Department of the Navy XML Registry Requirements

Version 1.0

September 2003

Preface

To support the reuse and interoperability of Extensible Markup Language (XML) objects[1] within the Department of the Navy (DON), the DONXML Work Group has formulated requirements for an XML registry. This report presents those requirements. It also addresses a wide range of implementation issues, including architecture, users, security, content, content organization, and interoperability.

An XML registry stores XML objects and the contextual information (metadata) that enable users to easily discover and adapt the XML objects for their own implementations. XML objects include XML schemas, DTDs, XML documents, and style sheets, as well as XML components such as XML elements, attributes, and data types. In formulating its requirements for the registry, the DON has drawn upon the work of the ebXML registry technical committee and the federal XML registry work group. By taking advantage of these and other XML registry capabilities, the registry will be able to serve the entire enterprise and meet the DON objective of facilitating reuse of interoperable XML with standard objects.

A draft of these requirements was distributed to DON commands for comment and a survey was taken to verify registry features and the potential for registry replication. The results of the survey confirmed support for the proposed requirements and led to additional requirements for search capabilities and Web services support.

[1] The term “XML objects” includes entities such as XML schemas, DTDs, XML documents, and style sheets, as well as XML components such as XML elements, attributes, and data types.

Revision History

Each version of this document issued is tracked in the following table.

Version	**Date (yy.mm.dd)**	**Section(s) Affected**	**Description**
0.1	02.09.16	All	DON XML WG, Team 3, Goal #1 starting draft.
0.2	02.10.17	All	DONXML WG, Team 3 Goal #1 modifications to address comments from Team 3 on version 0.1..
0.3	03.01.02	All	DONXML WG, Team 3 Goal #1 modifications to address comments from Team 3 on version 0.2. Most notably to overhaul Chapter 2 to replace prescription of architecture with consideration for designers of the architecture.
0.4	03.01.24	All	Version 0.3 was sent through an editor in advance of the DONXML WG meeting in January 2003.
0.5	03.06.24	All	Applied information learned from survey of DON on XML and registry requirements.
1.0	03.09.05	All	Final revisions to the draft in producing version 1.

Contents

FIGURES

TABLES

Chapter 1
Introduction

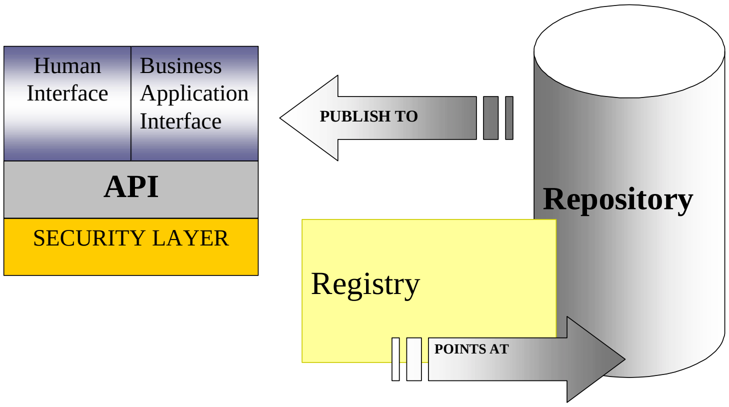
To support the reuse and interoperability of Extensible Markup Language (XML) objects[1] within the Department of the Navy (DON), the DONXML Work Group has formulated requirements for an XML registry. This report presents those requirements.

WHAT IS AN XML REGISTRY?

An XML registry is an information system that stores XML objects and contextual information (metadata) about the objects (or registered objects[2]). The storage facility (which may be a file system or database) is known as a repository. The part of the information system that maintains the metadata for the registered objects is a registry, while registry records containing information about particular registered objects are registry entries.

Figure 1-1 depicts the relationship between a registry and a repository.

Figure 1-1. Interaction Between a Registry and a Repository



Note: API = application program interface.
Source: ebXML.

A registered object does not need to be stored in a repository connected to the registry where it is registered (i.e., a unique identifier [UID] for the object may reference the location of the object in an "external" repository[3]). A registered

[1] XML objects are comprised of XML entities such as XML schema, document type definitions (DTDs), XML documents, and style sheets, in addition to XML components such as XML elements, attributes, and data types.

[2] Registered objects may also be non-XML objects.

[3] An external repository is not connected to the registry in which an object is registered.

object may also be referenced on the web through a uniform resource locator (URL) associated with the object.

In some XML registry standards, additional metadata for a registered object may reside in the repository rather than in the registry. Throughout this report, we assume that the metadata location is unimportant.

PURPOSE OF AN XML REGISTRY

The purpose of an XML registry is to provide a centralized source of standard constructs that enable platform-independent publishing and business transactions. Examples of possible business applications for an XML registry include:

- Financial management
- Transportation/shipping
- Materials management
- Personnel management
- Healthcare services.

BENEFITS OF AN XML REGISTRY

The benefits to the DON of an XML registry are numerous:

- *Promotes efficient discovery and maintenance of XML objects*—It allows DON agencies, trading partners (such as DoD components), and contractors to easily register, discover, and maintain DON XML objects.
- *Enables efficient version control*—It supports efficient tracking of multiple versions of a registered object.
- *Promotes unified understanding of registered objects*—It promotes an understanding of the purpose of the registered objects throughout the DON by making metadata available in a common environment.
- *Ensures availability and reuse of authoritative XML*—It promotes interoperability among trading partners and reuse by developers of new services because authoritative sources control the registration and validation of XML objects.
- *Promotes selective access to registered objects*—It ensures appropriate (read-only or open) access to registry content is granted according to the DON's security needs.

- *Enables collaborative development*—It supports the use and potential enhancement of XML objects by authorized parties, so an XML registry can be a common point for recording iterative development objects.

Registry Operations and Responsibilities

The DONXML Work Group is also developing a concept of operations (CONOPS) to describe the DON's perceptions of how the registry will be used and the roles of users in interacting with the registry. The registry CONOPS is expected to be made available shortly after the release of this requirements document.

Modes of Operation

This report addresses the two primary conditions under which an XML registry may function:

- *Defense condition*—The XML registry, and many of the information systems that may interact with the registry, needs to accommodate drastic changes in security and user access as systems react to threats according to Information Operations Conditions (INFOCONS).[4]

- *Afloat/ashore environment*—The XML registry needs to recognize the logistical and technical requirements that differentiate ashore versus afloat systems. The registry architecture will be a key element in ensuring that both environments are served adequately.

Conventions Used

The following conventions are used throughout this report:

- The words "must, must not, shall, shall not, should, should not, and may" are to be interpreted as described in Internet Engineering Task Force (IETF) Request for Comments (RFC) 2119.[5]

- The term "XML registry" is used instead of "XML registry/repository," because it is not always necessary to identify whether the registry has a local repository. In references to an object in an XML registry, the object is actually stored in the repository, while the metadata for that object are recorded in the registry.

- The term "XML object" refers to any XML entity that may be registered such as XML schemas, DTDs, XML documents, and style sheets, as well as elements and attributes.

[4] Department of Defense memo, *Information Operations Condition*, March 10, 1999.

[5] Accessed on the Internet at http://www.faqs.org/rfcs/rfc2119.html.

- The term “XML component” refers to XML objects such as elements, attributes, and data types that are used to construct objects such as schemas and documents.

- The term “registered object” refers to an XML object or non-XML object stored in an XML registry.

- The term “metadata” refers to contextual information for a registered object, regardless of whether the object resides in a registry or a repository.

REPORT ORGANIZATION

The remainder of this report is organized in the following order:

- Chapter 2 describes various registry architecture issues.

- Chapter 3 discusses a wide range of topics, including accessing the XML registry and messaging protocols.

- Chapter 4 covers various security-related topics such as general security concepts, including public key infrastructure (PKI), open XML-related security standards, and what these mean for the XML registry.

- Chapter 5 describes the attributes of metadata and the types of objects that may be stored in the XML registry.

- Chapter 6 examines the techniques for organizing registry contents such as associations, classifications, and namespaces.

- Chapter 7 discusses open registry standards and existing registries with which interaction may be highly desirable or necessary, in addition to compliance requirements with federal mandates.

- The appendix contains a summary of the proposed requirements from all the chapters.

Chapter 2
Proposed XML Registry Architecture

This chapter addresses two architecture issues for the proposed XML registry: networks and performance.

NETWORKS

This section examines DON-related networks and their possible applicability to the XML Registry.

Defense Information System Network

The Defense Information System Network (DISN) is the major defense network supporting Command, Control, Communications, Computers, and Information (C4I) objectives of an integrated and interoperable network for joint task force operations.[1]

NIPRNET/SIPRNET

The Non-Classified Internet Protocol Router Network (NIPRNET), created in 1995, is a network of government-owned Internet protocol routers for exchanging sensitive information. The Secret Internet Protocol Router Network (SIPRNET) is the secret portion of DISN. Each network is a candidate for hosting all or some portions of an XML registry. If both networks accept registration of unclassified objects, the synchronization of the registries will need to filter out classified objects on SIPRNET. An alternative would be to limit registrations of unclassified objects to a NIPRNET registry and classified objects to a SIPRNET registry. This alternative would require unclassified and classified versions of each namespace and external associations from SIPRNET to NIPRNET. DoD’s XML registry operates on both these networks.

Coalition Wide Area Network

The Coalition Wide Area Network (CWAN) provides C4I connectivity and interoperability among the commands of the United States and those of its allies. The specific networks that comprise CWAN could not be identified for this report.

[1] Defense Information System Network Mission Need Statement, March 30, 1995.

Joint Worldwide Intelligence Communications System

The Joint Worldwide Intelligence Communications System (JWICS) is a component of DISN targeted at secured video and data communications. Because of its focus on secured video, JWICS requires throughput that can accommodate large data transfers. JWICS has several fixed and contingency nodes covering all major commands.[2] The intelligence community has established a version of DoD's XML registry on JWICS.

Future Naval Networks

Additional network connectivity can be expected throughout the DON, although network architectures may need to be layered into categories of networks based on their security requirements.

PERFORMANCE ISSUES

The specific performance demand that would be placed on the registry has yet to be determined. Registry architects will need to evaluate various factors to determine an appropriate solution.

Afloat Registry Considerations

Shipboard XML transaction validation and development will require the registry to be fully accessible even while ships are deployed. The DON has three options for meeting this requirement.

- Option 1: A land-based registry would be available 24 hours a day, 7 days a week to ships through a global wireless mechanism; however, bandwidth and emission control (EMCON) requirements could complicate continuous access.
- Option 2: A registry would be aboard every ship that has XML-dependent applications. Although bandwidth could still be an issue, a CD-ROM, or similar storage device, would be required onboard as a backup and could provide the initial load of the registry. Each ship would also require a server to host the registry. The most demanding user of the registry will likely be the XML parser, so the best location for a shipboard registry would be near the XML parser.
- Option 3: XML validation would be built into every application that uses XML aboard every ship and limit or deny shipboard development that could create conflicts with the approved objects of the registry.

[2] Accessed December 2002 on the Internet at http://www.globalsecurity.org/intell/library/reports/2001/compendium/jwics.htm.

Although the third option is quite attractive, integrating validation into applications raises several issues:

- *Application burden*—Each application accepting XML messages would need its own separate validation routine.
- *Single purpose*—Because XML validation is not the main purpose of the application, validation routines tend to be bare bones and targeted specifically to individual applications. Reuse of the code typically would require customization for use in other applications.
- *Static implementations*—The highly tailored nature of application-based validation also tends to make upgrades to new releases difficult. If an entirely new technology for transaction processing becomes available, the ability to transition to the new technology will depend on individual applications.
- *Technical support burden*—The technical staff supporting applications aboard the ship would need to be familiar with the nuances of each validation routine. If the ship's software administrators do not have access to the routines, they may not be able to troubleshoot issues that arise in validating transactions within the applications.

Application-neutral validation routines require the ability to expand their functions to numerous transactions and associated business rules. These routines typically can automate upgrading to new releases and they are likely to have debugging tools to assist in troubleshooting common problems. If a better technology emerges, the routines will need to be capable of transitioning the business activities from XML to the new technology without a requirement to modify the applications.

Ashore Registry Considerations

Most land-based XML development and validation will probably occur where access and system performance for a registry on a network such as NIPRNET would not be a problem. A survey of DON commands on XML registry requirements revealed that about 12 commands believed they would seek to have a localized version of the registry employed by the DON.

Some installations could host synchronized registries to provide localized access points where ships could synchronize registries. Even if direct access to a primary registry is feasible, installation copies of the registry could provide backup for ship support.

Regardless of near-term registry architecture requirements, the number of XML-enabled systems and development work should grow significantly over the years. Building an easily scalable infrastructure is part of the ultimate architecture.

PROPOSED REQUIREMENTS

Table 2-1 lists proposed requirements for the topics in this chapter.

Table 2-1. XML Registry Architecture—Proposed Requirements

ID no.	Topic	Proposed requirement	Justification
2.1	Networks	Unclassified registry entries and registered objects *must* be duplicated to the classified registries. Unclassified registry entries and registered objects submitted to a classified registry *must* be verified as unclassified before they can be synchronized with an unclassified registry. Classified objects submitted to an unclassified registry *must* be rejected. Registry *shall* provide access for external consultants and trading partners. Registry *should* evaluate portals to the CWAN and JWICS networks.	NIPRNET and SIPRNET are likely environments for hosting registries, particularly for registration of web services.[a] It is assumed that registries on NIPRNET and SIPRNET should cover the widest possible group of users; however, DON participants in the U.S. intelligence community, served by the CWAN and JWICS networks, are participating in an effort to establish an XML registry for the intelligence community. Synchronizing unclassified objects between the two environments introduces additional complexity and risks to isolate classified objects in SIPRNET. However, the ability to submit and administer objects in one area is preferred to managing objects in different registries. Submissions to a classified registry are assumed to infer classified status unless explicitly designated by the submitter to reduce the risk of classified objects being ported to an unclassified registry during synchronization. The development and testing of XML-based applications may involve contractors operating off-site. Trading partners will need access to the registry to discover DON-approved standards for conducting business and to register their capabilities.
2.2	Performance issues	Registry requirements unique to shipboard platforms *mus*t be accommodated. Registry must be capable of supporting a replicated registry architecture. Registry architecture *must* be scalable.	XML is expected to be introduced as a major technology to support administrative systems and weapon systems. The use and development of XML on afloat systems requires a registry architecture that can take into account the ability to replicate registry content to afloat systems, particularly during the long periods of deployment between synchronizations.[b] Respondents to a survey of registry requirements indicated that they believe their system environment will require them to operate a localized version of the DON registry. Such support would also help facilitate the disaster recovery and continuous operations requirements discussed in Chapter 3. As more XML applications and services are produced, the demand on the registry will grow.

[a] Based on Chief of Naval Operations (CNO) memo, Request for Implementation of Joint-Allied Web Services Interoperability, May 22, 2002.

[b] Afloat synchronization is discussed further in the DON XML Registry CONOPS.

Chapter 3
XML Registry Access, Usage, and Administration

This chapter addresses several registry topics:

- Registry access
  - *User access*—How should users access the XML registry?
  - *Type of access*—Should the XML registry be accessed at development-time, run-time, or both?
- Registry usage
  - *Registry users*—Who should be authorized to submit objects to the XML registry and download objects from it?
  - *Search capability*—What kinds of registry searches should be available?
  - *Submission validation*—Should XML objects be validated on submission to ensure they are well-formed[1] and valid?[2]
  - *Messaging protocols*—What protocols should be used to send messages to and from the XML registry?
  - *Publish/subscribe*—Should publish/subscribe functionality be included in the XML registry?
  - *Object construction*—What support should the registry provide developers in constructing objects such as schemas?
- Registry administration
  - *Registered object life cycle*—What are the various phases that a registered object passes through?
  - *Logging/audit trail*—How much and what type of information should be maintained in a registry on actions in the registry and its contents?
  - *Disaster recovery/continuity of operations*—What system operational functions are needed to ensure a system such as the XML registry can remain in operation if a disaster occurs?

[1] Well-formed objects means that they conform to the definition of valid XML as stated in the XML 1.0 specification at http://www.w3.org/TR/2000/REC-xml-20001006.

[2] Valid objects means that an XML document conforms to its specified schema or DTD (if one is specified).

REGISTRY ACCESS

User Access

Several common methods exist for allowing user access to a registry. They include:

- Executing a program installed on a user’s client machine (also known as “thick client”)
- Executing a program that resides on a server (also known as “thin client”).

Implementing any of these methods must be consistent with DON architectures. DON efforts to support a net-centric approach through the Navy Enterprise Portal (NEP) suggests that the enterprise registry should be accessible through the NEP.

Type of Access

A registry has two primary types of access:

- Development-time
- Run-time.

DEVELOPMENT-TIME ACCESS

Development-time registry access means a user manually downloads a registered object (such as an XML schema). All interaction with the registry is accomplished by direct user action. A user may also access an XML schema that resides in a registry via another XML schema that has the UID or URL of the registered schema.

RUN-TIME ACCESS

Run-time registry access means interaction with the registry is automated at the time the dependent application is executing an XML-based process. Three examples of run-time access are provided below:

- Execution of a query for a registered object.
- Reference by UID or URL within an XML document of an XML schema stored in a registry; the XML document is then validated against that schema.
- An object submitted to one registry is automatically uploaded to a second registry, based on rules that determine the registry where objects are stored on submission.

REGISTRY USAGE

Registry Users

Registry users will have various authorizations. For example, one group of users may be allowed to submit to the registry, download from it, and update existing registered objects. Another group may be allowed to download only registered objects. Listed below are several types of registry users:

- *Guest*—Read-only access users who need to look up information occasionally.
- *Working*—Users who submit objects, collaborate on XML development, and conduct work associated with governance activities over registering objects.
- *Functional namespace coordinator (FNC)*—Same as working-type users, but with additional capabilities to oversee submitted and registered objects.
- *Registry administrator*—Users who maintain the registry and troubleshoot issues encountered by other users.
- *Automated information system (AIS)*—An authenticated information system with permissions to interact with the registry to conduct e-business transactions.

Search Capability

The registry needs an effective search capability. Searches based on user-supplied key words that involve Boolean expressions will give users an ability to identify possible components for reuse. The ability to extend searches to partner registries will further widen the user’s pool of resources and promote greater interoperability.

General users, but especially namespace managers, can benefit from search filters that narrow searches to particular namespaces of interest. Similarly, filters for object status can focus results only to those that are at a particular stage of the life cycle. Combining namespace filters with object status can significantly improve a registry’s effectiveness. For example, implementers could focus on only approved components for a particular namespace, while namespace managers could quickly identify all the objects in their namespace under development.

Submission Validation

A “well-formed” XML document conforms to the XML syntax rules. A “valid” XML document is a well-formed XML document that also conforms to the business rules of a DTD and schema. A registry can validate XML objects on receipt to ensure they are well formed and valid. This feature could also be offered as an option to registry users.

Messaging Protocols

Several XML-based messaging protocols are currently in use:

- *XML-RPC*—A specification developed by UserLand Software that allows software running on disparate operating systems and in different environments to make procedure calls over the Internet. It uses HTTP as its transport mechanism and XML to encode the message.

- *SOAP*—Simple Object Access Protocol (SOAP), the most widely used XML-based messaging protocol, is a specification for invoking methods on servers, services, and objects. It provides an open, extensible way for applications to communicate using XML-based messages over the Web, regardless of operating system, object model, or language applications use. SOAP uses HTTP and Simple Mail Transfer Protocol (SMTP) as its transport mechanism and XML to encode the message.

- *OASIS/ebXML Messaging Services*—A specification that defines a communications protocol-neutral method for exchanging electronic business messages over a communications protocol such as HTTP or SMTP. It defines a flexible enveloping technique that permits messages to contain payloads of any format type (such as XML, UN/EDIFACT, ASC X12, or HL7). The ebXML Messaging Service (ebMS) is a set of layered extensions to the base SOAP and SOAP Messages with Attachments specifications that provides additional security and reliability features.

- *Publish/Subscribe*—A practice in which registry users subscribe to various registered objects and are notified of changes or deletions to those objects. For example, users who subscribe to a specific XML schema are notified whenever the schema is updated.

Intent to Develop

The process for registering XML must include procedures for notifying others that development of a particular business activity is underway. Early notification through the Business Standards Council (BSC) that a project is in development allows the BSC to make the developer aware of related work before substantial effort has been invested. This process can also result in encouraging similar efforts to collaborate.

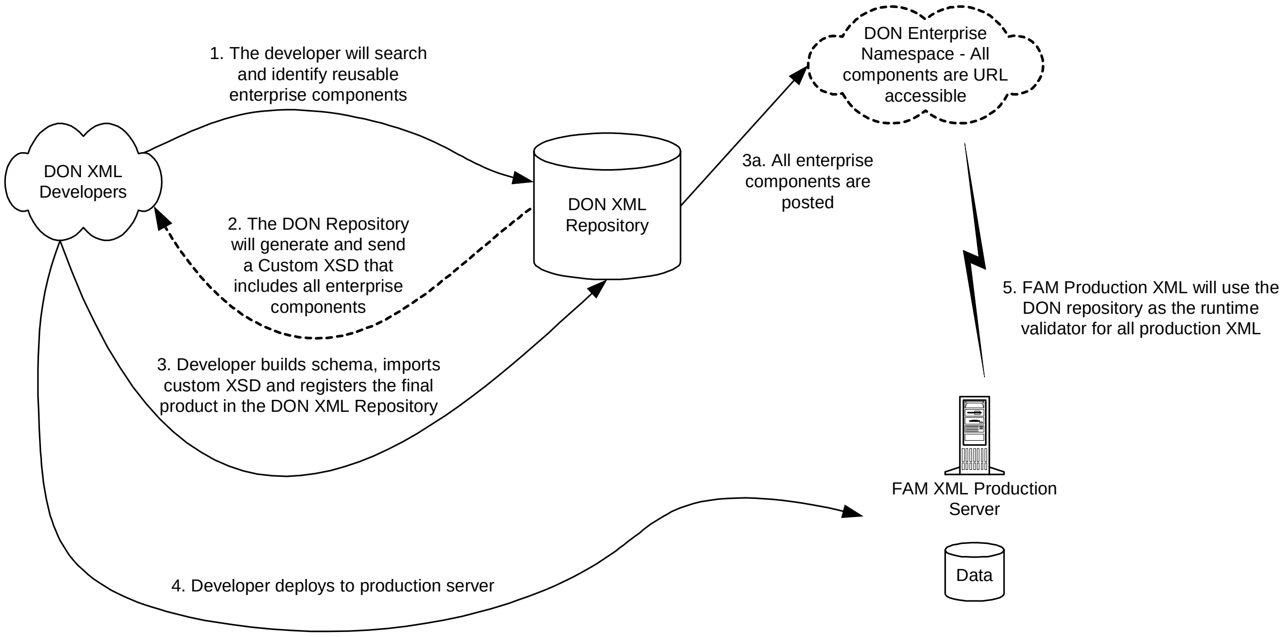
FNCs will use the registry as a source of information on new entries. They will be able to compare the provided information to other entries for purposes of identifying duplicates and facilitating notification of others about the existence of new XML projects.

In addition, Navy Knowledge Online (NKO) has provided a demonstrated capability for facilitating DON collaboration work. The DON would like a mechanism for integrating the collaboration support tools of developers and reviewers.

Object Construction

The DON would prefer that developers follow a modular construction process that reuses aggregated components when building objects such as schemas. The following figure roughly depicts the DON vision for how the registry would assist in the development of schemas.

Figure 3-1. DON XML Developer Registration Process


DON Enterprise Namespace - All components are URL accessible
1. The developer will search and identify reusable enterprise components
DON XML Developers
DON XML Repository
3a. All enterprise components are posted
2. The DON Repository will generate and send a Custom XSD that includes all enterprise components
5. FAM Production XML will use the DON repository as the runtime validator for all production XML
3. Developer builds schema, imports custom XSD and registers the final product in the DON XML Repository
FAM XML Production Server
Data
4. Developer deploys to production server

Note that this process could be ported to reusing components for constructing other objects such as stylesheets.

Taking this vision a step further, the DON would encourage the development of objects from components verified to represent technology neutral models. Developers could then construct their customized objects by identifying models and having the registry compile components using XML associated with the model. Chapter 5 expands on this feature.

REGISTRY ADMINISTRATION

Registered Object Life Cycle

The DON BSC Operating Procedures establish the phases and statuses of a DON XML object life cycle. The registry will track the status of XML objects using the following definitions:

- *Non-standard*—Legacy objects in use, not considered candidates for general reuse or an enterprise standard.
- *Developmental*—Objects not yet submitted for formal BSC review and not employed in production systems.
- *Submitted*—Objects currently being reviewed to become a DON enterprise standard.
- *Rejected*—Objects rejected as an enterprise standard during the BSC review process because of their failure to comply with approved technical or business standards.
- *Approved*—Objects formally accepted by the BSC as a registered DON enterprise standard and approved for enterprise-wide reuse.
- *Deprecated*—Objects phased out of use or version obsolete.

Logging/Audit Trail

A registry should create an audit trail every time it carries out an operation. At a minimum, the audit trail should include the following types of information:

- *User ID*—The user who performed an operation.
- *Operation*—The action the user performed (e.g., “user submitted object”).
- *Date/time*—When the operation was performed.
- *Object ID*—The object of the operation.

A registry should also provide the capability to generate reports.

Disaster Recovery/Continuity of Operations

A serious consideration in implementing an XML registry is the capability to recover from a disaster. A frequently used method is to establish a "hot site" that contains a copy of the system. The hot-site copy allows external systems and users to continue operational interactions with the registry during a disaster recovery.

PROPOSED REQUIREMENTS

Table 3-1 lists several proposed requirements for XML registry access, usage, and administration.

Table 3-1. Proposed Requirements for XML Registry Access, Usage, and Administration

ID no.	Topic	Proposed requirement	Justification
3.1	User access	Registry *must* be accessible through DON-approved network protocols.[c] Registry *must* support an NEP portal.	Human and automated users will look to access registries through various methods such as the Web and networks supported by DON policy.
3.2	Type of access	Registry *must* provide both development-time and run-time access.	System developers need development-time access to discover existing objects and collaborate on new objects. Automated systems need run-time access to validate documents received against registered schemas.
3.3	Registry users	Any DON agency, authorized trading partner, or authorized contractor *must* be able to access the unclassified XML registry, and the classified XML registry, as appropriate. Any DON agency, authorized trading partner, or authorized contractor *must* be able to submit to the unclassified XML registry, and the classified XML registry, as appropriate.	DON agencies will access the registry to support their systems. Authorized trading partners will access the registry to discover objects that support their transactions and maintain their profiles. Authorized contractors will access the registry to support development and maintenance of DON systems.

Table 3-1. Proposed Requirements for XML Registry Access, Usage, and Administration (Continued)

ID no.	Topic	Proposed requirement	Justification
3.4	Search capability	Registry *must* allow for key word searches. Registry *must* allow for Boolean defined searches. Registry *must* allow for integrated searches with federated registries. Registry *must* allow for searches based on functional namespace. Registry *must* allow searches based on object status. Registry *must* allow for searches of a functional namespace filtered for object status. Registry shall provide for the identification of potential duplicate object names or description.	Key word and Boolean searches will provide users with a minimum functionality to focus their searches for relevant components. The DON is promoting extending interoperability up through DoD, federal, and standards groups. Connecting searches to registries established by those groups will improve the DON's ability to identify and adopt relevant external components. At a minimum, namespace managers require the ability to identify objects within their namespace at each of the life-cycle statuses. General users will also benefit from this capability.
3.5	Submission validation	Registry *must* support checking well-formedness and validity of submissions when submitted to the registry. User *must* be able to turn on or off based on the type of submission. Registry *must* maintain sufficient metadata to indicate validation status.[d]	To be a relevant registry, approved objects must demonstrate that they are well formed and valid. Such checks can be performed outside the registry, but a more effective method would be to give users the option to execute checks when submitting. Because tools to check well-formedness and validation are inconsistent, the registry must allow submitters to bypass automated checks with the understanding that the submitter will provide external proof. Subscribers to objects need the registry to provide adequate metadata to express if an object has passed well-formedness and validity checks.
3.6	Messaging protocols	Registry *must* support SOAP. Registry *must* support ebMS.	SOAP is the widely accepted mechanism for XML transactions expected from automated systems. ebMS expands on SOAP to provide additional security capabilities.

Table 3-1. Proposed Requirements for XML Registry Access, Usage, and Administration (Continued)

ID no.	Topic	Proposed requirement	Justification
3.7	Publish/subscribe	Registry *must* contain publish/subscribe functionality. Registry *must* allow for AISs to subscribe to objects.	To support many interoperable systems, users need the capability to receive automatic registry notification of modifications that could affect their implementations. Since individuals frequently change locations and positions, it is important to be able to identify the AISs that are potentially impacted independent of original users.
3.8	Intent to develop	Registry *must* contain a mechanism for XML developers to declare their intent to develop new XML constructs and define points of contact. Registry *must* provide for a virtual workspace to support collaborative development efforts.	To support interoperability, users need to be involved with important development projects as early as possible. Retrofitting objects after implementation can be logistically difficult and expensive.
3.9	Object construction	Registry must be capable of auto-generating a schema from a developer's selections of registered components. Registry should be capable of auto-generating other XML objects from modular components.	To better facilitate the reuse of existing components within the registry, the DON is planning on the registry to be able to construct XML objects by aggregating registered components.
3.10	Registered object life cycle	Registry *must* support life-cycle functionality for registered objects. Registry *shall* support at a minimum the following life-cycle statuses: non-standard, development, submitted, rejected, approved, and deprecated. Registry shall provide for the recording of information related to an object's review by the DON.	To keep implementations up to date, the registry must make it clear to users when an object is in development, approved for implementation, or obsolete. The minimum status types are required to support BSC Operating Procedures.

Table 3-1. Proposed Requirements for XML Registry Access, Usage, and Administration (Continued)

ID no.	Topic	Proposed requirement	Justification
3.11	Logging/audit trail	Registry *must* contain adequately robust logging and audit trail functionality that includes at a minimum: - User ID - Operation performed - Date and time - Object UUID. Registry *should* include "standard" audit trail reports.[e] Registry *may* include user-defined audit trail reports.	To make properly informed decisions and follow-up on issues, users reviewing submissions need to know the history of changes to the object and who made them.
3.12	Disaster recovery and continuity of operations	Registry *must* implement disaster recovery capabilities. Registry *must* operate with a continuity of operations plan.	For the registry to support critical systems, a disaster recovery plan and a continuity of operations plan must be established.

[c] This feature would allow the XML registry to be accessible from any Web-enabled device such as hand-held devices and browsers.

[d] XML registry must maintain metadata such as whether a submission was validated, whether it was validated by the registry or the submitter (or both), and the tools used for validation.

[e] "Standard" means reports included with the registry software, as opposed to user-defined.

Chapter 4

XML Registry Security

This chapter addresses the following registry security topics:

- General security concepts
  - *Public key infrastructure (PKI) and digital certificates*—Security mechanisms that enable organizations to protect the security of their communications and business transactions on the Internet.
  - *Digital signatures*—Technology that allows the recipient of a message or file to verify the authenticity of the sender and integrity of the data.
  - *Secure sockets layer*—Security protocol that provides privacy and authentication for network traffic.
  - *Access control policies*—Software-enforced policies that ensure only authorized parties have access to certain information and can perform only authorized actions.
- *Changes in security and user access requirements*—Explains the need for changes in security and user access requirements as threat conditions change.
- *Open XML-related security standards*—Discusses various open XML-related security standards from W3C and OASIS.
- *High-level security functional categories*—Lists various high-level categories relating to security and describes how each is satisfied based on preceding concepts and standards.

GENERAL SECURITY CONCEPTS

Public Key Infrastructure and Digital Certificates

PKI consists of software, encryption technologies, and services that, when combined, protect the security of organizational communications and business transactions on the Internet. In PKI, a digital credential, known as a digital certificate, is used to assert the identity of an Internet user. Digital certificates may be server certificates or personal (client) certificates, and they may reside on a server machine, a client machine, or both.

ISO standard X.509 specifies the format and content of digital certificates, and certificates that comply with this format are referred to as “X.509 certificates.” All X.509 certificates contain information such as the following:

- *Serial number*—A unique number that distinguishes the certificate from others issued by the entity (known as a “certification authority” or CA) that created the certificate.
- *Signature algorithm*—The algorithm used by the CA to sign the certificate.
- *Validity period*—The length of time the CA considers the certificate to be valid.
- *Public key information*—Information the holder of the certificate uses to verify digital signatures.

All prominent web browsers support X.509, version 3, certificates.

Digital Signatures

A digital signature is an encrypted electronic fingerprint of a message or file that is uniquely tied to the message or file content. It allows the recipient of a message or file to verify the authenticity of the sender and ensure the author of a message or file cannot later deny creating the message or file (i.e., repudiation cannot occur). Digital signatures can also authenticate message integrity (i.e., ensure that a message or file has not been altered).

Secure Sockets Layer

The secure sockets layer (SSL) protocol, developed by Netscape, is the leading security protocol for the Internet. It provides privacy and authentication for network traffic. SSL allows a web browser to establish a secure session connection with a server by allowing the server to identify itself to a client browser using an X.509 certificate. The browser and server can then exchange data using secret key encryption. SSL was recently combined with other protocols and authentication methods into a new protocol Transport Layer Security (TLS).

Access Control Policies

An access control policy uses software to ensure that only authorized users have access to authorized information and can perform only specific actions. In a registry, this policy ensures that only authorized users can perform certain actions on registered objects (e.g., submit, access, update, or delete). Access control policies are often implemented by assigning roles to users and granting certain permission rights to those roles. For example, a role of “submitter” may include the rights to submit, access, and update registered objects, but not to delete them.

Similarly, a role of "administrator" may include the same rights as a "submitter," plus the right to delete registered objects.

CHANGES IN SECURITY AND USER ACCESS REQUIREMENTS

The XML registry, and many of the information systems that interact with it, will need to accommodate changes in security and user access requirements necessitated under INFOCONS. For example, access control policies and the roles they enforce may need to be changed according to altered security constraints.

OPEN XML-RELATED SECURITY STANDARDS

Several open XML-related security standards are being developed. A few of the most prominent efforts are summarized below:

- *XML signature*—The W3C XML Signature Specification[1] specifies XML digital signature processing rules and syntax. The XML Signature Working Group has developed an XML-compliant syntax for representing the signature of web resources and portions of protocol messages, and procedures for computing and verifying such signatures.

- *XML encryption*—The W3C XML Encryption Specification[2] is a process for encrypting data and representing the result in XML. The result of encrypting data is an "EncryptedData" element that contains or identifies the cipher data.

- *XML key management*—The W3C XML Key Management Specification[3] specifies protocols for distributing and registering public keys, suitable for use in conjunction with the XML signature and XML encryption specifications. The XML Key Management Specification comprises two parts—the XML Key Information Service Specification (X-KISS) and XML Key Registration Service Specification (X-KRSS).

- *SAML*—The OASIS Security Assertion Markup Language is an XML-based security standard for exchanging security information. This security information is expressed in assertions about subjects, where a subject is an entity (either human or computer) that has an identity in some security domain. SAML can be used to transfer credentials for distributed registries.

- *XACML*—The OASIS Access Control Markup Language is an XML specification for expressing policies for information access over the Internet. The XACML Technical Committee is defining a core schema

[1] W3C recommendation, February 2002.

[2] W3C candidate recommendation, August 2002.

[3] W3C working draft, March 2002.

and corresponding namespace for the expression of authorization policies in XML against objects that are themselves identified in XML.

HIGH-LEVEL SECURITY FUNCTIONALITY CATEGORIES

Table 4-1 lists several high-level security functionality categories and describes how each is satisfied based on the concepts and standards discussed earlier. This table ties together these concepts and standards under various umbrella categories.

Table 4-1. High-Level Security Functionality

Category	Description	How satisfied
User authentication	Determining the identity or role of a party attempting to perform some action such as accessing a resource or participating in a transaction	Digital certificates—X.509 certificate on client, SSL certificate on server[*] SAML
Non-repudiation	Preventing the author of a message or file from later denying creating the message or file	XML Digital signatures
Authorization	Enforcing access control policies to ensure that only authorized parties have access to certain information and can perform only specific actions	XACML
Message integrity	Ensuring that information has not been altered or tampered with during transmission	XML Digital signatures SSL/TLS
Confidentiality	Ensuring that content can be viewed only by intended parties, even when other access control mechanisms are bypassed	Encryption XML encryption SSL/TLS

[*] If PKI is too expensive, it is possible to implement user ID and password.

PROPOSED REQUIREMENTS

Table 4-2 lists several proposed requirements for registry security.

Table 4-2. XML Registry Security—Proposed Requirements

ID No.	Topic	Proposed requirement	Justification
4.1	Open XML-related security standards	Registry *must* use DON-approved open XML-related security standards.	A 2001Defense Authorization Act subsection on government information security reform directs the DoD to use NIST-specified security policies at a minimum. NIST security policies are based on open standards.
4.2	Changes in security and user access requirements	Registry *must* rapidly accommodate changing conditions in security requirements. The registry *must* be capable of restricting levels of access on demand.	INFOCONS details responses to threats posed to DoD information systems.
4.3	User authentication	Registry *must* employ user authentication mechanisms to ensure authentication of ownership of registry content. Authentication ensures the identity of the individual, but says nothing about the access rights of the individual.	To verify user rights granted under an account, the registry must authenticate the identity of all users. PKI is the preferred identification and authentication method, but user ID and passwords can also be used for this activity.
4.4	Non-repudiation	Registry *must* use non-repudiation mechanisms to ensure that repudiation of registry submissions does not occur.	To ensure the registry properly captures an action by a user, such as establishing trading partner agreements, the system must be able to prove under audit that the action was properly recorded and executed by the appropriate user.
4.5	Authorization	Registry *must* use role-based and organization-based access control policies to ensure the proper level of access to registry content is granted according to DON's security needs. Registry *must* support access control at the object level.	Secured systems often use access constraints based on organization (e.g., SIPRNET requires MIL domains); only individuals with a particular clearance are given access within organizations. Registry needs to limit access to certain objects by designating a subset of authorized users for security and control of early developmental projects.
4.6	Message integrity	Registry *must* use message integrity mechanisms to ensure that registry submissions have not been tampered with en route to the registry.	Content data submissions cannot be subject to changes in transit.

Table 4-2. XML Registry Security—Proposed Requirements (Continued)

ID No.	Topic	Proposed requirement	Justification
4.7	Confidentiality	Registry *must* provide confidentiality mechanisms during data transfer to ensure that transferred content is viewable only by authorized parties. Registry *may* use confidentiality mechanisms for stored content to ensure it is viewable only by authorized parties.[f] Registry *must* support DoD's PKI infrastructure.	No unclassified content document can be allowed to route to or from the directory along an unencrypted channel. Because the registry will protect against unauthorized access, restricted objects may not need encrypting inside the registry; such functionality may be desired in certain circumstances. DoD policy requires the use of PKI to support the common access card architecture.
4.8	Ownership of content	Registry *must* use ownership data for all components.	Ownership data is necessary for configuration management of changes and publish/subscribe capability.

[f] With only an SSL certificate on the server, the registry could provide data confidentiality through the encryption features of SSL V3 or TLS; however, to ensure that only authorized parties view registry contents, mutual authentication must be used.

Chapter 5
XML Registry Metadata and Contents

This chapter addresses three primary topics:

- *Registry metadata*—Object attributes[1] that should be maintained for registered objects in the XML registry.
- *Registry contents*—Types of objects that may be stored in the XML registry.
- *Registry extensibility*—Ability to expand object types and metadata in a registry without requiring a programmatic update to the registry software.

Registry Metadata

Listed below are several examples of metadata that can be maintained in a registry:

- *Name*—Human-readable name for the registered object.
- *Object type*—Nature of the registered object (e.g., XML schema, DTD, or XML document).
- *Description*—Purpose or definition of the registered object.
- *Version*—Distinction between two registered objects with the same UID.
- *UID*—Unique identifier for easy reference by automatic processes to a registered object.
- *URL, URI, URN*—Unique identifiers for a registered object that may be accessible through the Web.
- *Object status*—Life-cycle status for a registered object (such as approved or pending approval).[2]

In a registry, metadata attributes may be designated as “required” (must always be provided with submissions) or “optional” (may be provided with submissions).

[1] Metadata attributes are individual pieces of metadata.

[2] For the DON, object status also can include “developmental” and “operational.”

REGISTRY CONTENTS

The types of object that may be stored in the XML registry include:

- XML objects and non-XML objects
- Core components
- Business processes
- Trading partner profiles and agreements.

XML Objects and Non-XML Objects

XML objects are defined as entities such as XML schemas, DTDs, XML documents, and style sheets, and components such as XML elements, attributes, and data types.

Registry users can manually or automatically build XML objects from stored XML components. The registration of these components is transparent to users if registry software parses XML objects as they are submitted and automatically registers the XML components they contain. A registry can support the storage of non-XML objects, such as documentation[3] for an object.[4]

Core Components

Core components are basic data items business documents that describe common concepts used in general business activities. Because different industries use different terms to express the same ideas, businesses need a way to eliminate these semantic barriers and communicate with each other without asking organizations to change long-standing business practices. Using core components provides such a capability.

As an example, most business activities use their address in all correspondence. Basic data items that comprise an address—street, locality, state/province, and postal code—are core components. The high-level basic data item these core components comprise—an address—also is a component, an aggregate component comprised of several core components.

Context is important in core component because it gives meaning to core components. For example, in a purchase order, a business may record an address for the sender and the receiver. The context of sender and receiver clarifies the occurrence of their respective use in the address core component.

[3] Supporting documentation for non-XML objects can be Word documents, PDF documents, UML diagrams, Erwin diagrams, and others.

[4] See Chapter 6 for more detail.

The Core Component Technical Specification (CCTS) v2.0, which was recently approved by UN/CEFACT, includes the requirements for a registry to represent a core component. The ebXML Registry Technical Committee is using the CCTS specification to develop a normative reference document that describes how to include core components in an ebXML registry.

Business Processes

Business processes define the relationships called “collaborations” between business partners. They identify the parties in the transactions, the messages exchanged between the parties, the sequence of the messages, and the data elements in the messages. Business process models provide a means to define business activities by allowing organizations to represent these activities as general business models. Through collaboration, similar businesses can develop a general model that can be applied to many organizations. In turn, these models help define common information needed to perform the activity.

The general business models can be particularly useful to designers and implementers of business function objects.[5] A registry may include business process models and the objects those process use.

The Universal Business Language of OASIS is an example of a cross-functional attempt to model business activities. The UBL committee is using core components as the building blocks to model the business processes in UML.

Trading Partner Profiles and Agreements

A trading partner profile (TPP) describes the technological capabilities of a trading partner and the specific requirements for exchanging business documents with it. A TPP can include business processes an organization supports.

A trading partner agreement (TPA) defines the conditions needed for two partners to transact business with each other. A TPA, created manually or automatically from two TPPs, includes information such as the following:

- Business processes
- Messaging protocol
- Contingency issues
- Security requirements.

[5] An example of a business function object is an XML document containing a message in a business exchange.

Web Services

Web services are functional systems with interfaces defined and connected by XML. Registering web services allows registry users to determine what web services are available and information on how to use those services. Registry specifications for ebXML and UDDI, which are discussed in Chapter 7, include capabilities for registering web services. Other relevant specifications define the web services available in a registry such as Web Services Description Language (WSDL), ebXML Collaboration Partner Profile/Agreement (CPPA), and the developing DARPA Agent Markup Language-Services (DAML-S).

REGISTRY EXTENSIBILITY

The term "extensibility" refers to a level of expandability. In information technology, it describes a program, programming language, or protocol designed to allow users or designers to extend capabilities after implementation.

Extensibility can also apply to a registry, particularly in the following contexts:

- *Metadata extensibility*—The ability to expand allowed metadata attributes for a registered object without requiring a programmatic update to registry software.

- *Object type extensibility*—The ability to expand allowed object types for a registered object without requiring a programmatic update to the registry software.

Metadata Extensibility

The following scenario describes the concept of metadata extensibility.

Suppose in the 1970s someone created a paper form that described all features of a car, such as the number of doors and number of engine cylinders. Also suppose that the form did not contain a place for any additional information. The form worked well for a time, until a new feature such as Global Positioning System (GPS) capability was introduced. Since the form had no place to indicate if a car had GPS capability, it was not "extensible." If the form contained a place for additional information, the GPS information could be added to the form, along with other new features. The form would then be extensible.

A registry that supports metadata extensibility provides a simple series of menu choices in the registry user interface that allows submission of a new metadata attribute. The attribute could be captured and maintained in the registry. A registry that does not support metadata extensibility has a set of fixed metadata attributes for a registered object, and it cannot be expanded without a programmatic update to the registry software. It would be impossible for users of such a registry to capture a new metadata attribute for a registered object (e.g., the name of the authoring tool for an XML schema) without a programmatic update.

The type of extensibility that requires changes to the registry configuration is known as configuration-time metadata extensibility. A registry that allows submitters to define their own metadata attributes along with submissions is known as submission-time metadata extensibility. The captured metadata attributes are recorded in the registry for pertinent registered objects.

Object Type Extensibility

Continuing the example of the 1970s form, suppose a form was created for several types of vehicles (e.g., cars, trucks, and vans). In time, a new type of vehicle—an SUV—is introduced. With no “SUV” form, its impossible to describe the features of this new vehicle type because the entities on the forms were fixed. If the form had a general design, it could describe new vehicles as they appear.

A registry that supports object type extensibility uses a series of simple menu choices in the registry user interface that allows new object types to be recognized[6] in the registry. A registry that does not support object type extensibility cannot recognize fixed objects, and it cannot be expanded without requiring a registry software programmatic update. Without such an update, users could not designate that the registry should recognize a new type of object.

This type of extensibility is configuration-time object type extensibility because it requires changes to the registry configuration. A registry can also have submission-time object type extensibility that allows submission of an object of any type, whether that object type is explicitly recognized by the registry.

[6] A user could request to access this type of object in a query, and the registry would accommodate the request because it can “recognize” objects as being of that type.

PROPOSED REQUIREMENTS

Table 5-1 lists the proposed requirements for XML registry metadata and contents.

Table 5-1. Proposed Requirements for XML Registry Metadata and Contents

ID no.	Topic	Proposed requirement	Justification
5.1	Registry metadata	Registry must at a minimum maintain the following metadata attributes: • UUID • Object name • Object type • Description • Version • URL • Object status • Validation status • Validation tool • Authorative data source • Security classification[g] • Distribution statement[h]	The list of minimum metadata attributes provides information to identify, define, reference, and maintain an object. Security classification and distribution statement are necessary to identify objects with restricted access.
5.2	XML objects	Registry must support storage of the following types of XML objects: • DON XML schemas • DON DTDs • DON XML documents • DON style sheets • DON XML complex elements • DON XML simple elements • DON XML attributes • Partner XML schemas • Partner DTDs • Partner style sheets • Partner XML complex elements • Partner XML simple elements • Partner XML attributes Registry may support storage of the following types of XML objects: • Partner XML documents	Schemas, DTDs, and XML constructs must be stored to support development-time access and run-time validations. XML documents must be stored to allow the discovery of content such as policies and standards. Style sheets must be stored to support implementers and users who need the stored style sheet to render registered content.
5.3	Non-XML objects	Registry must at a minimum support storage of the following types of non-XML objects: • Supporting documentation • URLs[i] • URIs • URNs	Storage of non-XML objects allows registration of supporting documentation for registry submissions.

Table 5-1. Proposed Requirements for XML Registry Metadata and Contents

ID no.	Topic	Proposed requirement	Justification
5.4	Core components	Registry must support storage of core components.[j]	Core components are important for assisting developers in producing interoperable objects. The ebXML Core Components Technical Specification has become an accepted standard for standardizing business entities.
5.5	Business processes	Registry must support storage of business processes and UBL components such as standard formats for common business documents (e.g., invoices, purchase orders, and advance shipment notices).	Business processes will standardize multi-organizational business activities such as invoicing.
5.6	Trading partner profiles and agreements	Registry must support the storage of TPPs and TPAs.	TPPs are needed to provide for the discovery of DON trading partner capabilities. TPAs will assist in discovery of trading partner relationships.
5.7	Web Services	Registry must support the registration of Web services.	A number of survey respondents to this document listed the support of Web services as one of, if not the, most important functions of an XML registry.
5.8	Metadata extensibility	Registry must support configuration-time metadata extensibility. Registry may support submission-time metadata extensibility.	For cost and time efficiency, the registry administrator must be able to expand the metadata attributes through quick configuration changes. The ability of submitters to expand metadata attributes may be unwieldy.
5.9	Object type extensibility	Registry must support configuration-time object type extensibility. Registry may support submission-time object type extensibility.	For cost and time efficiency, the registry administrator must be capable of expanding the list of object types. It may be desirable for submitters to be able to expand object types, but that capability would need to be checked against the registry administration.
5.10	User roles extensibility	Registry must support user roles extensibility.	This support will promote widespread usage among subscribers.

[g] The minimum security classification for accessing the object (e.g., “classified”).

[h] For example, “NATO only.”

[i] Perhaps a website that contains information about a registered object.

[j] The storage of core components will need to be planned for a later release of a registry because the mapping of CCTS registration rules to the ebXML Registry Information Model has just begun in the OASIS ebXML Registry Technical Committee.

Chapter 6
Organization of XML Registry Contents

This chapter addresses the following topics on the organization of contents in the XML registry:

- *Associations*—Relationships between registered objects.
- *Taxonomies*—Features that allow registered objects to be grouped by common characteristics.
- *Namespaces*—Categories and collections where registered objects are placed.

ASSOCIATIONS

It is often necessary to link registered objects in a registry. This linkage makes it easier for users to access related objects or processes that automatically access registry contents. Linking also can associate objects with each other, as the following examples illustrate:

- An association can exist between an XML document and its corresponding XML schema, signifying that the XML document validates to the schema.
- A supporting document can be associated with an XML schema such as a PDF document that describes the schema contents in detail.

An association in a registry may contain a role that describes the function of that association. Sometimes it is insufficient to stipulate that two or more objects are associated; rather, it is more explicit to say how they are associated. Listed below are some examples of associations:

- *Is validated by*—An XML document is validated by an XML schema that resides in the registry.
- *Is related to*—A stipulation that two or more objects are related in some way.
- *Is qualified by*—An XML element stored in the registry as a registered object is “qualified by” an attribute stored in the registry.
- *Replaces*—A registered object replaces an older version in the registry.

Cardinality defines the possible quantifiable relationships between occurrences of entities on either end of the relationship line. For instance, one-to-one or one-to-many.

Associations in a registry may be strictly one-to-one (i.e., only two objects are associated), one-to-many, or many-to-many. For example, it may be necessary to stipulate that a given registered object "is related to" many registered objects, and that registered objects can be involved in several similar relationships. In this case, the associations would be many-to-many.

TAXONOMIES

A taxonomy is an arrangement or division of objects into groups based on common characteristics, such as origin, composition, structure, application, or function. A taxonomy depends on a pre-existing specification of a hierarchy of values, names, and codes called a "taxonomy scheme." Two examples of a taxonomy scheme are the five-level hierarchy North American Industry Classification System (NAICS) and the seven-level binomial nomenclature taxonomy used by biologists to classify living things (kingdom, phylum, class, and so forth). Taxonomy in a registry may be represented as a reference to a single node of a taxonomy scheme (e.g., NAICS code "11114" represents the "wheat farming" industry).

Three additional topics related to taxonomy follow:

- Multiple taxonomies per registered object
- Context-sensitive taxonomy
- External taxonomy.

Multiple Taxonomies per Registered Object

A registered object can be classified within several taxonomy schemes in a registry. For example, a registry may contain a taxonomy scheme that represents its organizational structure and one that is more subject-oriented. In this registry, a registered object can be classified by organizational affiliation and subject, which allows the registered object to be discovered by both the organizational department it is affiliated with and the subject the registered object relates to.

Context-Sensitive Taxonomy

Taxonomy schemes themselves can be classified in a registry, creating a "taxonomy of taxonomies." In the preceding example, a subject-oriented taxonomy could be classified according to an organizational structure, meaning a given node of the subject-oriented taxonomy scheme indicates not only a subject but also an organizational department. This feature means a registered object could be associated with a single node of the subject-oriented taxonomy scheme and be classified by both schemes, rather than being associated with two nodes (as with the “multiple taxonomies per registered object” scenario).

External Taxonomy

It is also possible for a registry to allow objects to be classified within a taxonomy scheme with nodes residing outside the registry. This technique specifies an identifier for the node of the external taxonomy scheme linked with the registered object and a reference to the taxonomy scheme by UID or URL. This approach may be useful when the taxonomy scheme is large (e.g., NAICS). Because external taxonomy schemes do not reside in the registry, fewer taxonomy schemes need to be maintained and updates to the taxonomy scheme automatically take effect in the registry (e.g., a node associated with a registered object moves to a different “place” within the taxonomy scheme tree as necessary).

NAMESPACES

This section describes two types of namespaces:

- *Functional namespaces*
- *XML namespaces.*

Functional Namespaces

Registered objects can be categorized in a registry according to their high-level usage such as the function an object serves. The type of namespace that performs this type of categorization is known as a functional namespace. For example, the DoD XML registry contains the following functional namespaces (known as enterprise namespaces):

- Acquisition logistics
- Cryptologic
- Combat support
- Finance and accounting
- Supply.

As an example, a registered object related to an agency’s finance function would be registered under the finance and accounting namespace. Functional namespaces can be used to “disambiguate” like-named objects. They also allow two different XML schemas to have the same name and even the same version number, but to exist in two different functional namespaces.

XML Namespaces

The Namespaces in XML Specification[1] defines XML namespaces as "a collection of names identified by a URI reference used in XML documents as element types and attribute names." XML namespaces are used in XML schemas to associate schema constructs with a "conceptual space" that defines a markup vocabulary. Namespaces are declared in the root element of an XML schema and an XML document using a URI[2] known as a "namespace identifier." XML constructs within a schema can be associated with a given namespace by associating them with the namespace identifier.

Just as functional namespaces can be used to disambiguate like-named objects, XML namespaces can be used to disambiguate like-named constructs. That is, they allow two different constructs (such as two elements of a different data type) to have the same name but exist in two different namespaces. This feature allows both elements to be used in the same XML schema or XML document without any processing conflicts.

If a registry is "XML namespace-aware," it may be possible to associate XML constructs with the namespace where they are assigned. When this occurs, other possibilities arise, including the capability to

- query all constructs in a given namespace,
- transfer constructs between one namespace and another, and
- discover the namespace where a construct belongs.

Relationship Between XML and Functional Namespaces

An XML namespace identifier may correspond to a functional namespace. For example, a DON functional area called "medical" could have the namespace identifier "www.don.mil/functionalareas/medical."[3] In an "XML namespace-aware" registry, it would be possible to register and associate medical specificXML constructs in an XML schema with a target namespace identifier shown in this "medical" functional namespace.

Functional namespaces provide an organization of registry contents (e.g., a transport functional namespace oversees schemas dealing with shipping). But in addition to namespaces for specific functional areas an enterprise namespace would be an area in the registry to collect and manage objects applying across the DON functional areas.

[1] http://www.w3.org/TR/REC-xml-names.

[2] URI is a general term for a construct that identifies a resource on the Web. A URI can be either a URL (Uniform Resource Locator) or URN (Uniform Resource Name).

[3] The URL www.don.mil is fictional and intended only to represent a unique identifier.

Technical namespaces refer to namespaces in XML schema and documents to distinguish elements and attributes designated to a namespace that may be a part of the schema. For example, a schema could have an American namespace and a British namespace, where the element “snaps” can appear more than once in the same schema but mean different things when associated with the American and British namespaces (e.g., fasteners vs. photographs). If a DON enterprise namespace referred to technical namespaces, it would imply the DON could establish elements and attributes that would have DON-specific meanings when applied to a DON technical namespace.

PROPOSED REQUIREMENTS

Table 6-1 lists the proposed requirements for the contents of an XML registry.

Table 6-1. Proposed Requirements for Organization of XML Registry Contents

ID no.	Topic	Proposed requirement	Justification
6.1	Associations	Registry *must* support use of associations and address the issue of cardinality. Registry must allow user to traverse associations.	Linking content to constructs supports run-time validations and makes it clear to developers when objects have an established relationship.
6.2	Taxonomies	Registry *must* support use of taxonomies. Registry *shall* support multiple taxonomies per registered object. Registry *may* support context-sensitive taxonomies. Registry *may* support external taxonomies.	Registry support for taxonomies are necessary to help organize the contents of the registry for efficient discovery. The cross-section of users for some objects makes supporting assignment of multiple taxonomies per object a good idea. Reducing duplication of externally maintained taxonomies improves content accuracy.

Table 6-1. Proposed Requirements for Organization of XML Registry Contents (Continued)

ID no.	Topic	Proposed requirement	Justification
6.3	Namespaces	Registry *must* support functional namespaces. Registry *must* support a DON enterprise namespace. Registry *must* support the ability to identify duplicate entries regardless of namespaces. Registry *must* be XML namespace-aware, which would make it possible to register and associate all XML constructs in an XML schema whose target namespace was the namespace identifier associated with the XML functional namespace. Registry *must* support the management of an enterprise functional area. Registry *must* support management of the functional namespace coordinator's functional areas. Registry *should* support namespaces for generic business functions that may encompass one or more functional namespace coordinators.	DON procedures calls for functional namespace coordinators to manage the development of XML relevant to their area. Entries will be associated with DON functional namespaces. The DON will coordinate management of objects at the enterprise level as they progress up from functional namespace reviews to become an enterprise standard. The DON enterprise namespace will seek to harmonize objects registered with other namespace managers. Development-time likely will make use of the registry's capability of being namespace aware during validation of referenced objects.

Chapter 7
XML Registry Interoperability/Compliance

This chapter addresses several topics related to XML registry interoperability and compliance:

- *Open registry standards*—Ensuring that the XML registry is based on an open registry standard.
- *Existing registries*—Ensuring that the proper level of interoperability exists between the XML registry and other registries where interaction is desirable or necessary.
- *Federal and DoD mandates*—Ensuring that the XML registry is compliant with federal and DoD mandates.

OPEN REGISTRY STANDARDS

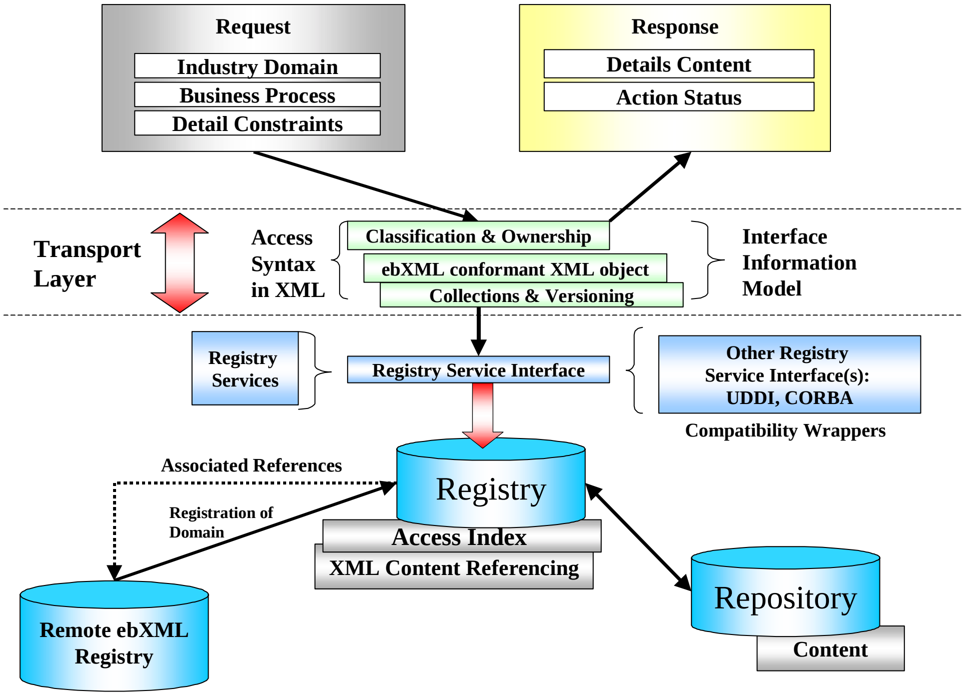
In compliance with Public Law 104-113 and Office of Management and Budget Circular A-119, the DON must use a COTS product for its registry software. This section describes some of the existing open registry standards.

OASIS/ebXML Registry

The products of a joint venture between OASIS and UN/CEFACT included XML registry standards called the ebXML Registry Specification and the Registry Information Model. The ebXML registry stores information submitted by organizations to facilitate ebXML-based business-to-business partnerships and transactions. The submitted content may be XML schema and documents, web services, process descriptions, Unified Modeling Language models, information about parties, and software components.

Figure 7-1 illustrates the ebXML registry architecture as it appeared in the ebXML Technical Architecture Specification v1.0.4, February 2001.

Figure 7-1. ebXML Registry Architecture



Source: ebXML.

As this figure shows, the ebXML registry interacts with a local repository and a remote ebXML registry. Requests are sent to the registry, and responses are received from the registry through a registry service interface. In the future, the registry service interface also may interact with other registry service interfaces such as the Universal Description, Discovery, and Integration (UDDI) and open interface standards such as the Common Object Request Broker Architecture (CORBA).

Universal Description, Discovery, and Integration

The UDDI initiative, which began as collaboration between IBM, Ariba, and Microsoft, seeks to help large organizations manage their network of smaller business customers through a shared operation of a business registry on the Web. The UDDI business registry is operated as a distributed service. Currently Microsoft and SAP operate registry nodes. As of July 2002, the UDDI specification is maintained by the UDDI Technical Committee under OASIS.

EXISTING REGISTRIES

The XML registry may require interoperability with several other XML registries, and one registry submission process is needed for all registries. Interoperability may be required on one or both of the following levels:

- *Metadata interoperability*—Ensuring that the XML registry retains sufficient metadata to allow at least a manual upload[1] of a registered object into another registry.
- *Run-Time interoperability*—Ensuring that registered objects can be automatically uploaded to another registry through interaction with the XML registry and automatically downloaded from another registry. Such interoperability requires metadata interoperability.

The XML registry may also require interoperability with the following entities:

- DoD XML registry
- Intelligence community registry
- Navy enterprise portal service registry
- XML.org registry
- DON applications and database management system
- Task Force Web (TFW) services registry.

DoD XML Registry

The DoD XML registry is used by the DON for registering its XML objects and non-XML objects. “DoD Policy for Registration of Extensible Markup Language (XML),” April 2002, established the registry to provide guidance in the generation and use of XML among DoD communities of interest and to be an authoritative source for registered XML data and metadata. The policy states:

> To support interoperability and minimize overhead, the Department is establishing a single clearinghouse and registry for creating, finding, reusing, and unambiguously identifying XML components. It will promote interoperability, efficiency, and reuse of XML components…The DOD XML Registry and Clearinghouse is the authoritative source for XML components. All Program Managers (PMs) that use XML as an interchange format must register XML components in accordance with procedures established by DISA.

The Defense Information Systems Agency (DISA) hosts the registry. Some objects that can be stored in the registry include DTDs, XML schemas, XML

[1] This “manual upload” may be a non-real-time batch process that executes daily, or it may be a manual user interaction with another registry.

documents, and XML elements, attributes, and data types.[2] The registry is organized by enterprise namespaces, including the following:

- Acquisition logistics
- Cryptologic
- Combat support
- Finance and accounting
- Supply.

New objects are submitted in a package with supporting information to one of the namespaces, and a namespace manager oversees the progression of each submitted object through its life cycle.

Intelligence Community Registry

The U.S. Intelligence Community (IC) Metadata Working Group is developing an XML registry. In late 2002 the working group identified a set of requirements for the registry. Those requirements are now incorporated into the DoD registry and the IC intends to maintain a copy of those requirements for its users.

Navy Enterprise Portal Service Registry

The Navy enterprise portal (NEP) contains several components including a service registry. The service registry is a private, globally distributed registry of web application and services information. All web applications and web services are required to provide metadata for the registry. The NEP service registry has been implemented as a UDDI registry.

XML.org Registry

XML.org is an open, vendor-neutral website for XML resources hosted by OASIS. It contains a registry that is a central clearinghouse for developers and standards bodies to submit, publish, and exchange XML schemas, vocabularies, and related documents.

DON Applications and Database Management System

The DON Applications & Database Management System (DADMS), formerly the Data Management & Interoperability Repository, is a CADM-based, web-enabled registry of systems and applications and their associated data structures and data exchange formats. It contains the DON data architecture and supports systems interoperability, application rationalization, and IT assessments. DADMS has

[2] This means element and attribute declarations and data type definitions.

been expanded to hold metadata, models, any attachment, conductivity/ connection information, POC information, links to functional area taxonomies, functional area data requirements architecture, message format information and more for systems, applications, and databases. DADMS will be used by functional area managers and functional data managers for application portfolio management, NMCI seat profiles, functional area data management, and more.

Task Force Web Registry

Task Force Web is responsible for supporting efforts to web-enable DON applications. As part of that effort, it is developing a web services registry. Web services registration and discovery involving a DON XML registry needs to be coordinated with the Task Force Web.

FEDERAL AND DoD MANDATES

Two federal mandates (the Clinger-Cohen Act of 1996 and Section 508 of the Rehabilitation Act Amendments of 1998), and two DoD mandates ("Policy Guidance for Use of Mobile Code Technologies" and common access card) affect the XML registry and its interoperability.

Federal Mandates

CLINGER-COHEN ACT

The Clinger-Cohen Act of 1996 encourages individual agencies to improve the cost effectiveness of their IT strategies by using incremental implementations and avoiding custom solutions where possible. To provide accountability, agency heads and chief information officers are required to implement policies for managing and measuring IT assets.

NATIONAL TECHNOLOGY TRANSFER ACT OF 1995

The National Technology Transfer Act of 1995 established a preference for the Federal government to adopt voluntary consensus standards over government unique solutions. Exceptions are made for explicit circumstances under law and where adoptions of standards is impractical. The principals of the Act are represented and clarified in OMB Circular A-119.

SECTION 508 OF THE REHABILITATION ACT

Section 508 of the Rehabilitation Act Amendments of 1998 (29 U.S.C. 794d) requires federal agencies that develop, procure, maintain, or use electronic and information technology (EIT) to ensure that federal employees and members of the public with disabilities have access to and use of information that is comparable to that provided to individuals without disabilities, unless it would pose an undue burden on the agency. These standards became effective December 21, 2000, and they apply to all EIT procured after June 25, 2001. The goal of Section 508 is to provide an environment where individuals with disabilities have the ability to independently access and use EIT.

DoD Mandates

MOBILE CODE

In November 2000, DoD issued "Policy Guidance for Use of Mobile Code Technologies in DoD Information Systems" to address the risk of executing mobile code on critical systems and in environments that contain sensitive information. The policy defines mobile code as "software obtained from remote systems outside the enclave boundary, transferred across a network, and then downloaded and executed on a local system without explicit installation or execution by the recipient." It further defines three categories of mobile code technologies and DoD policy toward mitigating the risks of technologies that fall under each of the categories.

COMMON ACCESS CARD

In November 1999, DoD directed the military services to adopt a single smart card architecture. The smart card, referred to as a common access card, could identify and authenticate the holder in various applications. The smart card contains the user's digital certificate and private key based on the DoD PKI.

PROPOSED REQUIREMENTS

Table 7-1 lists proposed requirements for topics in this chapter.

Table 7-1. Proposed Requirements of XML Registry Interoperability and Compliance

ID no.	Topic	Proposed requirement	Justification
7.1	Open registry standards	Registry *must* use DON-approved registry standards.	Using open standards provides the best mechanism for supporting interoperable implementations and widespread industry support for many XML standards and meets with requirements related to Clinger-Cohen, OMB Circular A-119, and the National Technology and Transfer Act of 1995. Some of the key DON approved standards are W3C Schema Specification, ebXML Registry Specification, and ebXML Registry Information Model. (See the DON approved list of XML standards on NKO for the DON XML community.)

Table 7-1. Proposed Requirements of XML Registry Interoperability and Compliance (Continued)

ID no.	Topic	Proposed requirement	Justification
7.2	Existing registries	Registry *must* have the capability to be interoperable with all DoD, federal, and other external XML registries on a metadata level.[k] Registry *may* have the capability to be interoperable with all DoD, federal, or other external XML registries on a run-time level.[l] Registry *shall* be interoperable with the IC registry on a metadata level for the storage of XML components. Registry *shall* be interoperable with the NEP service registry on a metadata level for the storage and discovery of Web services.[m] Registry *shall* be interoperable with the NEP service registry on a run-time level for the storage and discovery of Web services.[n] Registry *shall* be interoperable with DADMS on a metadata level. Registry *may* be interoperable with DADMS on a run-time level. Registry *must* be interoperable with TFWeb Web services registry on a metadata level. Registry *must* be interoperable with TFWeb Web services registry on a run-time level for the storage and discovery of Web services.	Several initiatives within the DoD seek to support interoperability between C4I systems. To the extent that some business processes can be aligned with those developed in federal agency registries, registry users may be able to leverage activities such as invoicing beyond DoD.
7.3	Federal and DoD mandates	Registry *must* be compliant with Section 508 of the Rehabilitation Act. Registry *must* be compliant with the DoD Mobile Code Policy. Registry *must* be Naval/Marine Corps Intranet (NMCI) certified and Network Security certified.	Refer to each policy for applicable compliance requirements.
7.4	One registration process	Registry submission process *must* be a single process good for all DoD and federal registries.	Promotes the “once and done” submission process for users.

[k] Objects submitted to the XML registry will be uploaded to the DoD, federal, or other external XML registries to avoid double submissions.

[l] Objects submitted to the XML registry may be automatically uploaded to the DoD, federal, or other external XML registries instead of being manually or batch uploaded.

[m] Web services submitted to the XML registry may be manually or batch uploaded to the NEP service registry.

[n] Web services submitted to the XML registry may be automatically uploaded to the NEP service registry.

Appendix
Summary of Proposed Requirements

This appendix contains a summary of proposed requirements from all chapters.

Table A-1. Proposed Requirements

ID no.	Topic	Proposed requirement	Justification
2.1	Networks	Unclassified registry entries and registered objects *must* be duplicated to the classified registries. Unclassified registry entries and registered objects submitted to a classified registry *must* be verified as unclassified before they can be synchronized with an unclassified registry. Classified objects submitted to an unclassified registry *must* be rejected. Registry *shall* provide access for external consultants and trading partners. Registry *should* evaluate portals to the CWAN and JWICS networks.	NIPRNET and SIPRNET are likely environments for hosting registries, particularly for registration of web services. [a] It is assumed that registries on NIPRNET and SIPRNET should cover the widest possible group of users; however, DON participants in the U.S. intelligence community, served by the CWAN and JWICS networks, are participating in an effort to establish an XML registry for the intelligence community. Synchronizing unclassified objects between the two environments introduces additional complexity and risks to isolate classified objects in SIPRNET. However, the ability to submit and administer objects in one area is preferred to managing objects in different registries. Submissions to a classified registry are assumed to infer classified status unless explicitly designated by the submitter to reduce the risk of classified objects being ported to an unclassified registry during synchronization. The development and testing of XML-based applications may involve contractors operating off-site. Trading partners will need access to the registry to discover DON-approved standards for conducting business and to register their capabilities.

Table A-1. Proposed Requirements (Continued)

ID no.	Topic	Proposed requirement	Justification
2.2	Performance issues	Registry requirements unique to shipboard platforms *mus*t be accommodated. Registry must be capable of supporting a replicated registry architecture. Registry architecture *must* be scalable.	XML is expected to be introduced as a major technology to support administrative systems and weapon systems. The use and development of XML on afloat systems requires a registry architecture that can take into account the ability to replicate registry content to afloat systems, particularly during the long periods of deployment between synchronizations. [b] Respondents to a survey of registry requirements indicated that they believe their system environment will require them to operate a localized version of the DON registry. Such support would also help facilitate the disaster recovery and continuous operations requirements discussed in Chapter 3. As more XML applications and services are produced, the demand on the registry will grow.
3.1	User access	Registry must be accessible through DON-approved network protocols.[c] Registry must support an NEP portal.	Human and automated users will look to access registries through various methods such as the Web and networks supported by DON policy.
3.2	Type of access	Registry must provide both development-time and run-time access.	System developers need development-time access to discover existing objects and collaborate on new objects. Automated systems need run-time access to validate documents received against registered schemas.
3.3	Registry users	Any DON agency, authorized trading partner, or authorized contractor *must* be able to access the unclassified XML registry, and the classified XML registry, as appropriate. Any DON agency, authorized trading partner, or authorized contractor *must* be able to submit to the unclassified XML registry, and the classified XML registry, as appropriate.	DON agencies will access the registry to support their systems. Authorized trading partners will access the registry to discover objects that support their transactions and maintain their profiles. Authorized contractors will access the registry to support development and maintenance of DON systems.

Table A-1. Proposed Requirements (Continued)

ID no.	Topic	Proposed requirement	Justification
3.4	Search Capability	Registry *must* allow for key word searches. Registry *must* allow for Boolean defined searches Registry *must* allow for integrated searches with federated registries. Registry *must* allow for searches based on functional namespace. Registry *must* allow searches based on object status. Registry *must* allow for searches of a functional namespace filtered for object status Registry *shall* provide for the identification of potential duplicate object names or description	Key word and Boolean searches will provide users with a minimum functionality to focus their searches for relevant components. The DON is promoting extending interoperability up through DoD, federal, and standards groups. Connecting searches to registries established by those groups will improve the DON's ability to identify and adopt relevant external components. At a minimum, namespace managers require the ability to identify objects within their namespace at each of the life-cycle statuses. General users will also benefit from this capability.
3.5	Submission validation	Registry must support checking well-formedness and validity of submissions when submitted to the registry. User must be able to turn on or off based on the type of submission. Registry must maintain sufficient metadata to indicate validation status.[d]	To be a relevant registry, approved objects must demonstrate that they are well formed and valid. Such checks can be performed outside the registry, but a more effective method would be to give users the option to execute checks when submitting. Because tools to check well-formedness and validation are inconsistent, the registry must allow submitters to bypass automated checks with the understanding that the submitter will provide external proof. Subscribers to objects need the registry to provide adequate metadata to express if an object has passed well-formedness and validity checks.
3.6	Messaging protocols	Registry must support SOAP. Registry must support ebMS.	SOAP is the widely accepted mechanism for XML transactions expected from automated systems. ebMS expands on SOAP to provide additional security capabilities.
3.7	Publish/subscribe	Registry *must* contain publish/subscribe functionality. Registry *must* allow for AISs to subscribe to objects.	To support many interoperable systems, users need the capability to receive automatic registry notification of modifications that could affect their implementations. Since individuals frequently change locations and positions, it is important to be able to identify the AISs that are potentially impacted independent of original users.

Table A-1. Proposed Requirements (Continued)

ID no.	Topic	Proposed requirement	Justification
3.8	Intent to develop	Registry *must* contain a mechanism for XML developers to declare their intent to develop new XML constructs and define points of contact. Registry *must* provide for a virtual workspace to support collaborative development efforts	To support interoperability, users need to be involved with important development projects as early as possible. Retrofitting objects after implementation can be logistically difficult and expensive.
3.9	Object construction	Registry must be capable of auto-generating a schema from a developer's selections of registered components. Registry should be capable of auto-generating other XML objects from modular components.	To better facilitate the reuse of existing components within the registry, the DON is planning on the registry to be able to construct XML objects by aggregating registered components.
3.10	Registered object life cycle	Registry *must* support life-cycle functionality for registered objects. Registry *shall* support at a minimum the following life-cycle statuses: non-standard, development, submitted, rejected, approved, and deprecated. Registry shall provide for the recording of information related to an object's review by the DON.	To keep implementations up to date, the registry must make it clear to users when an object is in development, approved for implementation, or obsolete. The minimum status types are required to support BSC Operating Procedures.
3.11	Logging/audit trail	Registry *must* contain adequately robust logging and audit trail functionality that includes at a minimum: - User ID - Operation performed - Date and time - Object UUID. Registry *should* include "standard" audit trail reports. [e] Registry *may* include user-defined audit trail reports.	To make properly informed decisions and follow-up on issues, users reviewing submissions need to know the history of changes to the object and who made them.
3.12	Disaster recovery and continuity of operations	Registry must implement disaster recovery capabilities. Registry must operate with a continuity of operations plan.	For the registry to support critical systems, a disaster recovery plan and a continuity of operations plan must be established.
3.13	Intent to develop	Registry must contain a mechanism for XML developers to declare their intent to develop new XML constructs and define points of contact. Registry must provide for a virtual workspace to support collaborative development efforts.	To support interoperability, users need to be involved with important development projects as early as possible. Retrofitting objects after implementation can be logistically difficult and expensive.

Table A-1. Proposed Requirements (Continued)

ID no.	Topic	Proposed requirement	Justification
4.1	Open XML-related security standards	Registry must use DON-approved open XML-related security standards.	A 2001Defense Authorization Act subsection on government information security reform directs the DoD to use NIST-specified security policies at a minimum. NIST security policies are based on open standards.
4.2	Changes in security and user access requirements	Registry *must* rapidly accommodate changing conditions in security requirements. The registry *must* be capable of restricting levels of access on demand.	INFOCONS details responses to threats posed to DoD information systems.
4.3	User authentication	Registry must employ user authentication mechanisms to ensure the identity of the individual.	To verify user rights granted under an account, the registry must authenticate the identity of all users. PKI is the preferred identification and authentication method, but user ID and passwords can also be used for this activity.
4.4	Non-repudiation	Registry must use non-repudiation mechanisms to ensure that repudiation of registry submissions does not occur.	To ensure the registry properly captures an action by a user, such as establishing trading partner agreements, the system must be able to prove under audit that the action was properly recorded and executed by the appropriate user.
4.5	Authorization	Registry must use role-based and organization-based access control policies to ensure the proper level of access to registry content is granted according to DON's security needs.	Secured systems often use access constraints based on organization (e.g., SIPRNET requires MIL domains); only individuals with a particular clearance are given access within organizations.
		Registry must support access control at the object level.	Registry needs to limit access to certain objects by designating a subset of authorized users for security and control of early developmental projects.
4.6	Message integrity	Registry must use message integrity mechanisms to ensure that registry submissions have not been tampered with en route to the registry.	Content data submissions cannot be subject to changes in transit.
4.7	Confidentiality	Registry must provide confidentiality mechanisms during data transfer to ensure that transferred content is viewable only by authorized parties.	No unclassified content document can be allowed to route to or from the directory along an unencrypted channel.
		Registry may use confidentiality mechanisms for stored content to ensure it is viewable only by authorized parties.[f]	Because the registry will protect against unauthorized access, restricted objects may not need encrypting inside the registry; such functionality may be desired in certain circumstances.
		Registry *must* support DoD's PKI infrastructure.	DoD policy requires the use of PKI to support the common access card architecture.
4.8	Ownership of content	Registry must use ownership data for all components.	Ownership data is necessary for configuration management of changes and publish/subscribe capability.

Table A-1. Proposed Requirements (Continued)

ID no.	Topic	Proposed requirement	Justification
5.1	Registry metadata	Registry must at a minimum maintain the following metadata attributes: • UUID • Object name • Object type • Description • Version • URL • Object status • Validation status • Validation tool • Authorative data source • Security classification[g] • Distribution statement[h]	The list of minimum metadata attributes provides information to identify, define, reference, and maintain an object. Security classification and distribution statement are necessary to identify objects with restricted access.
5.2	XML objects	Registry must support storage of the following types of XML objects: • DON XML schemas • DON DTDs • DON XML documents • DON style sheets • DON XML complex elements • DON XML simple elements • DON XML attributes • Partner XML schemas • Partner DTDs • Partner style sheets • Partner XML complex elements • Partner XML simple elements • Partner XML attributes Registry may support storage of the following types of XML objects: • Partner XML documents.	Schemas, DTDs, and XML constructs must be stored to support development-time access and run-time validations. XML documents must be stored to allow the discovery of content such as policies and standards. Style sheets must be stored to support implementers and users who need the stored style sheet to render registered content.
5.3	Non-XML objects	Registry must at a minimum support storage of the following types of non-XML objects: • Supporting documentation • URLs[i] • URIs • URNs.	Storage of non-XML objects allows registration of supporting documentation for registry submissions.

Table A-1. Proposed Requirements (Continued)

ID no.	Topic	Proposed requirement	Justification
5.4	Core components	Registry must support storage of core components.[j]	Core components are important for assisting developers in producing interoperable objects. The ebXML Core Components Technical Specification has become an accepted standard for standardizing business entities.
5.5	Business processes	Registry must support storage of business processes and UBL components such as standard formats for common business documents (e.g., invoices, purchase orders, and advance shipment notices).	Business processes will standardize multi-organizational business activities such as invoicing.
5.6	Trading partner profiles and agreements	Registry must support the storage of TPPs and TPAs.	TPPs are needed to provide for the discovery of DON trading partner capabilities. TPAs will assist in discovery of trading partner relationships.
5.7	Web services	Registry must support the registration of Web services.	A number of survey respondents to this document listed the support of Web services as one of, if not the, most important functions of an XML registry.
5.8	Metadata extensibility	Registry must support configuration-time metadata extensibility. Registry may support submission-time metadata extensibility.	For cost and time efficiency, the registry administrator must be able to expand the metadata attributes through quick configuration changes. The ability of submitters to expand metadata attributes may be unwieldy.
5.9	Object type extensibility	Registry must support configuration-time object type extensibility. Registry may support submission-time object type extensibility.	For cost and time efficiency the registry administrator must be capable of expanding the list of object types. It may be desirable for submitters to be able to expand object types, but that capability would need to be checked against the registry administration.
5.10	User roles extensibility	Registry must support user roles extensibility.	This support will promote widespread usage among subscribers.
6.1	Associations	Registry must support use of associations and address the issue of cardinality. Registry must allow user to traverse associations.	Linking content to constructs supports run-time validations and makes it clear to developers when objects have an established relationship.
6.2	Taxonomies	Registry must support use of taxonomies. Registry shall support multiple taxonomies per registered object. Registry may support context-sensitive taxonomies. Registry may support external taxonomies.	Registry support for taxonomies are necessary to help organize the contents of the registry for efficient discovery. The cross-section of users for some objects makes supporting assignment of multiple taxonomies per object a good idea. Reducing duplication of externally maintained taxonomies improves content accuracy.

Table A-1. Proposed Requirements (Continued)

ID no.	Topic	Proposed requirement	Justification
6.3	Namespaces	Registry must support functional namespaces. Registry must support a DON enterprise namespace. Registry must support the ability to identify duplicate entries regardless of namespaces. Registry *must* be XML namespace-aware, which would make it possible to register and associate all XML constructs in an XML schema whose target namespace was the namespace identifier associated with the XML functional namespace. Registry must support the management of an enterprise functional area. Registry must support management of the functional namespace coordinator's functional areas. Registry should support namespaces for generic business functions that may encompass one or more functional namespace coordinators.	DON procedures calls for functional namespace coordinators to manage the development of XML relevant to their area. Entries will be associated with DON functional namespaces. The DON will coordinate management of objects at the enterprise level as they progress up from functional namespace reviews to become an enterprise standard. The DON enterprise namespace will seek to harmonize objects registered with other namespace managers. Development-time likely will make use of the registry's capability of being namespace aware during validation of referenced objects.
7.1	Open registry standards	Registry must use DON-approved registry standards.	Using open standards provides the best mechanism for supporting interoperable implementations and widespread industry support for many XML standards and meets with requirements related to Clinger-Cohen, OMB Circular A-119, and the National Technology and Transfer Act of 1995. Some of the key DON approved standards are W3C Schema Specification, ebXML Registry Specification, and ebXML Registry Information Model. (See the DON approved list of XML standards on NKO for the DON XML community.)

Table A-1. Proposed Requirements (Continued)

ID no.	Topic	Proposed requirement	Justification
7.2	Existing registries	Registry must have the capability to be interoperable with all DoD, federal, and other external XML registries on a metadata level.[k] Registry may have the capability to be interoperable with all DoD, federal, or other external XML registries on a run-time level.[ll] Registry *shall* be interoperable with the IC registry on a metadata level for the storage of XML components. Registry shall be interoperable with the NEP service registry on a metadata level for the storage and discovery of Web services.[m] Registry shall be interoperable with the NEP service registry on a run-time level for the storage and discovery of Web services.[n] Registry shall be interoperable with DADMS on a metadata level. Registry may be interoperable with DADMS on a run-time level. Registry *must* be interoperable with TFWeb Web services registry on a metadata level. Registry must be interoperable with TFWeb Web services registry on a run-time level for the storage and discovery of Web services.	Several initiatives within the DoD seek to support interoperability between C4I systems. To the extent that some business processes can be aligned with those developed in federal agency registries, registry users may be able to leverage activities such as invoicing beyond DoD.
7.3	Federal and DoD mandates	Registry must be compliant with Section 508 of the Rehabilitation Act. Registry must be compliant with the DoD Mobile Code Policy. Registry must be Naval/Marine Corps Intranet (NMCI) certified and Network Security certified.	Refer to each policy for applicable compliance requirements.

Table A-1. Proposed Requirements (Continued)

ID no.	Topic	Proposed requirement	Justification
7.4	One registration process	Registry submission process *must* be a single process good for all DoD and federal registries	Promotes the "once and done" submission process for users.

[a] Based on Chief of Naval Operations (CNO) memo, Request for Implementation of Joint-Allied Web Services Interoperability, May 22, 2002.

[b] Afloat synchronization is discussed further in the DON XML Registry CONOPS.

[c] This feature would allow the XML registry to be accessible from any Web-enabled device such as hand-held devices and browsers.

[d] XML registry must maintain metadata such as whether a submission was validated, whether it was validated by the registry or the submitter (or both), and the tools used for validation.

[e] "Standard" means reports included with the registry software, as opposed to user-defined.

[f] With only an SSL certificate on the server, the registry could provide data confidentiality through the encryption features of SSL V3 or TLS; however, to ensure that only authorized parties view registry contents, mutual authentication must be used.

[g] The minimum security classification for accessing the object (e.g., "classified").

[h] For example, "NATO only."

[i] Perhaps a website that contains information about a registered object.

[j] The storage of core components will need to be planned for a later release of a registry because the mapping of CCTS registration rules to the ebXML Registry Information Model has just begun in the OASIS ebXML Registry Technical Committee.

[k] Objects submitted to the XML registry will be uploaded to the DoD, federal, or other external XML registries to avoid double submissions.

[l] Objects submitted to the XML registry may be automatically uploaded to the DoD, federal, or other external XML registries instead of being manually or batch uploaded.

[m] Web services submitted to the XML registry may be manually or batch uploaded to the NEP service registry.

[n] Web services submitted to the XML registry may be automatically uploaded to the NEP service registry.