrastructure used by the data panels probably belongs to the DoD CIO.

3. Incentives and motivation: why this solution will work

Suppose we charter a COI data panel and allow it to build the common data representation that captures semantic knowledge for the COI domain. This by itself doesn't do any good. We are no closer to data interoperability until this knowledge is acquired by the rest of the community and used by system developers to implement data exchange. How can we make that happen? Experience teaches that it is not enough to simply give orders to the developers. Instead, we need to make COIs part of the work process they are already following. We also need to show how the COI process makes it easier for developers to satisfy the interoperability requirements they already have.

There are several ways that the COI process can be inserted into the existing system acquisition process. The following suggestions call for the definitions in a COI's common data representation to be used in products that are already being created. If a

program needs a definition that does not exist in any CDR, then that program has not been adequately participating in the corresponding COI data panels.

- *MNS and ORD:* At a minimum, we should require each Mission Needs Statement and Operational Requirements Document to specify the list of COIs in which that system will participate. (A system which does not belong to any COI is by definition a stovepipe. Who wants to fund any more of those?) We might also measure the proportion of nouns and verbs in the functional requirements that are defined in the ontologies from the corresponding COIs.
- *C4I Support Plan:* These are tools for managing implementation issues related to C4I infrastructure for a specific system or program. They may describe information exchange requirements; when they do, these should be expressed using abstract schemas defined by the relevant COIs.
- *Operational and system architectures:* All information exchange requirements described in C4ISR Architecture Framework products should be entirely captured in an abstract schema defined by a COI. The logical data model (OV-7) should map to an abstract or concrete schema defined by a COI.
- *Data exchange agreements, message formats, and ICDs:* All of these should map to a concrete schema defined by a COI.
- *Database definitions:* Definitions of the shared portion of a physical database model should be drawn from concrete model defined by the relevant COIs. That is, when you are building a physical database model, you should define the meaning of your data elements using definitions from the COI's definitions whenever these are available.
- *Semantic tags:* The terms used in profiles for information dissemination should be drawn from an ontology defined by a COI.

The COI approach can make the developer's job of building interoperable systems easier. To a certain extent the COI data panels only reorganize work that programs are already performing. Programs already invest much effort in building interfaces with other systems. Part of that work is creating a shared understanding of the data to be exchanged. That work is no harder when done in a data panel instead of in pairwise interface meetings. Data panels offer the possibility of reuse; that is, some of the shared semantic knowledge needed to interface with system A may be helpful when interfacing with system B. Finally, parts of the CDR produced by the data panels can be used to implement data exchanges with other systems; for example, as inputs to a data mediator.

4. Conclusion

The hardest part of data interoperability is the semantic problem: arranging a shared understanding of what the data means among all the people involved in the data exchange. This same problem occurs when systems share data through direct database exchange, when systems pass XML documents to each other, when separate architecture teams try to integrate their work, and when producers and consumers try to use semantic

"tags" for information dissemination. We can approach the semantic problem in all of these cases by dividing people into communities of interest and making explicit the knowledge each community needs to communicate. That will take hard work, but it is work that would have to be done anyway, possibly over and over. By inserting our "community of interest" approach into the normal system acquisition process, we can arrange for that work to be done once. The result will be a data interoperability solution that can work, because it is technically feasible, and will work, because it will actually be executed by the programs.

References:

[1] C4ISR Architecture Framework, version 2.0, December 1997. http://www.c3i.osd.mil/org/cio/i3/AWG_Digital_Library/pdfdocs/fw.pdf

[2] Department of Defense, Joint Chiefs of Staff, *Joint Vision 2020*, June 2000. http://www.dtic.mil/jv2020/jvpub2.htm

[3] G. Beeler et. al., *Message Development Framework,* Health Level Seven, Inc., 1999. http://www.hl7.org/Library/mdf99/mdf99.pdf

[4] T. Berners-Lee, J. Hendler and O. Lassila, "The Semantic Web", *Scientific American,* May 2001. http://www.sciam.com/2001/0501issue/0501berners-lee.html

[5] S. Renner and J. Scarano, "Organization, Process, and Technical Foundation for Data Interoperability", *DoD Database Colloquium*, San Diego, September 1997.

[6] S. Renner, A. Rosenthal, and J. Scarano, "Data Interoperability: Standardization or Mediation", *1st IEEE Metadata Conference,* Silver Spring, MD, 1996. http://www.nml.org/resources/misc/metadata/proceedings/renner/data-interop.html

[7] A. Rosenthal, E. Sciore, S. Renner, "Toward Integrated Metadata for the Department of Defense", *2nd IEEE Metadata Conference,* Silver Spring, MD, 1997. http://www.llnl.gov/liv_comp/metadata

```
**********************************************************************************
**********************************************************************************
**********************************************************************************
**********************************************************************************
**********************************************************************************
**********************************************************************************
```

```
     XX   XXX X   XXXX
      X  X   XX  X    X
      X  X    X   XX
      X  X    X     XX
      X  X   XX  X    X
      X   XXX X   XXXX
      X       X
  X   X  X    X
   XXX    XXXX
```

```
                                                                  XXXXX
                    X              X                            X     X
                                   X                                  X
 X XXX   X XXX     XX    X XXX   XXXXX    XXXX   X XXX   X     X      X
 XX   X  XX   X     X    XX   X    X     X    X  XX   X  X     X   XXX
 X    X  X          X    X    X    X      XX     X       X     X  X
 X    X  X          X    X    X    X        XX   X        X   X  X
 XX   X  X          X    X    X    X  X  X    X  X         X X   X         XX
 X XXX   X         XXX   X    X     XX    XXXX   X          X    XXXXXXX   XX
 X
 X
 X
```

```
   XXXXX   XXXX   XXXXX            XXXX  XXXXXXX                      X
     X    X    X X     X          X         X                        X
     X   X       X               X          X                        X
     X   X       X               X          X     XXXX   X    X   XXX X   XXXX
     X   X        XXXXX          X XXXX     X         X  X    X  X   XX  X
     X   X  XXXX        X         XX    X    X     XXXXX  X    X  X    X  X
     X   X     X        X         X     X    X    X     X X    X  X    X  X
 X   X    X    X X     X   XX    X     X    X    X    XX X   XX  X   XX  X
  XXX      XXXX   XXXXX    XX     XXXXX     X     XXXX X  XXX X   XXX X   XXXX
                                                          X
                                                       X    X
                                                        XXXX
```

User: jgs
Title: JGS.6Taydo
Date: May 22 13:51:47
Printed via EPrint

```
**********************************************************************************
**********************************************************************************
**********************************************************************************
**********************************************************************************
**********************************************************************************
**********************************************************************************
```

DEPARTMENT OF THE AIR FORCE

HANSCOM AFB, MA 01731-2109

OFFICE OF THE DESIGNATED ACQUISITION COMMANDER, C2 ENTERPRISE INTEGRATION

16 JUL 01

MEMORANDUM FOR SEE DISTRIBUTION

FROM: DAC, C2 Enterprise Integration
ESC/CC
9 Eglin Street
Hanscom AFB, MA 01731-2109

SUBJECT: DAC Enterprise Directive 002 – Web-Enabling Systems for the C2 Enterprise

REFERENCE: CSAF & SECAF Policy Memorandum – Information Technology (IT) Web-enabling Technologies and Standards for AF Applications, 9 July 2001

The referenced Air Force Policy Memorandum establishes formal policy direction affecting our C2 Enterprise. The changes this policy intends to put in place are critical to achieving C2 integration. Therefore, I am directing the following essential components be implemented as quickly as possible for AF Information Technology and National Security Systems within the C2 enterprise shown at Attachment 1.

a. Interfaces will be standardized through the Internet Protocol (IP). By using the IP protocol AF C2 systems will have standard addresses through which they can be accessed – just like the telephone system. This provides a common mechanism to move data from one machine to another permitting operations on standard networks anywhere in the world.

b. Data transfers will use the Extensible Markup Language (XML) to facilitate publish and subscribe information brokering via a standard language – just like English is the standard language for pilots. With XML, an IT system can "publish" data and make it available to many systems and users.

c. Browser-based applications with standard internet addressing (Universal Resource Locator) will be used. Browser enabling an information system permits access to the application/data from anywhere on the network through wide variety of devices (e.g. handhelds) not just PCs – keeping up to date with consumer electronics.

The use of Universal Resource Locators and browser-based applications, when combined with XML and IP provide users with the tools they need to efficiently and quickly access and use applications. IP enabling our information systems will permit them to be connected to the Joint/AF shared network (e.g. NIPRNET or SIPRNET). Like the telephone system, once connected, you only need the address (telephone number) of the party to be called. You can call from anyplace on the globe and the connection path and media are transparent. Adopting a URL provides a network address for any information system so it can be accessed from anywhere on the net via a directory. By making applications browser-based users are presented with a simple and consistent way of interacting with them – just like the computers you use at home.
Lastly, XML enabling an application permits you to send both data and a description of that data in a way that an XML enabled recipient can understand without agreeing in advance on a fixed format.

Request each DAC/PEO in the C2 Enterprise provide a schedule by 31 August 2001 depicting when their programs could comply with each of the above components within their current funding profile. The purpose is not to produce implementation plans at this stage, but to compile an integrated look at when our C2 systems (in the main) might be compliant. This should be a first order look at the overall picture. In the future, each DAC/PEO will need to continue to work with their customers for appropriate requirements and funding changes as required. The opportunity the enterprise approach presents is to begin to work together to make these changes to benefit interoperability across our C2 system. We have scheduled a PEO/DAC forum for 5 September 2001 and these schedules will be a topic of discussion.

If you have questions regarding this directive please contact Lt Col Bill Nelson, ESC/CXP at DSN 478-6420 (bill.nelson@hanscom.af.mil).

LESLIE F. KENNE
Lieutenant General, USAF
DAC, C2 Enterprise Integration

Attachments
1. C2 Enterprise Portfolio
2. Schedule Format

Distribution:

AFPEO/AT
AFPEO/C2&CS
ASC/CC, RA
ESC/AC, AW, CX, DI, FD, FM, GA, JS, MC, ND, SR
ESC/BP, EN, FA, FM, IN, PK, TE, XP
ESC/CV, CD, CO, CCC, CCT, CCX, CCZ
ESC/MSG, SSG, 38EIW, CPSG
OC-ALC/CC
OO-ALC/CC

cc:

AAC/CC
AC2ISRC/CC
AF-CIO
AFMC/CC, DR, EN, LG
AFOTEC/CC
AFPEO/SP
AFPEO/FB
AFRL/CC
ESC/BC, HO, IG, PA, SE
SAF/AQ, AQI
SMC/CC
WR-ALC/CC

C2ISR Enterprise Portfolio

AFPEO/C2 & Combat Support
Brig Gen Riemer

GCSS-AF

FIRST	JAMSS
ILS-S	GTN
GCSS-AF	Remaining
IMDS	PEO/IL Prog
Log Info Sys	

Platform & Sensors
AWACS
R/ADS(R/SAOC)
CRC (GTACS)
CBMS
JSTARS
MP-RTIP

Strategic/Nuclear C2
SWPS
ISC2-N/UWSS
CMC/CMU
MCCC
SEWS/JDEC
SBMCS
Global Hawk (after D

ESC/DAC
Lt Gen Kenne

Integration between Domains
Enterprise system arch definition
Enterprise sys config mgt & control
Enterprise Performance certification
Terminated Enterprise Infrastructure Acq
Major cross program execution

	Programs	GCSS-AF
CIL		App Int
Link-16		AOC
IES	UAWR	DCGS
TDC	ISR SM	TBMCS
CHS	JTRS	Info Ops Sys
GATM/MC2	Contr Info Sys	Func Supt Info
FTE		Force Protection
DII	Integration Efforts	

SIAP/FIOP JDEP
Domain Architectures for GCSS-AF & OCCS-AF
Crypto Systems Fin Info Sys
Med Info Systems Bus Info Sys

ASC/DAC
Lt Gen Reynolds

UAV Enhancements
U-2
Rivet Joint
Compass Call
Commando Solo
Big Safari
JSAF

OO-ALC/DAC
MGen Bergren

Photo/Recon/Imaging
Telecommunications
Ground Based Sensors

OC-ALC/DAC
MGen Johnson

HFGCS

AFPEO/AT
BGen Chedister

NAS JPALS GATM

Program Name

	Start Date	Completion Date	Comments
IP			
XML			
URL			
Browser-based Applications			

POC: Name/Office/Phone/email

SECRETARY OF THE AIR FORCE
WASHINGTON

JUL 9 2001

MEMORANDUM FOR ALMAJCOM-FOA-DRU/CC

SUBJECT: Information Technology (IT) Web-enabling Technologies and Standards for AF Applications

Information Technology (IT) Systems and National Security Systems (NSS) must be interoperable within the Air Force, among the joint services and other communities of interest. Currently, the wide variety of standards limit information support to our warfighters, decision support, and Command and Control processes across the Air Force. To move ahead, we must integrate web-enabling technologies and standards to govern information interchange and promote greater interoperability.

The following four IT commercial web-enabling technologies and standards are Air Force requirements for Joint systems that apply to all Air Force systems. Air Force Community of Interest domains will migrate to these standards.

a. Internet Protocol (IP)
b. Extensible Markup Language (XML)
c. Web/Universal Resource Locator (URL)
d. Web Browser (HTTP/HTTPS/HTML supported)

All systems in development will apply these four technologies and standards, and plans must be developed and executed to migrate legacy systems. Attachment 1 provides details on the fundamental technologies.

Lead MAJCOM, Functional Area, FOA and DRU commanders are responsible for new IT/NSS development and legacy system maintenance and modernization. The Air Force-CIO will review IT/NSS program standards migration plans as part of AF POM preparations to develop a consistent and cost effective IT investment strategy compliant with the Clinger-Cohen Act.

MICHAEL E. RYAN
General, USAF
Chief of Staff

JAMES G. ROCHE
Secretary of the Air Force

Attachment 1

Amplification of Information Technology (IT) Web-enabling Technologies and Standards for AF Applications

1. The following is an expansion of the IT Standards outlined in paragraph 2 with applicable notes

a. Internet Protocol (IP) applied at the network layer for communications systems* to provide a foundation for seamless end-to-end communications for AF/Joint Command, Control, Communications, Computer, Intelligence, Surveillance and Reconnaissance (C4ISR) systems,

*Certain networks, e.g., space, airborne data links, 'special use' airborne/terrestrial circuits, etc. will continue to use special protocols to satisfy unique warfighter/user performance requirements. These special networks will connect to the IP backbone through translating gateways.

b. Extensible Markup Language, Intelligence, Surveillance and Reconnaissance (C4ISR) systems, data exchanges and to the maximum extent possible, intra-Domain COI data exchanges: to facilitate publish and subscribe mission relevant information brokering vital to effective Information Management,

c. Web/Universal Resource Locator (URL) enabled AF applications and systems: web-enabled applications access should provide the maximum functionality possible to facilitate the establishment of Warfighter Portal services and data access for all missions the AF supports, and

d. Web Browser (HTTP/HTTPS/HTML* supported) entry to all AF IT/NSS systems in development and for legacy systems where practical to facilitate ready application data access and (self-service) system operation.

* "Secure Sockets layer (SSL) protocol or better" mandated by Department of Defense Public Key Infrastructure (PKI) and PKI Enabled Applications memo, OASD/C3I, November 2000

2. **Exemption Waivers to Applying Web-enabling Technologies and Standards:** Certain IT/NSS systems will continue to use special protocols, message formats, and presentation methods to satisfy unique military requirements or to support legacy technologies. These special IT/NSS systems should use translation mechanisms (e.g., gateways) to enable information exchange across the standards-based IT infrastructure described herein. Furthermore, it is recognized that the feasibility of legacy system migration is often dependent on diverse political, programmatic and economic considerations. Consequently, valid business case substantiated analyses in support of exemption waivers against any web-enabling standards dictates in this Policy Memo will be considered.

Questions regarding application of IT architectures and standards should be directed to Mr. Gilligan or Mr. Phil Huber in the AF-CIO at james.huber@pentagon.af.mil, DSN 329-3574 or (703) 601-3574.