EPA

Logistics Management Institute

Requirements for an XML Registry

EP005T4

LMI®

Logistics Management Institute

Requirements for an XML Registry

EP005T4

May 2001

Joseph M. Chiusano
Terence J. Schmitt
Mark Crawford

Logistics Management Institute
2000 Corporate Ridge
McLean, VA 22102-7805

Executive Summary

Before the Internet came to fruition, the World Wide Web Consortium (W3C) recognized the need for a standardized language to universally represent structured data (i.e., bold, italics, font size). As a result, W3C created Hypertext Markup Language (HTML) for marking up documents with specific data tags that provide flexible, self-describing text for Web pages.

Although an effective standard, users found HTML to be too constraining for information exchanges. In 1998, the W3C published the Extensible Markup Language (XML). XML is a web-enabled subset of the Standard Generalized Markup Language (SGML). Unlike HTML, which combines data and its presentation requirements, XML separates the two elements. This allows XML-formatted data to be used for different purposes and displayed on different devices (Web browsers, cellular phones, etc.) with minimal additional processing. XML also allows data to be transferred easily between disparate systems.

The boon of XML is its extensibility. The bane of XML is its extensibility. To use XML for information exchanges, standardization of the numerous components, such as those that determine data exchange formats and provide trading partner information, are required. Recording these components in an XML registry enables XML to be used consistently, both in projects and between organizations.

The benefits of an XML registry are numerous:

- Promotes reuse
- Enables efficient version control
- Promotes unified understanding of registered objects
- Ensures consistency across organizational areas
- Promotes selective access to registered objects
- Enables collaborative development.

The U.S. Environmental Protection Agency (EPA) has tasked LMI with determining requirements for an XML registry. As EPA continues with its e-government initiatives, it needs to implement an XML registry to offer a central location for

XML and e-business resources. The registry will enable XML components to be standardized and shared among EPA, states, and industry partners, thus making data more coherent throughout the agency.

We recommend that

- EPA implement an XML registry based on the electronic business XML (ebXML) model for storing document type definitions, Schemas, XML documents, and business information content;
- XML tags used in document type definitions and Schemas be stored in a second registry based on the International Standards Organization and International Electrotechnical Commission 11179 model;
- EPA use its Environmental Data Registry (EDR) as the 11179 registry because the EDR is an implementation of an 11179 registry; and
- the XML registry be accessible from the Environmental Information Exchange Network designed by the State/EPA Information Management Work Group.

We examined two registry models: the ebXML model and the Organization for Structured Information Standards (OASIS) model. We recommend implementing the ebXML model for the following reasons:

- *International backing*: The ebXML project is backed by UN/CEFACT, the United Nations body for trade facilitation and electronic business.
- *Recognition by major standards bodies*: The Accredited Standards Committee X12 has stated that their proposed XML standards will be based on ebXML recommendations.
- *Wide availability of commercial off-the-shelf (COTS) software*: We anticipate that COTS software for automating the ebXML registry model will be widely available later this year. OASIS has not publicly announced making COTS software available for its registry model.
- *Wide scope of functionality*: The ebXML model has much greater functionality, including protocols for business processes, security, and messaging service, than the OASIS model.

In this report we describe a proposed architecture and raise functional and policy issues that EPA must consider when implementing an XML registry. We describe in detail the functionality of both the ebXML and OASIS models so the reader may compare the two.

Contents

Chapter 1 Introduction 1-1

OBJECTIVE 1-1

REQUIREMENTS 1-2

REPORT ORGANIZATION 1-3

Chapter 2 XML Registry—General Concept 2-1

WHAT IS AN XML REGISTRY? 2-1

WHAT ARE THE BENEFITS OF AN XML REGISTRY? 2-2

Chapter 3 Models and Standards 3-1

ISO/IEC 11179 3-1

EBXML 3-2

OASIS 3-3

NETWORK BLUEPRINT 3-4

OTHER INITIATIVES 3-5

Federal CIO Council 3-5

XML.gov 3-5

OTHER REGISTRY MODELS 3-6

UDDI 3-6

BizTalk 3-6

XML.org 3-6

Chapter 4 Proposed Architecture 4-1

MULTIPLE CONTEXTS 4-1

ARCHITECTURE ISSUES 4-2

Centralized Versus Distributed 4-2

Static Versus Dynamic 4-2

RECOMMENDATIONS 4-2

Chapter 5 Information Models 5-1
Registered Objects and Metadata 5-1
Terminology 5-2
Registry Information Models 5-2
11179 5-3
ebXML 5-4
OASIS 5-4
Alternate Names 5-6
External Data 5-7
Recommendations 5-8
Chapter 6 Business Processes 6-1
Business Processes 6-1
Trading Partner Interchange 6-2
Recommended Modeling Methodology 6-4
Business Operational View 6-5
Functional Service View 6-10
Recommendations 6-14
Chapter 7 Organizational Roles and Responsibilities 7-1
Roles 7-1
Registration Authority 7-1
Responsible Organization 7-2
Submitting Organization 7-3
Role Use 7-4
Recommendations 7-5
Chapter 8 Associations and Classifications 8-1
Associations 8-1
ebXML and OASIS 8-1
11179 8-3
Classifications 8-3
ebXML and OASIS 8-3
11179 8-6

RECOMMENDATIONS 8-8

Chapter 9 Registry Services 9-1

REGISTRY SERVICES INTERFACE 9-1

OASIS 9-1

ebXML 9-2

11179 9-3

QUERY SERVICES 9-3

OASIS 9-3

ebXML 9-4

11179 9-5

RECOMMENDATIONS 9-6

Chapter 10 Registry Administration 10-1

SUBMISSION TO THE REGISTRY 10-1

OASIS 10-1

ebXML 10-2

11179 10-2

OBJECT LIFE CYCLE 10-2

OASIS 10-2

ebXML 10-3

11179 10-4

OBJECT STABILITY 10-7

VERSION CONTROL 10-8

OASIS 10-8

ebXML 10-8

11179 10-8

AUDIT TRAIL 10-9

OASIS 10-9

ebXML 10-9

11179 10-10

RECOMMENDATIONS 10-11

Chapter 11 Infrastructure 11-1
TRADING PARTNER PROFILES AND AGREEMENTS 11-1
ebXML 11-3
OASIS 11-3
11179 11-3
MESSAGING SERVICE PROTOCOL 11-4
ebXML 11-4
OASIS 11-6
11179 11-7
SECURITY 11-7
ebXML 11-8
OASIS 11-10
11179 11-10
RECOMMENDATIONS 11-11
Chapter 12 Additional Considerations 12-1
LEVELS OF CONFORMANCE 12-1
ebXML 12-1
OASIS 12-2
11179 12-2
QUALITY CONTROL 12-2
ebXML 12-2
OASIS 12-2
11179 12-3
INTEROPERABILITY 12-3
ebXML 12-3
OASIS 12-3
11179 12-3
LEGAL LIABILITY 12-3
DISASTER RECOVERY 12-4
Chapter 13 Summary of Recommendations 13-1

FIGURES

Figure 1-1. EPA as Coordinator of XML Standards 1-2
Figure 2-1. Overview of Interaction Between a Registry and a Repository 2-1
Figure 3-1. ebXML Registry Architecture 3-3
Figure 3-2. Exchange Network 3-5
Figure 5-1. 11179 Registry Metamodel—Administration Metamodel Section 5-3
Figure 5-2. ebXML Registry Information Model 5-4
Figure 5-3. OASIS Registry/Repository Information Model—Registered Object 5-5
Figure 5-4. OASIS Registry/Repository Information Model—Registry Entry 5-5
Figure 6-1. Trading Partner Interchange 6-3
Figure 6-2. Open-EDI Reference Model 6-4
Figure 6-3. Business Operational View 6-5
Figure 6-4. Example of a Use-Case Diagram 6-7
Figure 6-5. Example of an Activity Diagram 6-8
Figure 6-6. Example of a Sequence Diagram 6-8
Figure 6-7. Example of a Class Diagram 6-9
Figure 6-8. Example of a Collaboration Diagram 6-9
Figure 6-9. Example of a State Diagram 6-10
Figure 6-10. The Functional Service View 6-11
Figure 6-11. The FSV Implementation Phase 6-12
Figure 6-12. The FSV Discovery and Retrieval Phase 6-13
Figure 6-13. The FSV Run-Time Phase 6-13
Figure 7-1. Example of a Hierarchical Registration Authority 7-2
Figure 7-2. Registration Environment 7-4
Figure 8-1. ebXML Classification Scheme 8-4
Figure 8-2. Automotive Classifications 8-5
Figure 8-3. Context-Sensitive Classification 8-6
Figure 8-4. An Example of Binomial Nomenclature Taxonomy 8-7
Figure 9-1. ebXML Registry Services Model 9-2
Figure 10-1. Life Cycle of an ebXML Registered Object 10-3
Figure 10-2. The Review Process 10-7
Figure 11-1. Format of an ebXML Message 11-4

Figure 11-2. ebXML Messaging Layers 11-5
Figure 11-3. ebXML Messaging Service Architecture 11-6
Figure 11-4. Security Levels for Information Flow 11-7
Figure 11-5. ebXML Security Information Model 11-9

TABLES

Table 4-1. Recommendations, Determinations, and Issues—Proposed Architecture 4-3
Table 5-1. Recommendations, Determinations, and Issues—Information Models 5-8
Table 6-1. Recommendations, Determinations, and Issues—Business Processes 6-14
Table 7-1. Recommendations, Determinations, and Issues—Organizational Roles and Responsibilities 7-5
Table 8-1. Association Roles in a Registry 8-2
Table 8-2. NAICS Code Structure 8-3
Table 8-3. Recommendations, Determinations, and Issues—Associations and Classifications 8-8
Table 9-1. Recommendations, Determinations, and Issues—Registry Services 9-6
Table 10-1. Example of the 11179 Standard Registration Process 10-5
Table 10-2. Recommendations, Determinations, and Issues—Registry Administration 10-11
Table 13-1. Recommendations, Determinations, and Issues—Sorted By Topic 13-1
Table 13-2. Recommendations, Determinations, and Issues—Sorted By Timeframe 13-6
Table 13-3. Recommendations, Determinations, and Issues—Sorted By Recommendation Type 13-11

Appendix A Glossary

Appendix B Bibliography

Appendix C Abbreviations

Chapter 1
Introduction

OBJECTIVE

As EPA continues with its e-government initiatives, it needs to implement an XML registry to offer a central location for XML and e-business resources. By working with the Environmental Council of the States (ECOS) and the states to implement an XML registry, EPA will be able to collaborate continually with states and other organizations. States can create XML document type definitions (DTDs) and Schemas and submit them to the XML registry for EPA or other states to use.[1] The Environmental Information Exchange Network, which is being designed by the State/EPA Information Management Work Group (IMWG) to improve the quality and availability of environmental data, will help make the XML registry accessible to states and industry partners.

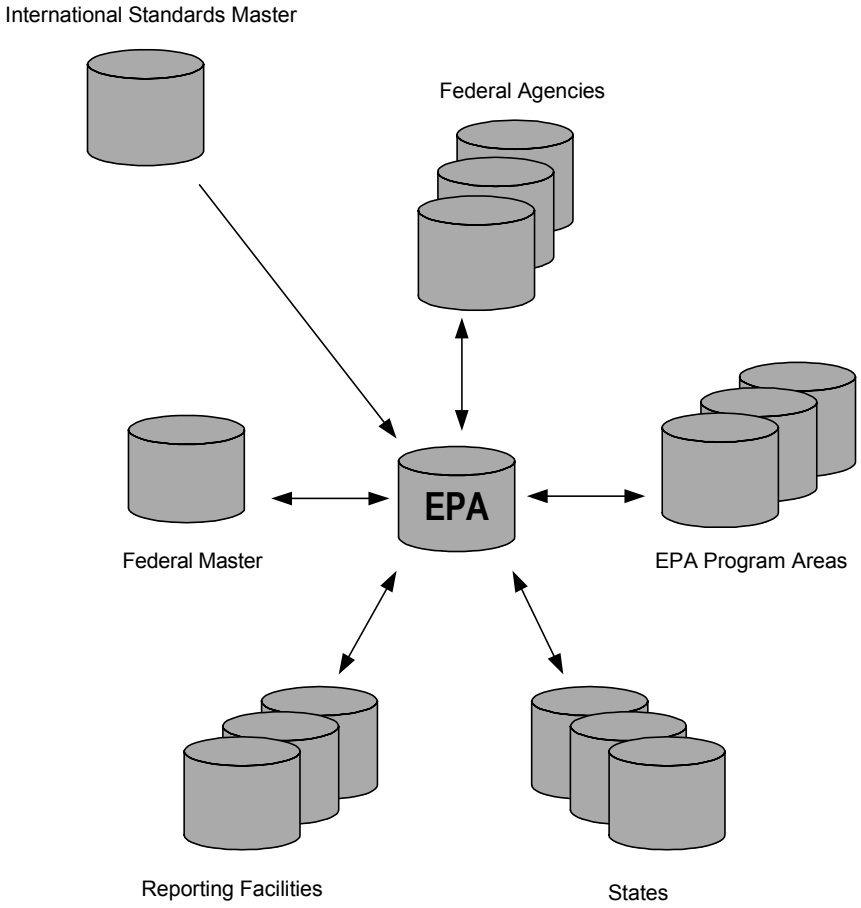
EPA can benefit greatly by using an XML registry, which will make data more coherent throughout the agency by standardizing formats for transmitting data. The registry can promote reuse by enabling system developers to access available resources (such as schemas and trading partner agreements) through a central registry rather than creating their own resources. Without an XML registry, users risk making format errors in data transmissions, increasing the time and effort required to implement such transmissions.

By implementing an XML registry, EPA can coordinate XML and data standards across industry and state data exchanges using federal standards, as shown in Figure 1-1.

In this report, we discuss the requirements for implementing an Extensible Markup Language (XML) registry in EPA. We outline industry models and standards that pertain to the concept of an XML registry. We also describe a proposed architecture and raise functional and policy issues EPA must consider before implementing an XML registry. Where appropriate, we've given examples that are specific to EPA's business needs.

[1] We use the term "schema" throughout this report to refer to an XML information model that may be a DTD or a W3C XML Schema.

Figure 1-1. EPA as Coordinator of XML Standards



Source: LMI.

REQUIREMENTS

Although an XML registry must have *repository* functionality, we focus on *registry* functionality in this report. A registered object does not have to be stored in a repository connected to the registry in which it is registered—i.e., a unique identifier (UID) for the object may reference the location of the object elsewhere. Therefore, a registry can either operate independently or be paired with a repository. In addition to determining registry functionality, EPA will need to determine where its repository will physically reside, and if more than one repository will be required.

EPA's XML registry must

- store XML schemas, XML documents, and business information content;
- accept submissions from states and industry partners, including modifying existing XML components;

- be accessible from the Environmental Information Exchange Network;
- align with commercial efforts as stated in the Office of Management and Budget (OMB) Circular A-119 and the National Technology Transfer and Advancement Act of 1995 (NTTA);
- contain appropriate security features;
- adhere to widely adopted XML registry standards;
- have commercial off-the-shelf (COTS) software available for implementation;
- assure consistency of XML tags used in XML documents;
- enable human and machine discovery of XML registry content;
- have a centralized architecture, with support for distributed architecture in future; and
- have one or more repositories.

REPORT ORGANIZATION

This report is not an implementation guide for the registry models; rather, it is an overview of functionality and issues EPA should consider before implementation. We describe registry concepts at a high level wherever possible, and in more detail where necessary to convey the proper context. We organized the chapters by functionality rather than by model for ease of reading—descriptions of OASIS and ISO/IEC 11179 registry models are included for informational purposes only.

At the end of each chapter, we describe our specific recommendations. Each recommendation fall into one of three categories—specific recommendations, determinations yet to be made, or open issues to be addressed.

Each recommendation, determination, or issue is assigned a number in the form of "X.Y," where X is the chapter number and Y is a consecutive number, starting with "1." Each item also contains a "type" code with the following possible values:

- R = recommendation
- D = determination
- I = issue.

Each item also is assigned a specific timeframe. We use the following timeframe categories:

- ST/IM = short term–immediate
- ST/NI = short term–non-immediate
- MT = medium term
- LT = long term

The two short-term categories differentiate the immediacy of each item. This approach enables EPA to construct an implementation timeline. EPA should determine the duration of each task itself before implementation.

The remainder of this report is organized as follows:

- Chapter 2, *XML Registry–General Concept*, describes the general concept and benefits of an XML registry.
- Chapter 3, *Models and Standards*, is an overview of the registry models and standards described in this report, and describes developing standards that pertain to implementing an XML registry.
- Chapter 4, *Proposed Architecture*, further describes the architecture that we recommend EPA use for implementing an XML registry.
- Chapter 5, *Information Models*, defines the information model for each registry model we investigated, including the types of objects that can be stored in an XML registry and the metadata to be stored for each object.
- Chapter 6, *Business Processes*, describes the high-level business process concepts defined as part of the ebXML specification.
- Chapter 7, *Organizational Roles and Responsibilities*, defines the roles and responsibilities of the different organizations that may operate and use an XML registry.
- Chapter 8, *Associations and Classifications*, describes the concepts of associations and classifications in a registry model, and gives examples of their use.
- Chapter 9, *Registry Services*, defines the registry services aspect of the registry models and discusses operations that can be performed on registered objects.
- Chapter 10, *Registry Administration*, is an overview of administrative functions for an XML registry.

- Chapter 11, *Infrastructure*, discusses concepts of the infrastructure that surrounds the XML registry.
- Chapter 12, *Additional Considerations*, addresses levels of conformance, legal liability, and disaster recovery.
- Chapter 13, *Summary of Recommendations*, a summary of the recommendations that we describe at the end of each chapter.
- Appendix A—*Glossary*
- Appendix B—*Bibliography*
- Appendix C—*Abbreviations*

Chapter 2
XML Registry—General Concept

This chapter describes the general concept of an XML registry and the benefits that may be realized by implementing an XML registry in an organization.

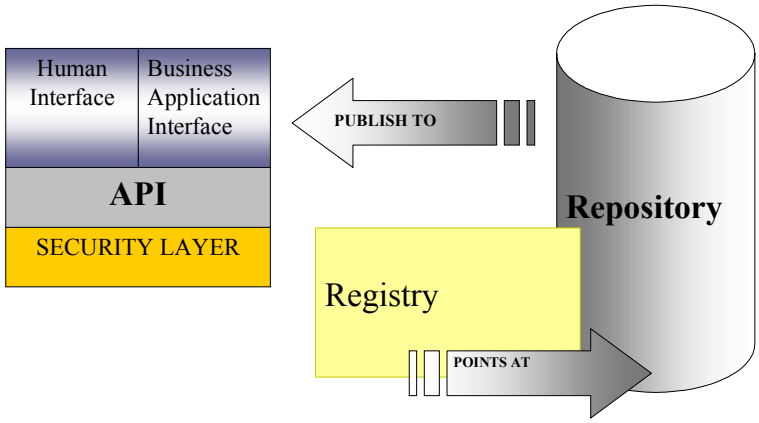
WHAT IS AN XML REGISTRY?

An XML registry is an integrated software system that consists of two distinct components–a registry and a repository. A *registry* is a facility that stores relevant descriptive information (known as metadata) about registered objects, and allows the metadata to be operated on in various ways. A *repository* is a storage facility for registered objects with an access method that enables retrieving individual objects, perhaps with an additional authentication and permission layer.

We use the term "XML registry" to refer to what is commonly known as an "XML registry/repository." Registry means "registry/repository" only if the implementation also stores registered objects. We use the word "repository" only when referring to the name of an official committee or document, such as the "OASIS Registry and Repository Technical Committee" or the "OASIS Registry/Repository Technical Specification."

The interaction of a registry and repository is shown in Figure 2-1.

Figure 2-1. Overview of Interaction Between a Registry and a Repository



Source: ebXML.

A *registered object* is something that an author or producer wants to have visible to the world so a client or supplier can use it. Examples of registered objects are

- XML schemas,
- XML documents,
- trading partner agreements,
- business process descriptions, and
- software components.

For example, if a laboratory wants to submit emissions data to EPA electronically using XML, a representative of the laboratory may request a schema from an XML registry for structuring their submission. The representative may search in the registry for schemas that are designed for submitting emissions data, and select and download the proper schema on the basis of additional criteria (perhaps the type of emission). If multiple versions of the required schema exist, the representative may select a version based on particular criteria (perhaps a version that they used in an earlier submission). Once the representative has a local copy of the schema, he can use it to define the structure of his data submission.

WHAT ARE THE BENEFITS OF AN XML REGISTRY?

The benefits of an XML registry are numerous:

- *Promotes reuse*—If system developers can locate an XML schema in an XML registry and use that schema, they will save the time and effort of developing the schema themselves.
- *Enables efficient version control*—An XML registry enables tracking multiple versions of a registered object efficiently.
- *Promotes unified understanding of registered objects*—Because metadata for registered objects are accessible from a single location, a unified understanding of the purpose of the registered objects is promoted throughout an organization.
- *Ensures consistency across organizational areas*—A central location for registered objects helps ensure they are used consistently in different parts of an organization.
- *Promotes selective access to registered objects*—Access controls in an XML registry can ensure that read-only or open access to registered objects is given according to organizational area.

- *Enables collaborative development*—Users can create items, such as schemas, and submit them to an XML registry for use and potential enhancement by others. For example, a state can create a schema and submit it to the XML registry; another state can download this schema, update it to meet their needs, and resubmit it as a different version of the schema. Others then can access the new version as well.

Chapter 3
Models and Standards

This chapter is an overview of the registry models and standards that we discuss in this report, as well as background information about the sponsoring organization for each model and standard. We also discuss the Network Blueprint, which is a blueprint for the Environmental Information Exchange Network developed by the IMWG. We then discuss industry and federal government initiatives that EPA should participate in or monitor. Finally, we discuss other registry models that we did not consider for the XML registry but that are important to be aware of.

ISO/IEC 11179

ISO and IEC form a specialized system for worldwide standardization. The ISO/IEC 11179 standard, Specification and Standardization of Data Elements, describes the standardization and registration of data elements for sharing them and making them understandable. This standard gives guidance for formulating and maintaining descriptions of the meaning of data elements (metadata) so they are formulated consistently and are standardized. The 11179 standard enables the end user to interpret the intended meaning of data elements confidently, correctly, and unambiguously.

The 11179 standard does not specify using XML—rather, it explains the concept of data elements and how they are defined.

The 11179 standard is divided into six parts:

- *11179-1, Framework for the Specification and Standardization of Data Elements*, is an introductory section.
- *11179-2, Classification of Data Elements*, is a basis for building classification structures and documenting the classification aspects of data elements through a specific set of attributes.
- *11179-3, Basic Attributes of Data Elements*, describes the attributes that should be defined for data elements to promote a common understanding of the meaning and representation of data elements.
- *11179-4, Rules and Guidelines for the Formulation of Data Definitions*, contains the rules and guidelines for constructing well-formed definitions of data elements to convey their proper meaning.

- *11179-5, Naming and Identification Principles for Data Elements*, contains the rules, principles, and guidelines for naming and identifying data elements so they are uniquely identified in a registry.

- *11179-6, Registration of Data Elements*, describes the procedure for registering and assigning an internationally unique UID for data elements, and defines the roles and responsibilities of the organizations involved in the registration process.

This report draws information from parts 11179-1, 11179-2, 11179-3, and 11179-6.

The 11179 standard is the basis for a document, "Concept of Operations for a Data Registry," which addresses registration of data elements as they are described in the 11179 standard. This document defines a data registry as "an information resource kept by a Registration Authority that describes the meaning and representational form of data units including data element identifiers, definitions, units, allowed value domains, etc." We refer to this as the "11179 registry model." EPA's EDR is an implementation of an 11179 registry.

ISO/IEC 11179-3 is being revised to incorporate the notions of the American National Standards Institute (ANSI) X3.285 standard, "Metamodel for the Management of Shareable Data." This new version, known as the "registry metamodel (MDR3)," extends the original version to include registering objects. We refer to this new version as the "11179 registry metamodel." The X3.285 standard specifies the structure of a data registry as a conceptual data model and provides the attributes for identifying the characteristics of data that are necessary to clearly describe, inventory, analyze, and classify data. All other parts of the ISO/IEC 11179 standard also are being harmonized to use the same terminology as the 11179 registry metamodel.

EBXML

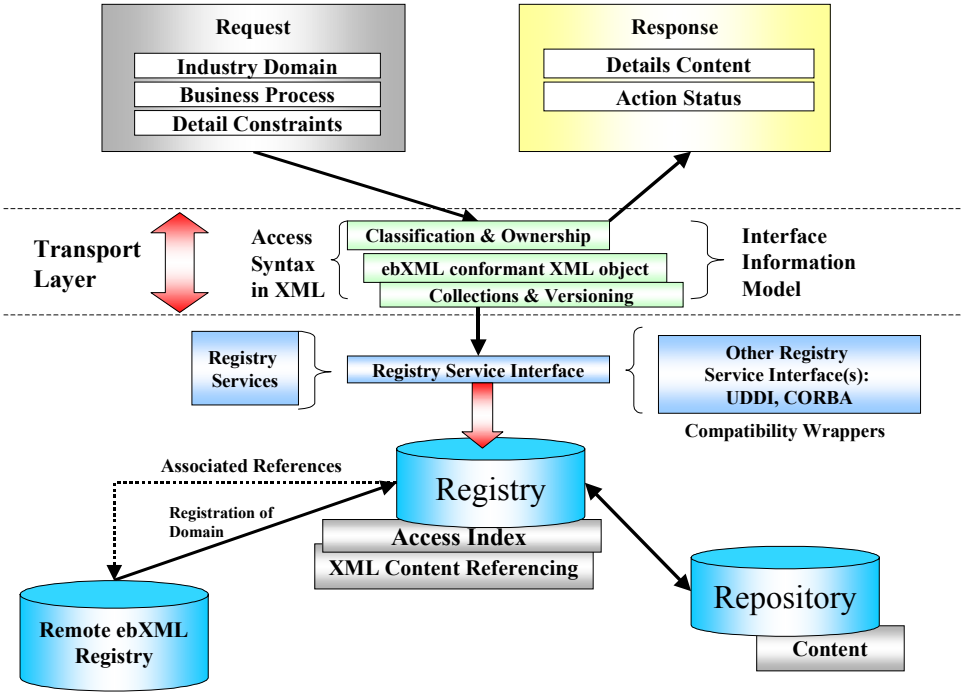
ebXML is a joint venture between OASIS and UN/CEFACT. The vision of the ebXML project is the enabling of a global electronic marketplace where enterprises of any size and in any geographic location can meet and conduct business with each other by exchanging XML-based messages.

The ebXML project defines an entire e-commerce infrastructure, of which the registry is an integral part. The ebXML registry specification grew out of the OASIS registry specification—therefore, significant overlaps in functionality between the two specifications exist.

A number of teams are working on the ebXML project, each of which creates specifications for a specific technical area. One team, the ebXML Registry and Repository Project Team, has members from various organizations, including Sun

Microsystems and the Automotive Industry Action Group. Figure 3-1 shows the ebXML registry architecture.

Figure 3-1. ebXML Registry Architecture



Source: ebXML.

The figure depicts an ebXML registry interacting with both 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).

OASIS

OASIS is a nonprofit international consortium that creates interoperable industry specifications based on public standards, such as XML and Standard Generalized Markup Language (SGML). OASIS is not a standards body—rather, its work complements that of standards bodies. OASIS provides an open forum for its members to discuss market needs and directions and to recommend guidelines for product interoperability. OASIS members include organizations and individuals who provide, use, and specialize in implementing the technologies that make these standards work in practice.

Although the ebXML project defines an entire e-commerce infrastructure, the OASIS project is concerned solely with the specification for an XML registry.

OASIS has technical committees whose main purpose is to create standards and specifications for specialized areas of technology. One committee, the OASIS Registry and Repository Technical Committee, has members from different organizations, including

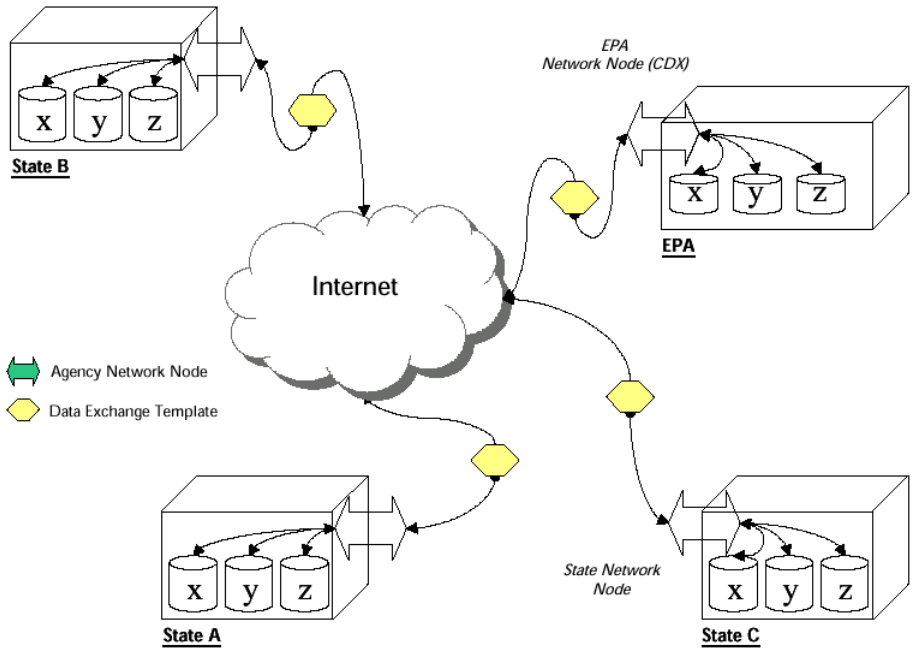
- SUN Microsystems,
- National Institute of Standards and Technology (NIST),
- IBM,
- Data Channel, and
- Boeing.

NETWORK BLUEPRINT

The IMWG has created a blueprint, known as the "Network Blueprint," for an Environmental Information Exchange Network. The network is expected to improve dramatically the quality and availability of environmental data to environmental agencies and the public, and its design is based on the observations and findings of the IMWG. The IMWG is a work group comprising EPA and state environmental agencies organized through the Environmental Council of the States (ECOS), whose mission is to build locally and nationally accessible, cohesive, and coherent information systems. XML is the sole choice of format for data exchange on the network.

In addition to architecture, the Network Blueprint contains information about formats for data exchange and trading partner agreements, technical infrastructure, and network administration. The initial participating entities will be state environmental agencies and EPA. The conceptual diagram of the exchange network is shown in Figure 3-2.

Figure 3-2. Exchange Network



Source: IMWG.

OTHER INITIATIVES

Other initiatives are taking place in industry and the federal government that do not currently affect the XML registry implementation, but which EPA should participate in or monitor.

Federal CIO Council

The federal government's Chief Information Officers Council (Federal CIO Council) is the principal interagency forum for improving practices in the design, modernization, use, sharing, and performance of federal agency information resources. The Federal CIO Council hosts an XML working group whose purpose is to accelerate, facilitate, and catalyze the effective and appropriate implementation of XML technology in the information systems and planning of the federal government.

XML.gov

XML.gov is a website hosted by the Federal CIO XML working group whose purpose is to facilitate the efficient and effective use of XML through cooperative efforts among government agencies, including partnerships with commercial and industrial organizations. The public can contribute to this site. The working group is considering establishing a registry of "inherently governmental" data elements and schemas, and is asking for the public's input on this issue.

OTHER REGISTRY MODELS

Several other registry models exist that we did not recommend for consideration in implementing an XML registry. We mention them below only for information.

UDDI

The UDDI initiative is a collaboration between IBM, Ariba, and Microsoft. Its primary focus is on enabling large organizations to manage their network of smaller business customers through a shared operation of a business registry on the web. Discussions are under way to allow interoperability between ebXML and UDDI for business discovery, in which UDDI acts as a central "registry of registries" for ebXML. In this scenario, a user can use UDDI to discover an ebXML registry that is likely to contain the content in which they are interested. The user then can go to that registry and discover potential business partners by using the discovery capabilities of ebXML.

We did not consider UDDI because it does not address registration of XML components.

BizTalk

BizTalk is a Microsoft framework for XML application integration and e-commerce that enables organizations to produce XML schemas that are consistent. BizTalk framework schemas may be stored on the BizTalk.org website by individuals or organizations by using automated submission and validation. Individuals and organizations can freely use XML schemas from the BizTalk.org website if the schema is published for public use.

We did not consider BizTalk because it is too proprietary a solution, and has limited administrative controls.

XML.org

XML.org is an open, vendor-neutral website for XML resources hosted by OASIS. The XML.org site contains a registry that is a central clearinghouse for developers and standards bodies to publicly submit, publish, and exchange XML schemas, vocabularies, and related documents. As with BizTalk, the XML.org registry has limited administrative controls.

Chapter 4
Proposed Architecture

In this chapter, we describe a proposed architecture for an XML registry. The proposed architecture is based on the ebXML model, with the XML tags used in schemas stored in a second registry that is based on the ISO/IEC 11179 registry model. Although the ISO/IEC 11179 registry model has more comprehensive administrative controls than the ebXML model, from our discussions with vendors we understand that COTS software known as "ebXML-compliant" software will contain administrative controls. For this reason, we recommend that all of the functionality for registered objects be concentrated solely in the ebXML registry, and that the 11179 registry be used only for storing XML tags.

EPA's EDR will be used as the 11179 registry. The EDR is an implementation of an 11179 registry, and is a comprehensive, authoritative source of reference about environmental data. The EDR does not contain environmental data itself, but rather metadata, which describes the data to make the data more meaningful. The EDR contains the following information:

- EPA data standards and business rules
- Specifications for well-formed data elements in systems design
- Numerous international- and agency-standard code sets that can be downloaded for reuse for developing application systems
- EPA application system data elements and values.

XML tags will be created for all EDR data standards, and for new data elements that are submitted. EPA already is creating XML tags for data standards.

Schemas that are submitted to the XML registry are expected to contain data elements that are extracted from the EDR. If the elements in a submitted schema do not exist in the EDR, they will be added to the EDR. This will ensure that a user can identify in the EDR the elements contained in a schema. EPA should create policies and procedures for adding data elements to the EDR.

MULTIPLE CONTEXTS

The EDR also enables listing data elements under different contexts. For instance, the term *analyte* may have one meaning in one EPA program, but another meaning in another program. The 11179 registry model allows for a data element in an 11179 registry to exist with more than one context, using a different name for

each context. EPA can extend this capability to the registration of different XML tags for different contexts, if this is needed.

ARCHITECTURE ISSUES

Centralized Versus Distributed

An XML registry implementation may have a *distributed* architecture, in which several registries exist and interact with a "central" XML registry. For example, a state may have its own XML registry that interacts with the EPA XML registry. A distributed architecture requires complex version control mechanisms to ensure that all XML registries remain synchronized. For this reason, we recommend that EPA start with a centralized approach, in which all users interact with one central XML registry. A central registry enables stricter control of versions and what is submitted. Over time, the architecture can be changed to a "distributed" architecture.

Static Versus Dynamic

An XML registry implementation may have a *static* configuration, in which objects can be submitted, but registered objects cannot be updated and deleted, or a *dynamic* configuration, in which submitting, updating, and deleting objects are allowed. We recommend that EPA implement a combination of both configurations. The combination would enable users to update registered objects by submitting new versions but would not allow deleting objects (the registration authority would delete the objects).

If users were allowed to delete or modify registered objects, a user who inadvertently deleted his local copy of a registered object and attempted to download the same version of the registered object could find that the version no longer exists. The missing version could cause functionality to break if the functionality relied on that version of the registered object.

RECOMMENDATIONS

Table 4-1 lists the recommendations, determinations, and issues for this chapter and shows their timeframes.

Table 4-1. Recommendations, Determinations, and Issues—Proposed Architecture

ID No.	Topic	Type	Description	Time-frame	Comments
4.1	XML Tags	R	Allow XML tags to be stored in EDR	ST/IM	In progress
4.2	Registration of Data Elements	R	Create policies and procedures for adding data elements contained in schemas to EDR	ST/NI	
4.3	Multiple Contexts	R	Allow XML tags to be registered in EDR under multiple contexts	MT	
4.4	Architecture	R	Implement centralized architecture	ST/NI	
4.5		R	Implement distributed architecture	LT	
4.6		R	Implement combination of static and dynamic architecture	N/A	
4.7	Repository	D	Determine physical location of repository	ST/IM	
4.8		D	Determine number of repositories required	ST/IM	

Note: R = recommendation, D = determination, ST/IM = short term–immediate, ST/NI = short term–non-immediate, MT = medium term, LT = long term, C = continuous.

Chapter 5
Information Models

This chapter describes the information model for each of the registry models that we discuss in this report. An information model is a representation of the components that compose a system and the relationship between those components. The information in this chapter can help EPA define what types of objects will be stored in the XML registry, how these objects will be organized, and the types of metadata that will be stored for these objects.

REGISTERED OBJECTS AND METADATA

A *registered object* is something that an author or producer wants to have visible to the world so a client or supplier can use it. An example of a registered object is an XML schema. A *registry entry* is relevant descriptive information, or *metadata*, about a registered object. Registered objects are stored in a *repository*, while registry entries are stored in a *registry*.

In general, the following metadata may be stored for a registered object:

- *Name*: Name for the registered object intended for human reading.
- *Version*: Used to distinguish between two registered objects having the same name.
- *Object identifier*: A UID for the registered object that a registry entry references.
- *Object type*: Valid values that differ by model, examples are:
  - *Definition*: An XML definition document, such as a schema.
  - *Instance*: An XML document.
  - *Classification scheme*: A specification of a hierarchy of values, names, and codes on which a classification is based.
  - *MIME type*: One of the valid Multipurpose Internet Mail Extension (MIME) media types as specified by the Internet Assigned Numbers Authority (IANA), such as text, images, audio, and video.
  - *Other*: A document with no specific object type, for example a Word document.

- *Process*: An object that describes a business process.

- *Registration status*: Status of a registered object in a registry at a specific time.

- *Submitting organization (SO)*: An individual or organizational element designated to identify and report data elements suitable for registration. The entity that originally registered an object.

- *Responsible organization (RO)*: An individual or organizational element responsible for the accuracy, reliability, and currency of the metadata for a registered object. The RO may be the same as the SO. Also referred to as a *data steward*.

- *Registration authority (RA*): A recognized expert organization that is responsible for populating and maintaining the registry.

- *Expiration date*: Assigned by the RA at the suggestion of the SO.

- *Stability*: Likelihood that a registered object will change in the future.

TERMINOLOGY

Some terminology differs depending on the models. The ebXML model refers to a registered object as a repository item. The 11179 registry metamodel refers to a registry entry as an administration record. The 11179 registry metamodel also refers to the registration of both data elements and administered components. An administered component is a component for which administrative information is recorded. In this report, we refer to both data elements and administered components when referring specifically to the 11179 standard, with the appropriate term used that is consistent with the current 11179 standard terminology. When we refer to the registration of a data element in an 11179 registry, the same concept applies to the registration of objects where it makes sense in the specific context.

For consistency, we use the terms “registry entry” and “registered object” throughout this report. The only exception is in those places where an ebXML model figure is listed, when we use the term “repository item.”

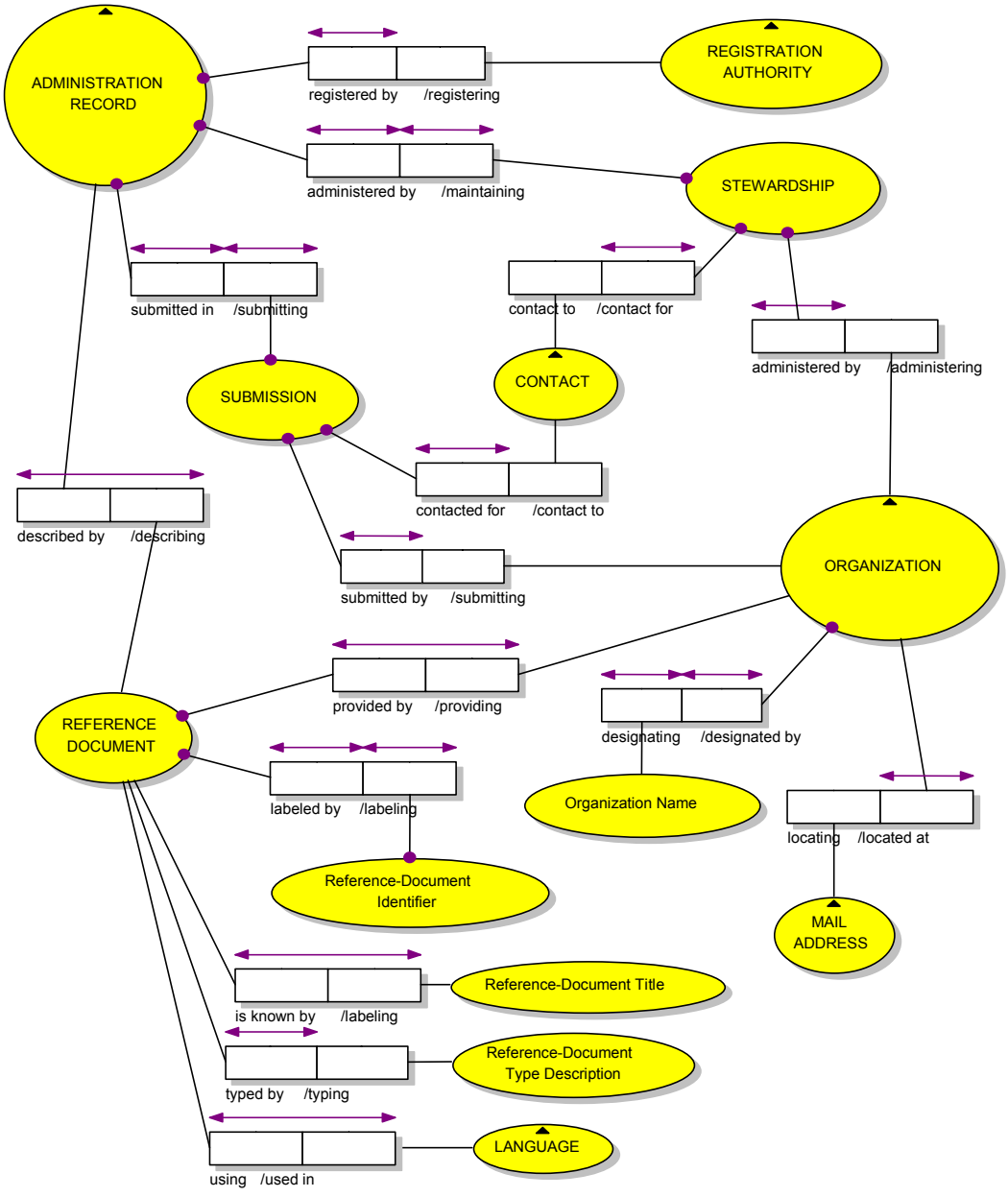
REGISTRY INFORMATION MODELS

This section describes the registry information model for each of registry models that we discuss in this report.

11179

Figure 5-1 is the administration metamodel section of the 11179 registry metamodel, with an administration record as its central focus.

Figure 5-1. 11179 Registry Metamodel—Administration Metamodel Section

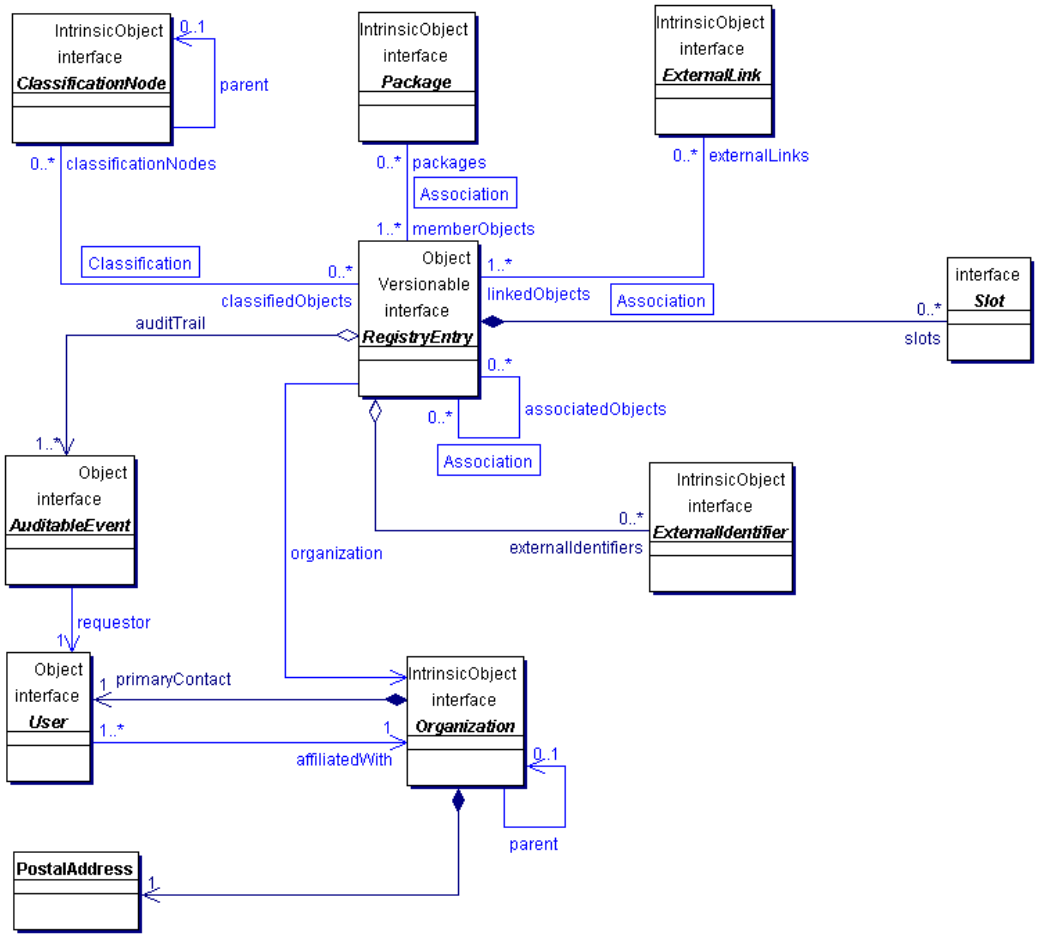


Source: ISO/IEC.

ebXML

Figure 5-2 is the ebXML registry information model, with a registry entry as its central focus.

Figure 5-2. ebXML Registry Information Model



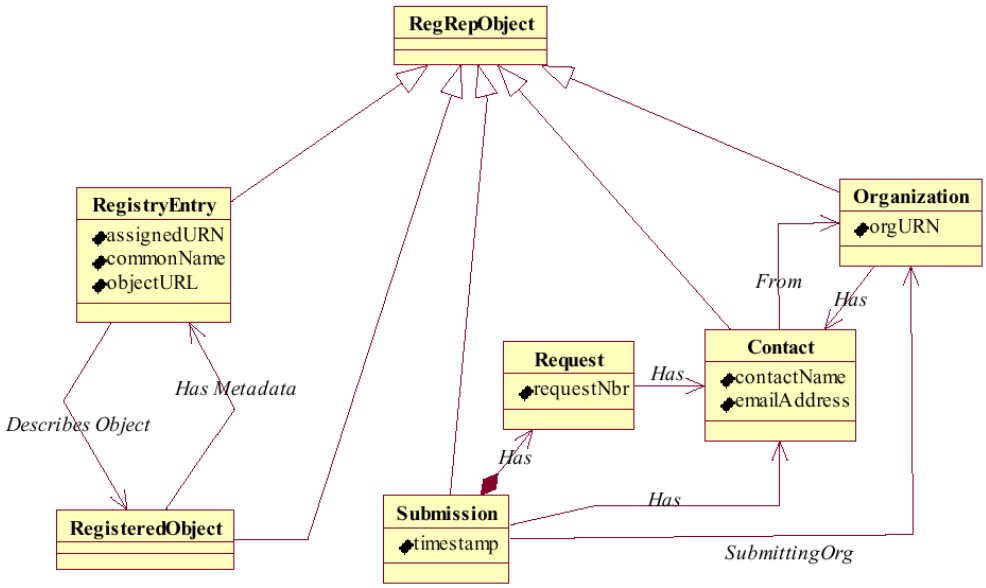
Source: ebXML.

In the ebXML registry information model, metadata can be dynamically added to a registry entry by using “slots.” A registry entry may contain no slots or many slots. This concept enables extensibility in the registry information model.

OASIS

The OASIS registry/repository information model is separated into two class diagrams, one describing a registered object and the other describing a registry entry. Figure 5-3 shows a registered object.

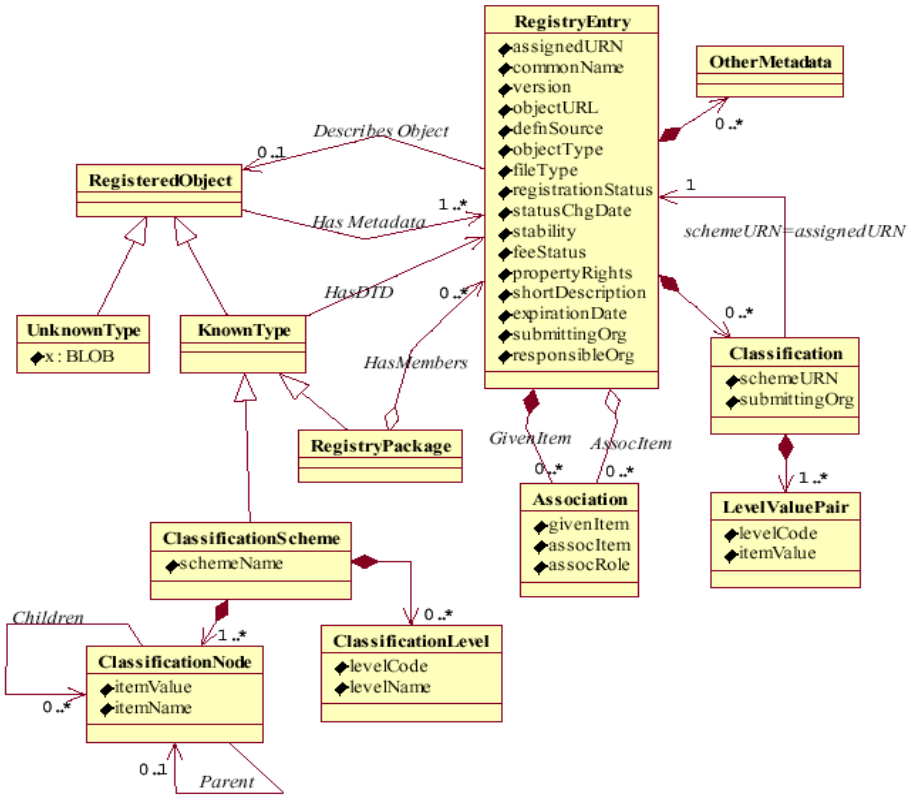
Figure 5-3. OASIS Registry/Repository Information Model—Registered Object



Source: OASIS.

Figure 5-4 shows the OASIS registry/repository information model with a registry entry as its central focus.

Figure 5-4. OASIS Registry/Repository Information Model—Registry Entry



Source: OASIS.

The models shown above have some object classes in common. The common object classes are:

- *Submission*: Meaning differs depending on model.
- *Organization*: An organization that has a relationship to a registered object. The organization can be an SO, an RO, or an RA.
- *Association*: Defines associations between registered objects. For instance, a registered object may be associated with a submission (in which the object was registered), or a classification node (which specifies a classification for the registered object).
- *Contact*: A person, role, or other entity in an organization that has some relationship to a registered object (perhaps the person who submitted the object initially for registration).
- *Classification*: Used to classify registered objects in a classification scheme.

ALTERNATE NAMES

An *alternate name* is a name valid in a particular context that is associated with a registered object. By using alternate names, a registered object can be considered in multiple contexts. Although the ebXML model has deferred specifics about alternate names to a future release, the OASIS model states that an SO may provide one or more alternate names for an object when it is registered, or for an existing object. In the latter case, the alternate name is provided as an "add alternate name" request. For example, for a schema that is named "Air Emissions Point Source Submission schema," a program that deals exclusively with air emissions may want to refer to it as "Point Source schema." In this case, the program can register an alternate name of "Point Source schema" for this schema so they can refer to it as such in their activities.

An alternate name may have one of the following contexts in the OASIS model:

- *Language*: A name in a human-readable language (such as French or Spanish) that enables users who speak the language to retrieve registered objects.
- *Additional name*: A name for the registered object that is shorter or longer than its original name.
- *System-related*: A name appropriate for use in a particular programming language, such as Java or C++.

The SO for an alternate name does not need to be the organization that originally submitted the registered object. Thus, an SO can register an alternate name for any object that exists in the registry.

The 11179 standard allows for specifying a synonymous name and context for each data element. These two attributes together can provide an alternate name for a data element, thereby promoting cross-cultural and cross-organizational information interchange.

EXTERNAL DATA

The term *external data* refers to information items that are related to a registered object but reside outside the registry. Such items provide support information. Examples of external data are

- user documentation for an XML schema,
- a white paper that describes a registered business object,
- vendor information,
- graphic visualizations, or
- the home page of a submitting organization.

For example, a schema that is used for submitting data that is specific to an EPA program may have an accompanying user manual that specifies how to extract data from the originating system for submission.

External data may be submitted for an object that is being registered, or for an existing (registered) object. In the OASIS and ebXML models, an external data item does not have the life cycle of a registered object—i.e., it cannot be withdrawn, replaced, etc. In both models, the relationship between a registered object and its external data is recorded by using an association in the registry.

The 11179 registry metamodel refers to reference documents rather than external data. A reference document is "a document that provides pertinent details for consultation about a subject." An administered component may have one or more reference documents. For this report, we view a reference document as equivalent to external data.

RECOMMENDATIONS

Table 5-1 is a list of the recommendations, determinations, and issues about information models that relate to EPA.

Table 5-1. Recommendations, Determinations, and Issues—Information Models

ID No.	Topic	Type	Description	Time-frame	Comments
5.1	Types of Registered Objects	D	Determine what types of registered objects will be allowed in the XML registry	ST/IM	
5.2	Registry Entry Contents	D	Determine the information that will be contained in a registry entry	ST/IM	
5.3	Naming of Registered Objects	D	Determine standards for naming registered objects	ST/NI	
5.4	Submission Validation	R	Ensure that all submitted XML documents and schemas are validated before being stored in the XML registry	ST/NI	
5.5		D	Determine procedures for handling submissions that do not validate properly	ST/NI	How to notify submitter?
5.6		I	Should an object that fails validation be stored until the valid version is received, or rejected completely?	ST/NI	May store with a special registration status, such as "invalid—new submission pending" May notify original submitter if new submission not received in a given timeframe
5.7	Alternate Names	R	Allow alternate names to be submitted for registered objects	ST/NI	
5.8		D	Determine policies and procedures for submitting alternate names	ST/NI	
5.9		I	Should the number of alternate names that can be associated with a registered object be limited?	ST/NI	
5.10	External Data	R	Allow external data to be submitted for registered objects	MT	
5.11		D	Determine policies and procedures for submitting external data	MT	What types of external data can be submitted (i.e., white papers, user manuals, etc.)?
5.12		I	Should the amount of external data that can be associated with a registered object be limited?	MT	
5.13	Versions	D	Determine policy for handling versions of registered objects in ebXML registry	ST/NI	How will SO determine version numbers? Should this be left entirely up to the SO?
5.14	Expiration Dates	I	Should an expiration date be assigned to registered objects?	MT	
5.15		D	If so, who will assign expiration date?	MT	RA or SO?

Chapter 6
Business Processes

This chapter describes the high-level concepts of business processes that are defined as part of the ebXML model, and how these processes are integrated with an XML registry. The information in this chapter can help EPA identify business processes and business process components that can be registered in the XML registry so EPA can integrate them into its business functions.

The ebXML specifications are a framework in which Electronic Data Interchange's (EDI's) substantial investments in business processes can be preserved in an architecture that exploits the technical capabilities of XML. The specifications describe using an ebXML-compliant registry as a support mechanism for realizing an organization's business goals.

Because the OASIS and 11179 registry models solely specify registry concepts, they do not address the business process concepts discussed in this chapter.

BUSINESS PROCESSES

A business process is defined as "a collection of business transactions between business partners." The process also may be an internal activity in one business. A business transaction is defined as "a logical unit of business conducted by two or more parties." A business transaction is a clearly defined exchange of business messages (in the form of business documents) resulting in a new legal or commercial state between two partners, known as *trading partners*. Although business practices vary greatly from one organization to another, most activities can be separated into processes that are specific for a type of business. An XML registry can be a storage facility for business processes developed by industry groups, small and medium-sized enterprises (SMEs), and other organizations.

A trading partner agreement (TPA) is a document that defines the conditions under which two partners will transact business together. The TPA outlines items, such as the following:

- Business scenarios to be used
- Messaging protocol
- Contingency issues
- Security requirements.

The ebXML specifications refer to a TPA as a collaboration protocol agreement (CPA). However, we use the term “trading partner agreement” throughout this report.

Business scenarios are XML versions of the business processes and associated business messages that an organization is able to engage in. The following are examples of business scenarios:

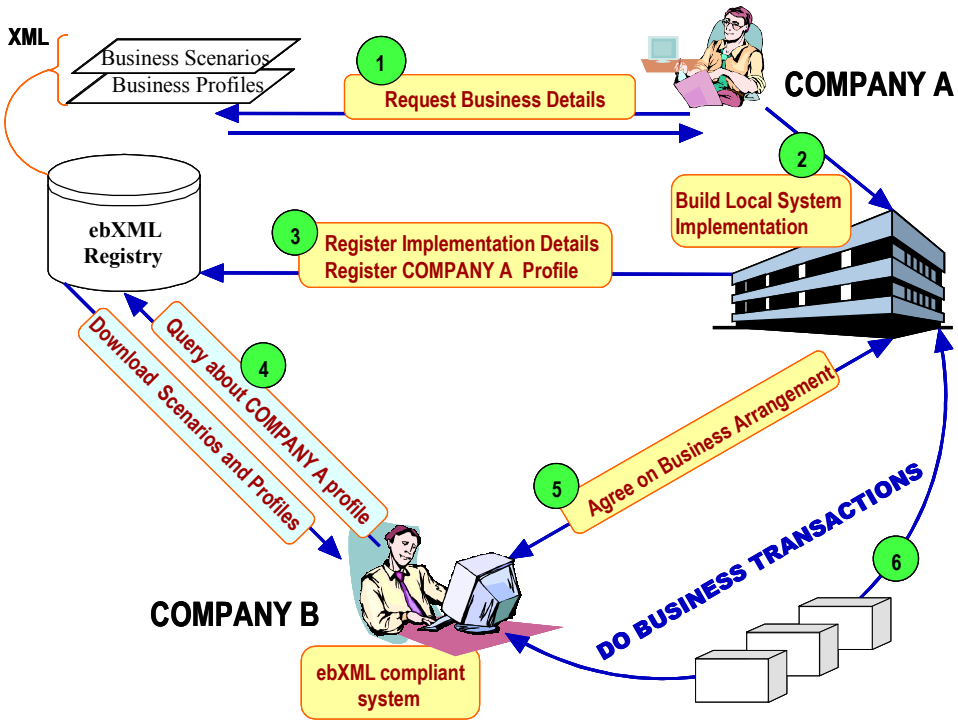
- Two trading partners set up an agreement and run the electronic exchange.
- Three or more trading partners set up a business process implementing a supply chain and run the associated exchanges.
- A company sets up a portal that defines a business process using external business services.
- Three or more trading partners conduct business using shared business processes and run the associated exchanges.

A trading partner profile (TPP) is a document that describes the business processes that an organization is able to engage in. The TPP lists the specific technological capabilities that a trading partner supports and the specific requirements that need to be met to exchange business documents with that trading partner. The ebXML specifications refer to a trading partner profile as a collaboration protocol profile (CPP). However, we use the term “trading partner profile” throughout this report.

TRADING PARTNER INTERCHANGE

The following is a high-level conceptual model for two trading partners who engage in a simple business transaction interchange using ebXML. The components of this model may be implemented incrementally as shown in Figure 6-1.

Figure 6-1. Trading Partner Interchange



Source: ebXML.

In the scenario shown in the figure, Company A has become aware of an ebXML-compliant registry that is accessible on the Internet. Company A reviews the contents of the registry to determine if it wants to become an ebXML-compliant participant (step 1). Company A decides it wants to become an ebXML-compliant participant and builds its own ebXML-compliant application (step 2). Company A also can elect to purchase a commercially available ebXML-compliant application rather than build its own. Company A then submits its trading partner profile to the ebXML-compliant registry (step 3). The trading partner profile describes the specific business scenarios that Company A is able to engage in. After receiving verification that the information in the trading partner profile is valid, the registry sends an acknowledgment to Company A.

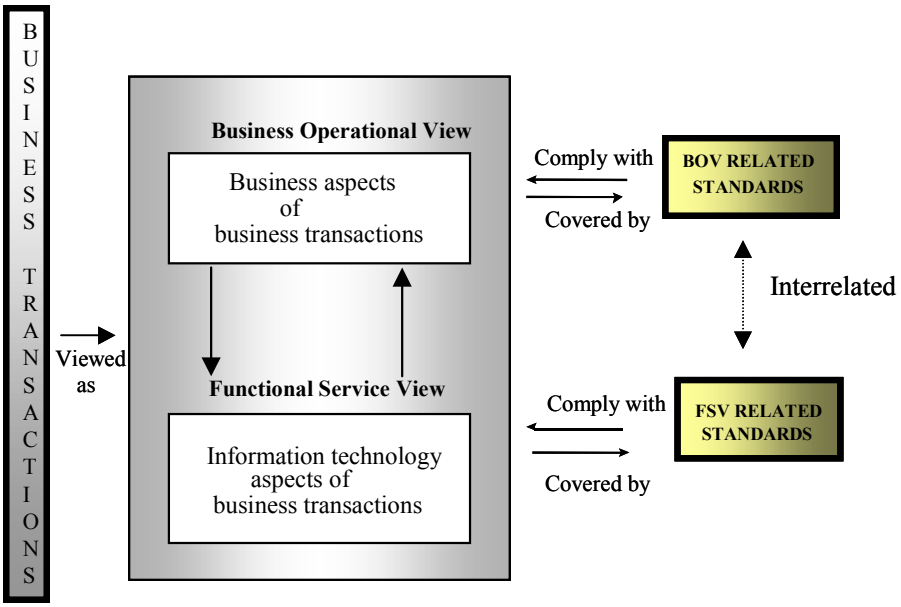
Company B discovers the business scenarios supported by Company A in the ebXML-compliant registry (step 4). Company B then sends a request to Company A stating that it would like to engage in a business scenario with Company A using ebXML. Company B submits a proposed trading partner agreement directly to Company A's ebXML-compliant software interface. The agreement outlines the specific business scenarios in which Company B would like to interact with Company A, and the conditions under which the business scenario will be conducted. Company A then accepts the business agreement (step 5). Company A and Company B then can engage in e-business using ebXML (step 6).

RECOMMENDED MODELING METHODOLOGY

Although business processes may vary greatly from one organization to another, most activities can be decomposed into generic business processes that are specific to certain types of industries. These business processes then can be modeled to create standardization in an organization. Business process modeling is not mandatory for ebXML compliance; however, if it is done, the UN/CEFACT Modeling Methodology (UMM) should be used. The UMM uses the Unified Modeling Language (UML), an object-oriented modeling tool developed by Rational Software and adopted as a standard by the Object Management Group (OMG). One of the primary benefits of using a consistent modeling methodology is that models can be compared to avoid duplication of existing business processes.

The UMM uses two separate views to describe the relevant aspects of business transactions—the business operational view (BOV) and the functional service view (FSV). These concepts stem from earlier work on a model called the open-EDI reference model (ISO/IEC 14662), shown in Figure 6-2.

Figure 6-2. Open-EDI Reference Model



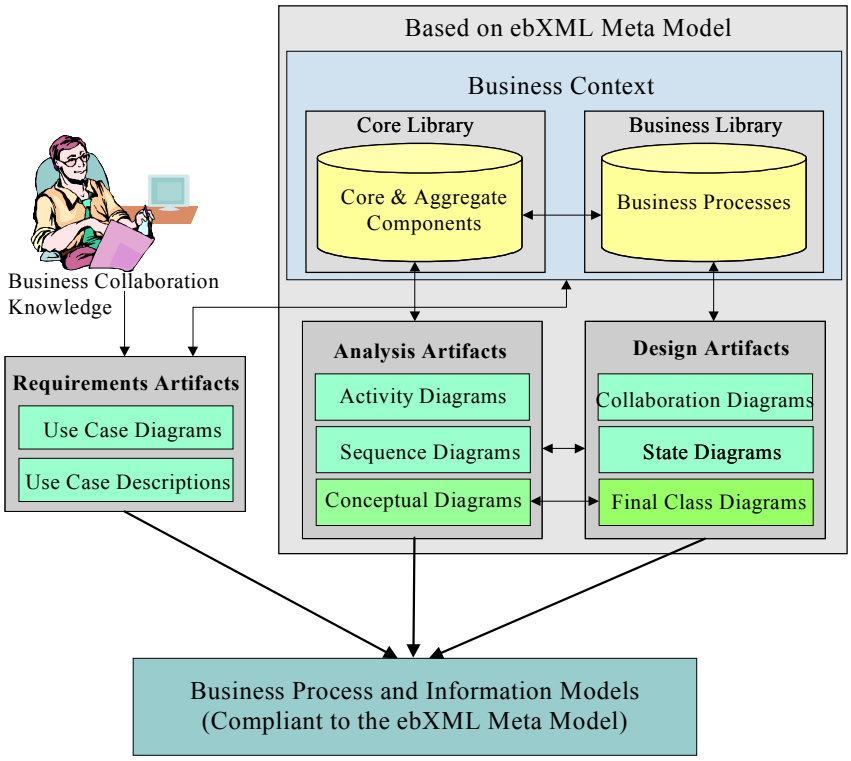
Source: ISO.

We discuss both of these views in further detail below.

BUSINESS OPERATIONAL VIEW

The business operational view deals with high-level business operation issues that apply to the business needs of ebXML trading partners. The BOV is shown in Figure 6-3.

Figure 6-3. Business Operational View



Source: ebXML.

In Figure 6-3, business collaboration knowledge is captured in a core library. A core library contains data and process definitions, including relationships and cross-references, expressed in business terminology that may be tied to an accepted industry classification scheme or taxonomy. The core library is the bridge between the specific business or industry language and the knowledge expressed by the models in a more generalized context-neutral language.

A core library comprises core processes and core components. A core process consists of a set of business actions independent of any industry specifics. Because core processes are generic, they can be reused with specific context and business rules in different vertical industries. A business process, therefore, can be composed of one or more core processes. A core component is a reusable, low-level data structure that captures information about a real-world business concept, and the relationship between that concept and other business concepts. A core component can be either an individual piece of business information or a naturally

integrated family of business information objects that can be assembled into aggregate information entities. The component is "core" because it occurs in many different areas of industry or business information interaction. Core components together comprise business objects.

A business object is a conceptual object used to directly describe and represent a business concept or purpose. A business document may be considered a business object. A business object exists in one or more business domains, or industry sectors. A business object that exists in more than one business domain is known as a common business object. A business library containing common business processes is created by analyzing existing business objects used by many industries today in conjunction with the core library content.

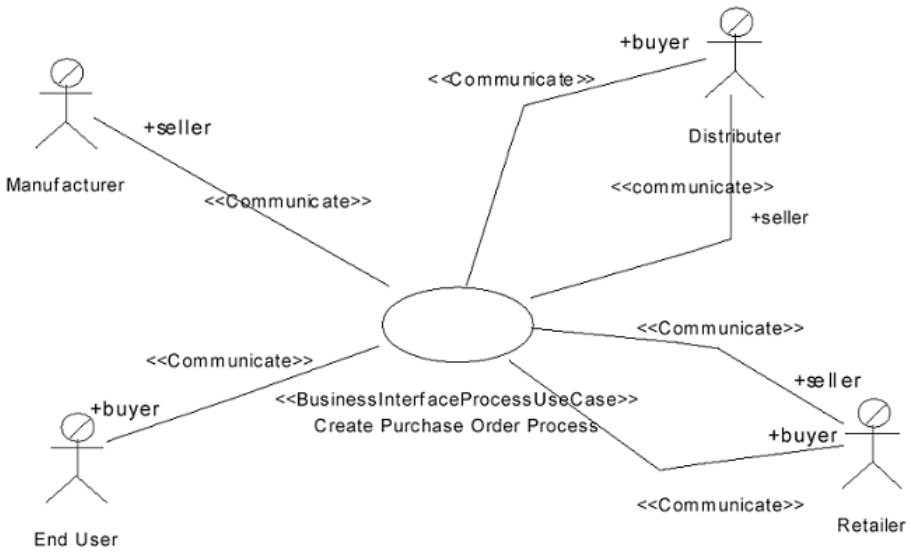
An example of an EPA core process is permit issuance. Permit issuance consists of different business actions (e.g., completing, submitting, and reviewing permits) that are applied to numerous industries. Information about permit issuance can be stored in a core library. Wastewater discharge permit issuance is an example of an EPA business process that contains the core process of permit issuance along with other processes. Information about wastewater discharge permit issuance can be stored in a business library. Permit issuance uses a permit business document (paper or electronic) that also is considered a common business object because it is applied across various industries. A permit consists of core components, such as facility information (identification number, address, etc.) and permit information (number, type, submitter name, etc.). These components can be stored in a core library. A wastewater discharge permit is considered a business object that contains the core components and non-core components, such as information about water sources, drainage, and flows per day.

Standardization is divided into three phases in the business operational view:

- Definition
- Analysis
- Design.

In the definition phase, the business problem is described using use-case diagrams and descriptions. A use case is a collection of possible sequences of interactions between a system and its users in relation to a particular goal. Use-case diagrams capture requirements from the perspective of how the user will use the system rather than from the perspective of the features that the system is required to incorporate. Figure 6-4 shows an example of a use-case diagram for creating a purchase order.

Figure 6-4. Example of a Use-Case Diagram



Source: ebXML.

If core library entries are available from an ebXML-compliant registry they are used during this phase, otherwise, new core library entries are created and registered in an ebXML-compliant registry.

The analysis phase reflects the business collaboration knowledge in the core library. Common business processes in the business library also may be referenced. In this phase, activity and sequence diagrams are created that describe the business processes. An activity diagram is a flow diagram that models business workflow, and a sequence diagram is used to model details in terms of objects and the passing of messages between objects. Figure 6-5 is an example of an order-processing activity diagram.

Figure 6-5. Example of an Activity Diagram

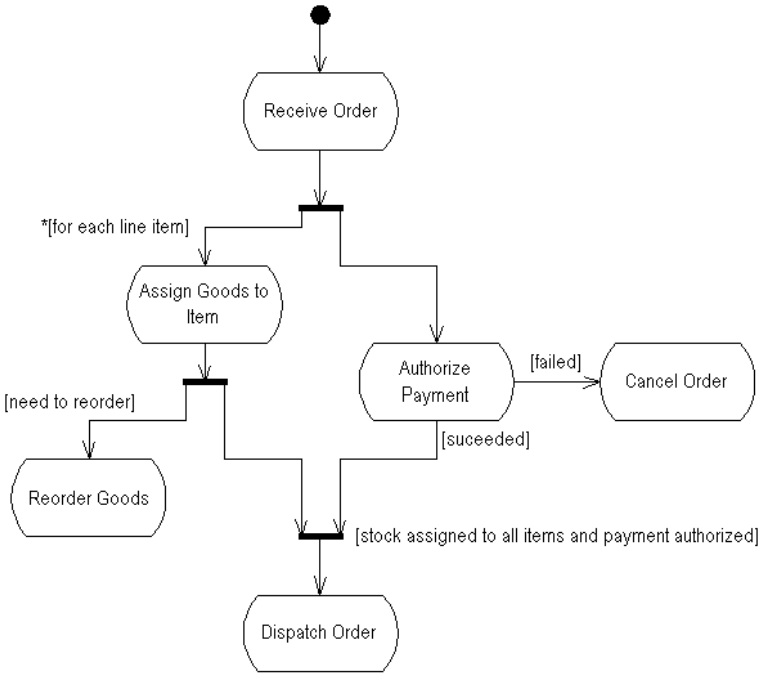
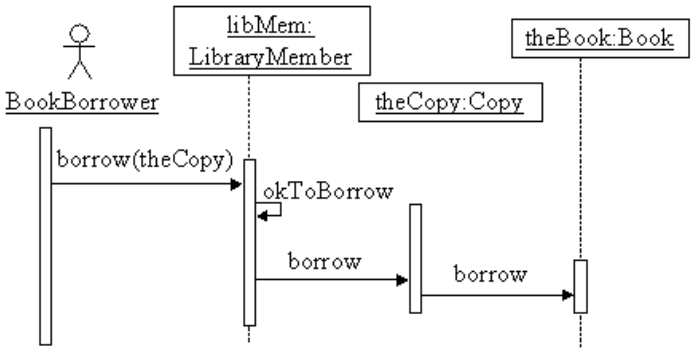


Figure 6-6 is an example of a sequence diagram that depicts the process of borrowing a book from a library.

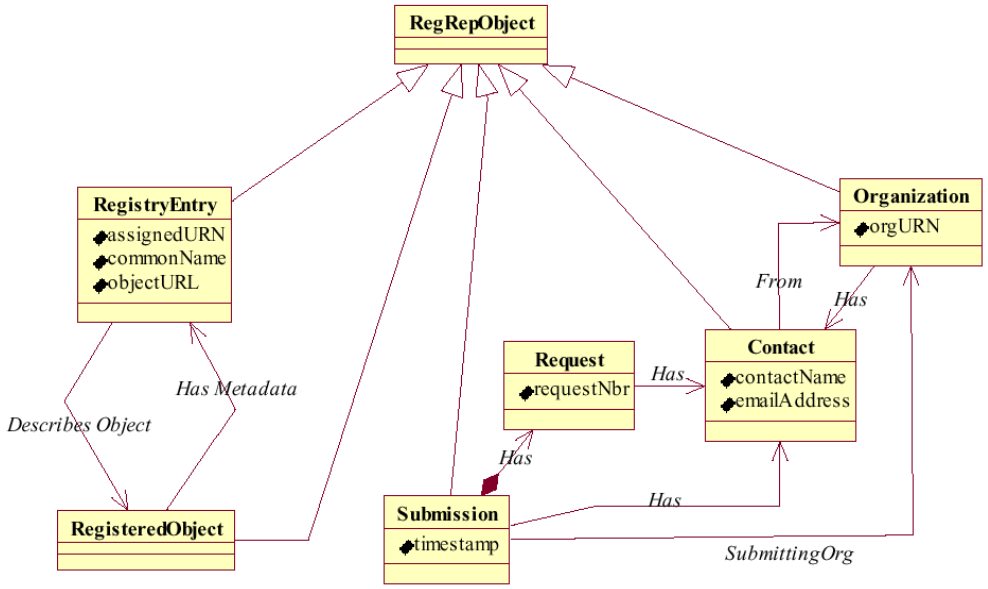
Figure 6-6. Example of a Sequence Diagram



The analysis phase reflects the business collaboration knowledge in the core library. Class diagrams also are created during the analysis phase to capture the associated business messages. A class diagram shows the existence of object classes and their relationships in the logical view of a system. The class diagrams used in this phase are freely structured data diagrams; no effort is made to force the application of object-oriented principles. These class diagrams are aligned with other models in the same industry and across other industries.

Figure 6-7 is an example of a class diagram.

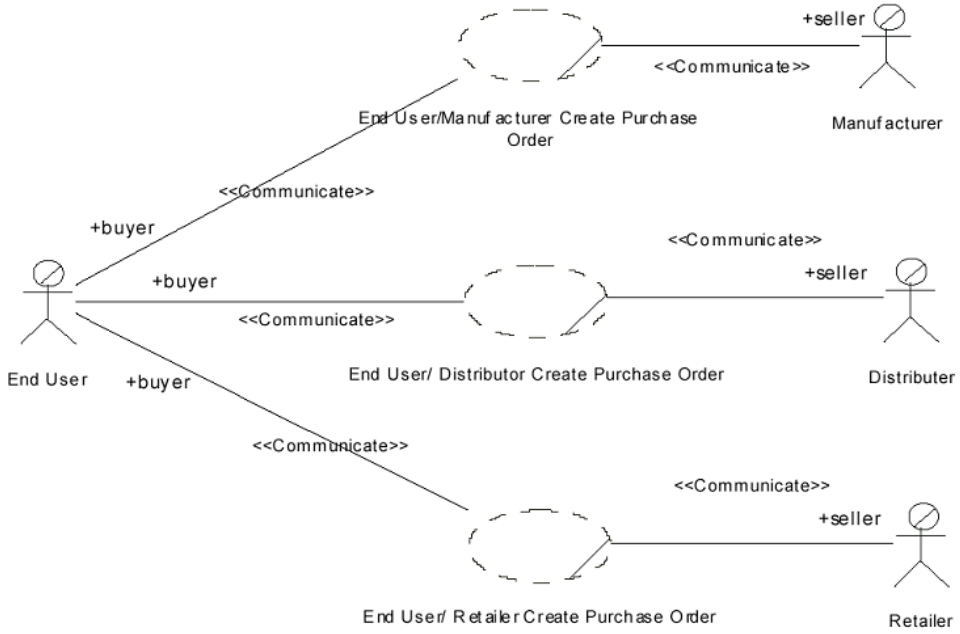
Figure 6-7. Example of a Class Diagram



Source: OASIS.

The last phase of standardization is the design phase. During this phase, object-oriented principles may be applied to generate collaboration diagrams and possibly a state diagram. A collaboration diagram illustrates aspects of business partner collaboration that are not applicable to the general use-case diagram created during the definition phase. Figure 6-8 is an example of a collaboration diagram for a purchase order creation process.

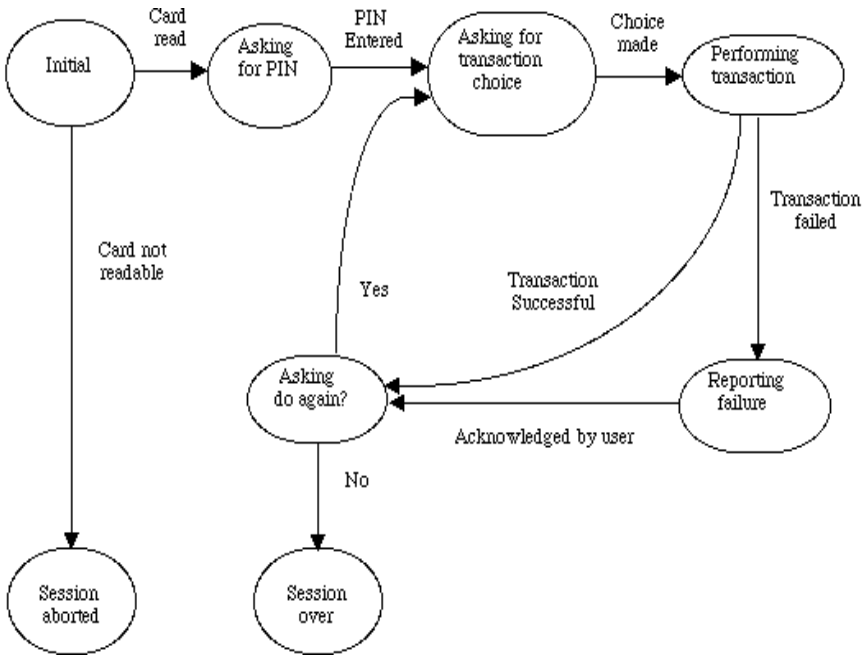
Figure 6-8. Example of a Collaboration Diagram



Source: ebXML.

Figure 6-9 is an example of a state diagram for an ATM transaction.

Figure 6-9. Example of a State Diagram



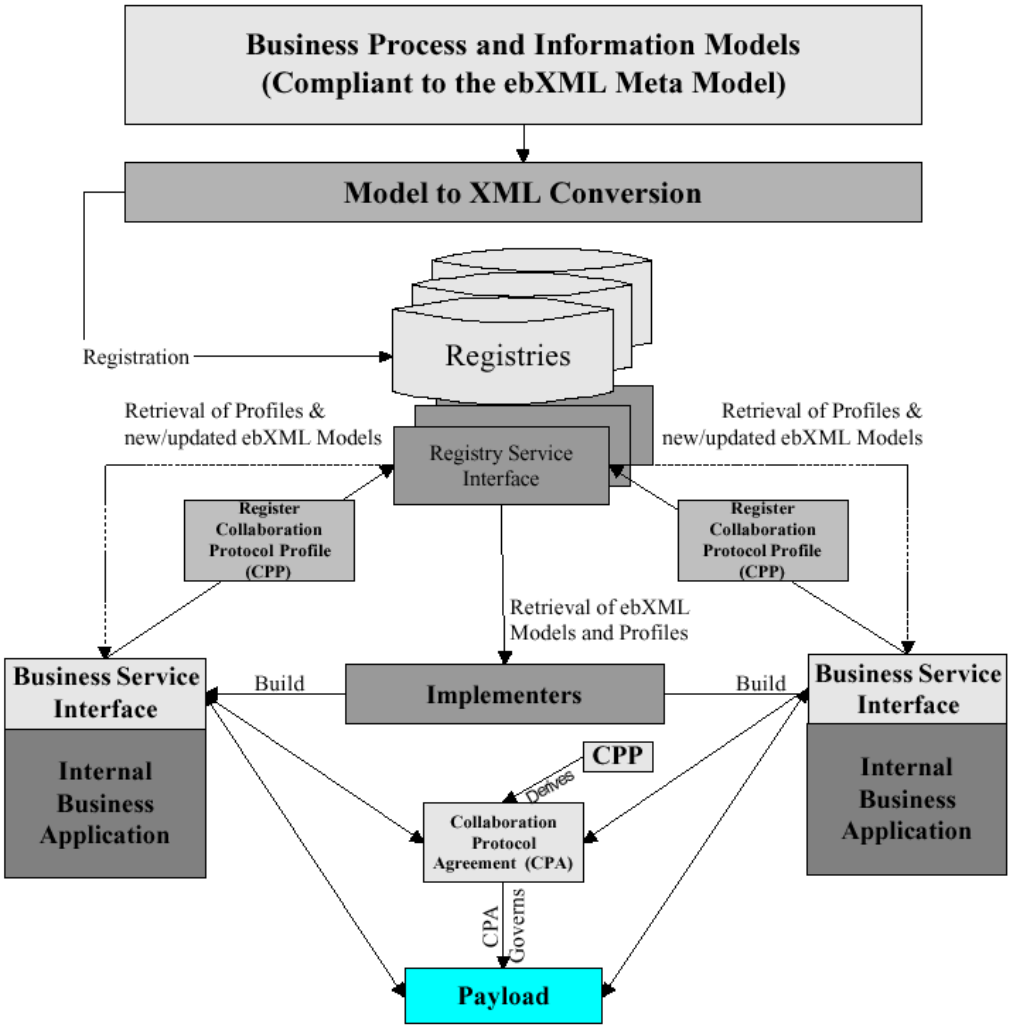
FUNCTIONAL SERVICE VIEW

The functional service view focuses on the mechanistic requirements for ensuring the operations described in the BOV. As the BOV expresses the users' requirements for achieving the common business goal, the FSV describes how the BOV is implemented using the selected technology. The FSV, therefore, centers on functional capabilities, business service interfaces, and protocols, including

- capabilities for implementation, discovery, deployment, and run-time scenarios;
- user interfaces;
- data-transfer infrastructure interfaces; and
- protocols for enabling interoperability of XML vocabulary deployments from different organizations.

Figure 6-10 shows the functional service view.

Figure 6-10. The Functional Service View



Source: ebXML.

As can be seen in the figure, the ebXML registry is an integral part of the ebXML framework. All of the content described in the BOV (business processes, core components, etc.) is held in an ebXML registry.

The implementation of the functional service view consists of three phases:

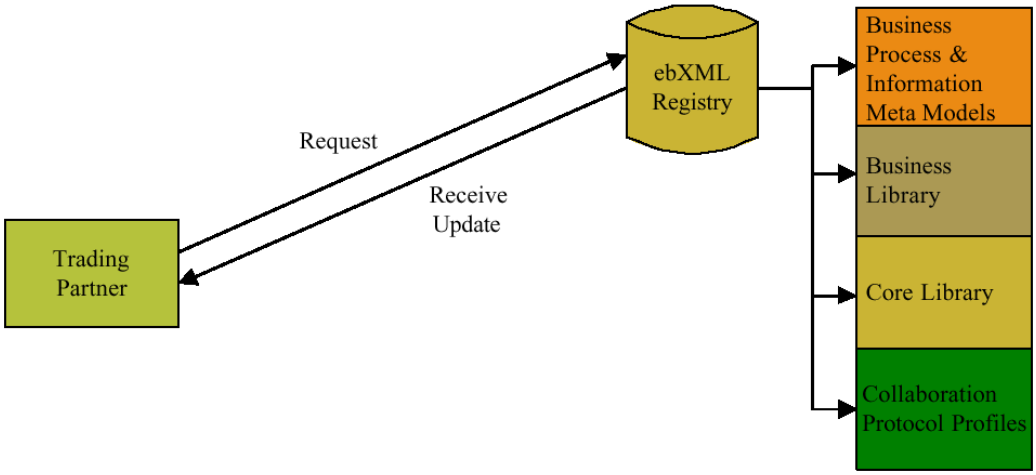
- Implementation
- Discovery and retrieval
- Run-time.

The implementation phase deals specifically with the procedures for creating an application of the ebXML infrastructure. During this phase, a trading partner wanting to engage in an ebXML-compliant transaction may do the following:

- Download a copy of the ebXML specifications
- Download the core library and business library
- Request other trading partners' business process information for analysis and review
- Submit its business process information to an ebXML-compliant registry.

Figure 6-11 shows the implementation phase.

Figure 6-11. The FSV Implementation Phase



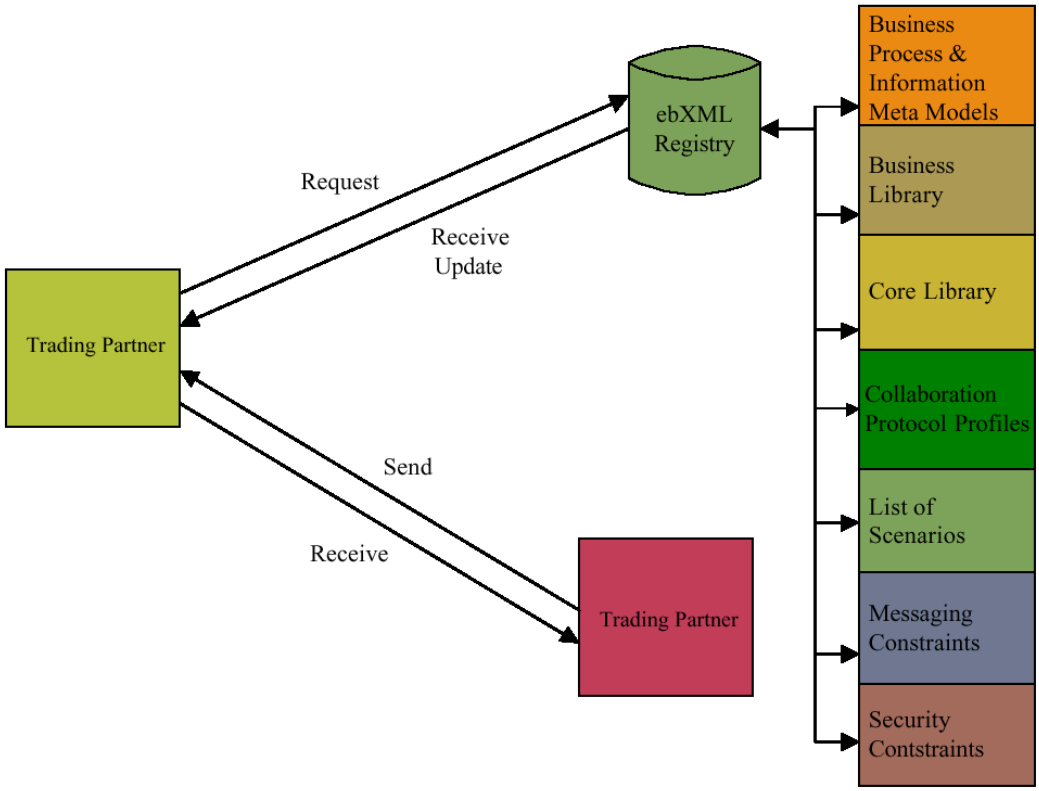
Source: ebXML.

After the trading partner has implemented an ebXML business service interface, it can begin discovery and retrieval. The discovery and retrieval phase covers all aspects of the discovery of ebXML-related resources, such as trading partner profiles, core libraries, and business libraries. A trading partner may do the following during this phase:

- Request the trading partner profile of another trading partner
- Request updates to core libraries and business libraries.

Figure 6-12 shows the discovery and retrieval phase.

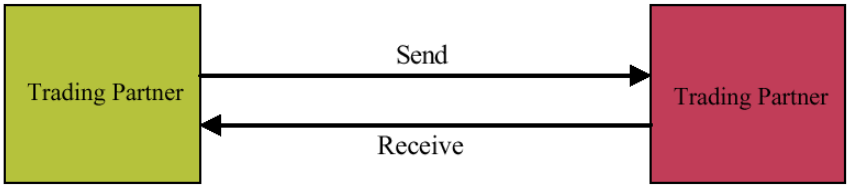
Figure 6-12. The FSV Discovery and Retrieval Phase



Source: ebXML.

During the run-time phase, ebXML messages are exchanged between trading partners using the ebXML messaging service. All calls to the registry during the run-time phase are considered reversions to the discovery and retrieval phase. Figure 6-13 shows the run-time phase.

Figure 6-13. The FSV Run-Time Phase



Source: ebXML.

RECOMMENDATIONS

Table 6-1 is a list of the recommendations, determinations, and issues about business processes that relate to EPA.

Table 6-1. Recommendations, Determinations, and Issues—Business Processes

ID No.	Topic	Type	Description	Time-frame	Comments
6.1	Business Scenarios	D	Determine business scenarios to be supported by using the XML registry	ST/NI	Reference examples on page 6-2
6.2	Business Process Integration	D	Determine existing business processes that will need to be integrated with the XML registry	ST/NI	
6.3	Business Process Creation	D	Determine new business processes that need to be created for using the XML registry	ST/NI	
6.4	User Access	D	Determine how users will access the XML registry	ST/NI	
6.5	Discovery	D	Determine how trading partners will "discover" the XML registry	MT	
6.6	Business Processes	R	Identify business processes for business operational view	LT	
6.7		R	Identify core processes for business operational view	LT	
6.8	Business Objects	R	Identify business objects for business operational view	LT	
6.9		R	Identify core components for business operational view	LT	
6.10	Functional Service View	D	Determine how much functionality in functional service view is applicable to EPA's needs	LT	

Chapter 7
Organizational Roles and Responsibilities

This chapter describes the roles and responsibilities of the organizations that may be involved in the daily operation of an XML registry. The information in this chapter can help EPA determine what organizations and individuals should be involved in the daily operation of an XML registry, and what the delineations of responsibilities between those organizations and individuals should be.

The concepts described here are based on the 11179 standard, and most have been adopted by the OASIS and ebXML models.

ROLES

The 11179 standard defines the following roles associated with a registry and its content:

- Registration authority
- Responsible organization
- Submitting organization.

Registration Authority

The RA exists so a single organization is responsible for establishing and maintaining information about the data elements of a particular community of individuals or organizations. An RA typically is established at an appropriate organizational level so the data needed for the organization's operations are authorized at that level.

An RA is responsible for

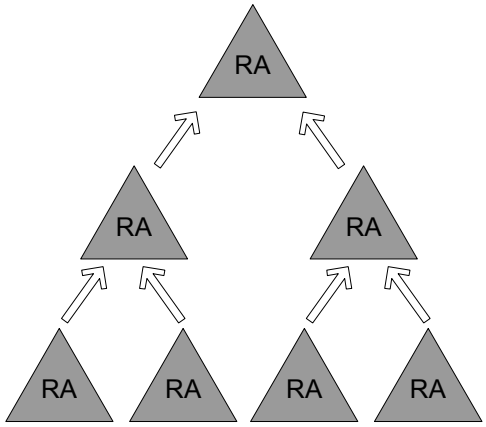
- establishing policies and procedures for using the registry,
- ensuring that registered data elements are reused and shared with and across functional elements and among members external to the organization,
- receiving and processing applications for the registration of data elements, and

- assigning appropriate registration and administrative statuses to data elements.

The 11179 standard requires that each organization that wants to become an RA have an internationally recognized organization code called a *Registration Authority Identifier* (RAI). This identifier is used with the data element identifier and the data element version to construct internationally unique identifiers for each data element in a registry.

The 11179 standard specifies the concept of a hierarchical RA, in which multiple Registration Authorities "report" to a single RA. For EPA, this could be implemented as a single RA for each state or county, each "reporting" to a higher level. This concept is shown in Figure 7-1.

Figure 7-1. Example of a Hierarchical Registration Authority



Source: LMI.

Responsible Organization

The RO exists to identify specific points of contact responsible for coordinating the identification, organization, and standardization of particular data elements in a data subject area throughout an organization. ROs typically are responsible for data elements in a data subject area (such as "employee") that cut across multiple operational functions, such as human resources, sales, marketing, and production.

An RO is responsible for

- advising, at the request of the RA, about the semantics, name, and permissible values of data elements submitted for registration;
- notifying the RA about amendments to the data elements assigned to the RO; and

- ensuring the quality of the metadata of the data elements assigned to the RO.

Submitting Organization

The SO authorizes individuals or organizational elements from among those responsible for designing, developing, implementing, or operating data resources of an RO's subject area to propose data elements for registration. A submitting organization may be

- a data administrator responsible for the data of one or more systems in a functional area,
- a data system manager,
- a data modeler,
- a data designer,
- a database administrator responsible for operating or developing one or more databases, or
- a functional area business manager.

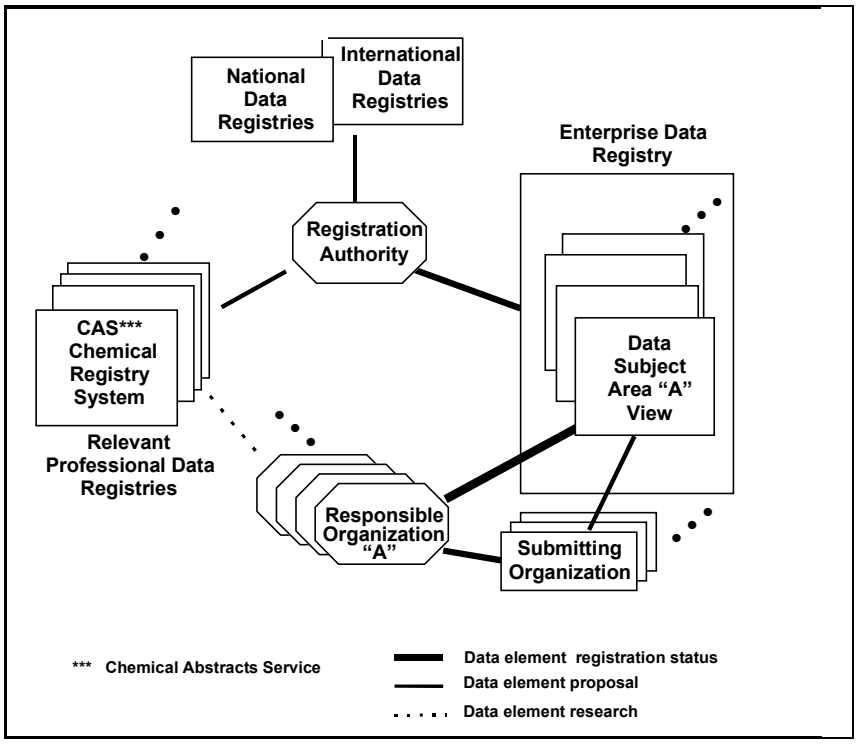
An SO is responsible for the completeness of data registration proposals. In particular, an SO is responsible for

- identifying and documenting data elements appropriate for registration, and
- submitting proposals for registering data elements to the appropriate RO.

Figure 7-2 illustrates a fully mature registration environment. The figure depicts the relationship between SOs, ROs, the registry itself, and external registries.

Figure 7-2 also shows that one of the roles of an RO is be a focal point for SOs proposing data elements for registration. Designated SOs support each responsible organization with documenting metadata required for each registration. Each RO operates under the policies and procedures established by the RA.

Figure 7-2. Registration Environment



Source*:* ISO/IEC.

ROLE USE

The OASIS and ebXML models have, in general, adopted the 11179 standard for the roles and responsibilities described above. The OASIS model requires an organization that wishes to become an SO to apply to a recognized RA, identifying itself and providing general information, such as its address and parent organization. This information ensures that the RA is known to the SO and responsible contacts in that organization before accepting follow-on submissions from the SO. If the RA isn't known to the SO, the RA will reject the submission and return an error message to the SO.

The ebXML model does not require SOs to register before submitting content. The ebXML model states that an organization may submit content as long as the content is digitally signed by an approved certification authority.

Neither the OASIS nor the ebXML model references obtaining or using an RAI.

RECOMMENDATIONS

Table 7-1 is a list of the recommendations, determinations, and issues about RAs, ROs, and SOs that relate to EPA.

Table 7-1. Recommendations, Determinations, and Issues—Organizational Roles and Responsibilities

ID No.	Topic	Type	Description	Time-frame	Comments
7.1	Registration Authority	D	Determine criteria for identifying and establishing registration authority, for both immediate and future selection	ST/NI	
7.2		D	Determine who the registration authority will be according to established criteria	ST/NI	
7.3		D	Determine responsibilities of registration authority	ST/NI	
7.4		D	Determine what information an RA needs to provide to the XML registry for record keeping	ST/NI	
7.5		I	Will Registration Authority Identifiers be required of RAs?	ST/NI	
7.6		I	Will hierarchical RAs be used?	LT	
7.7	Responsible Organization	D	Determine criteria for identifying and establishing responsible organizations, for both immediate and future selection	ST/NI	
7.8		D	Determine who ROs will be according to established criteria	ST/NI	
7.9		D	Determine responsibilities of ROs	ST/NI	
7.10		D	Determine what information an RO needs to provide to the XML registry for record keeping	ST/NI	
7.11	Submitting Organization	D	Determine criteria for identifying and establishing submitting organizations, for both immediate and future selection	ST/NI	
7.12		D	Determine who SOs will be according to established criteria	ST/NI	
7.13		D	Determine responsibilities of SOs	ST/NI	
7.14		D	Determine what information an SO needs to provide to the XML registry for record keeping	ST/NI	

Chapter 8
Associations and Classifications

This chapter describes the concepts of associations and classifications in a registry model, and gives examples of their use. Associations describe relationships between registered objects, and classifications allow registered objects to be grouped on the basis of characteristics that the objects have in common. The information in this chapter can help EPA determine the associations and classifications that should be used in the XML registry.

ASSOCIATIONS

ebXML and OASIS

An association represents a relationship between two registered objects. For example, an association may exist between an XML document and its corresponding schema that signifies that the XML document validates to the schema. An association is, therefore, a registry entry that signifies a connection between two registered objects. Each association entry contains an attribute known as an *association role* that describes the type of association between the two registered objects. In the example above, the association role is "Validates To."

A registered object can have any number of associations. Table 8-1 shows the association roles that may exist in a registry, and the XML model that specifies them. Several of the associations listed below reference the concept of a *registry package.* A registry package is a set of registered objects that can be operated on as a group. For instance, all registered objects in a package can be deleted at one time.

An SO may provide an association for an object when it is submitted, or for an existing (registered) object. For example, a state may want to submit a white paper that describes the functionality of a system with which a schema is associated. The state may submit the white paper as external data and an "Externally Identifies" association that relates the white paper to the schema.

Table 8-1. Association Roles in a Registry

Role	Description	Example	Model
Contains	A registry package contains one or more registered objects	N/A	ebXML and OASIS
Contained By	A registered object is contained by a registry package	N/A	ebXML
Extends	A registered object inherits from or specializes another registered object	No other information available	ebXML
Externally Links	A registered object externally links another registered object	A registered object (such as a schema) external links a data item (such as a user manual)	ebXML
Externally Identifies	A registered object externally identifies (is externally linked by) another registered object	A data item (such as a user manual) is eternally linked by a registered object (such as a schema)	ebXML
Has Member	A registry package has a registered object as a member	N/A	ebXML
Implements	A registered object implements functionality defined by another registered object	No other information available	ebXML
Instance	A registered object is an instance of another registered object	No other information available	ebXML
Related To	A registered object is related to another registered object	An external data item (such as a user manual) is related to an existing registered object (such as a schema)	OASIS
Replaces	A registered object replaces another registered object	A new, upward-compatible version of a registered object replaces an existing version. The old version is no longer available	ebXML
Replaced By	A registered object is replaced by another registered object	An existing version of a registered object is replaced by a new, upward-compatible version. The old version is no longer available.	ebXML and OASIS
Requires	A registered object requires the presence of another registered object. Therefore, the required registered object must be retrieved before the other registered object can be processed or used.	An XML element requires the presence of some other XML element or entity that it references	OASIS
Supersedes	A registered object supersedes another registered object	A new version of a registered object replaces an existing version. The old version is still available.	ebXML
Superseded By	A registered object is superseded by another registered object	An existing version of a registered object is replaced by a new version. The old version is still available.	ebXML and OASIS
Uses/Used By	A registered object uses, or is used by, another registered object in some manner	No other information available	ebXML
Validates To	A registered object validates to the specification provided by another registered object	An XML document validates to an XML schema	OASIS

11179

The 11179 registry model implements some of the associations shown above through its version control mechanisms. For instance, the "Superseded By" association is represented by a scenario in which a data element has been updated and the new version of that data element supersedes the previous version. The "Replaced By" association is represented by a scenario in which a data element has been updated, and the previous version of the data element is retired.

CLASSIFICATIONS

ebXML and OASIS

A classification is an arrangement or division of objects into groups that are based on characteristics that the objects have in common, e.g., origin, composition, structure, application, or function. A classification depends on a pre-existing specification of a hierarchy of values, names, and codes called a *classification scheme*. A classification scheme is itself a registered object. An example of a classification scheme is the 5-level hierarchy North American Industry Classification System (NAICS). For example, NAICS code *11114* represents a node at the fourth level of the NAICS classification tree. Table 8-2 shows that the code is really a sequence of 4 values.

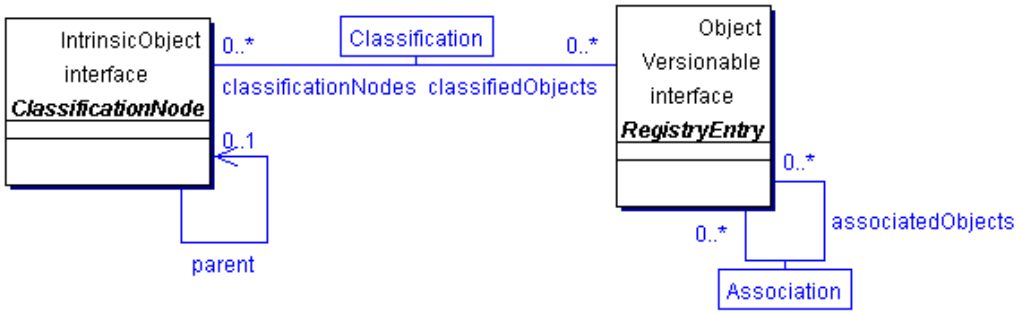
Table 8-2. NAICS Code Structure

Value	Represents
11	Agriculture, forestry, fishing, and hunting
1	Crop production
1	Oilseed and grain farming
4	Wheat farming

Other examples are the 3-level International Press Telecommunications Council (IPTC) scheme for classifying news articles, and the 7-level binomial nomenclature taxonomy used by biologists to classify living things (kingdom, phylum, class, etc.).

A classification is a reference to a single node of a classification scheme. A classification is, therefore, a specialized form of association as shown in Figure 8-1.

Figure 8-1. ebXML Classification Scheme



Source: ebXML.

For example, a schema that is used for lab reporting of microbiological analyte test results for drinking water may be classified as follows:

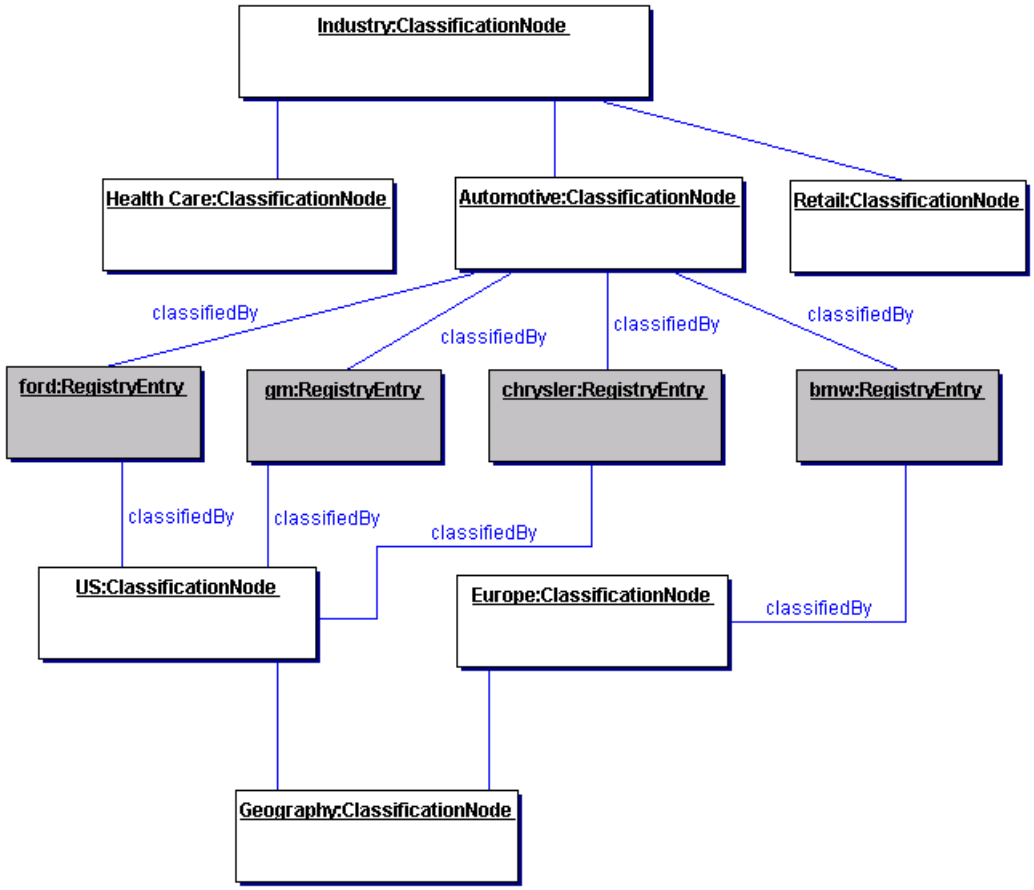
- *Major Purpose:* Reporting
- *Type of Reporting:* Test results
- *Subject of Testing:* Drinking water
- *Type of Testing:* Analyte
- *Type of Analyte:* Microbiological.

A registered object can have any number of classifications—i.e., a registered object may be classified in more than one classification scheme. When an object is added to a registry, typically it will be classified by the SO according to one or more previously registered classification schemes. In the future, any organization may submit a classification for a registered object according to classification schemes they are most comfortable with, even if they were not the original submitting organization for the object. For example, a state may submit a new classification for the schemas shown in the above classification example according to the origin of the drinking water (e.g., surface or ground) rather than the type of analytes that were tested for.

An organization also may submit a classification scheme as a registered object. This may occur if the organization intends to submit a new object according to the classification scheme (i.e., the classification scheme must first exist in the registry before the object is submitted). This also may occur if the organization intends to submit a new classification for a registered object after a classification scheme is submitted.

Figure 8-2 depicts registered objects classified by multiple classification schemes. Several automobiles are classified according to two separate classification schemes—one being industry, the other geography. The dark nodes are not part of the classification scheme—rather, they are the registered objects that are being classified.

Figure 8-2. Automotive Classifications



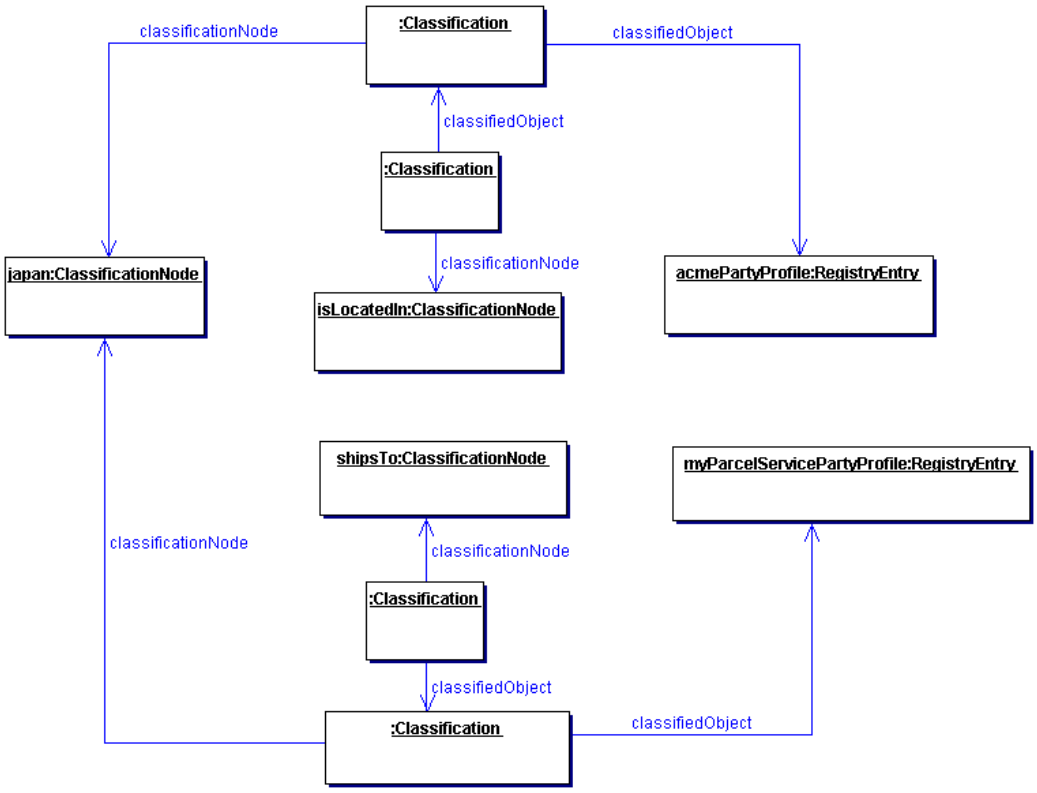
Source: ebXML.

The first level of this classification scheme is industry. The first level then is subdivided into different industries, such as health care, automotive, and retail. The registered objects are, therefore, classified under the automotive industry. The registered objects also are classified according to geography, to depict the geographic origin of the automobile model. In this case, there are two choices—Europe and U.S. As shown in the figure, each automobile is associated with a single node of both classification schemes.

The ebXML model uses an additional concept known as *context-sensitive classification*, in which a classification may be associated with multiple contexts and an additional classification node is used to clarify the context for each case. For example, if a geography scheme is used for classifying a trading partner profile in

a registry, a trading partner profile may be classified by the Japan classification node of the geography classification scheme. Without a specific context, the classification could mean that the organization is in Japan, or that it ships products to Japan. To clarify this, the classification may be associated with an additional classification node called “isLocatedIn” or “shipsTo” that gives the context for the classification, as shown in Figure 8-3.

Figure 8-3. Context-Sensitive Classification



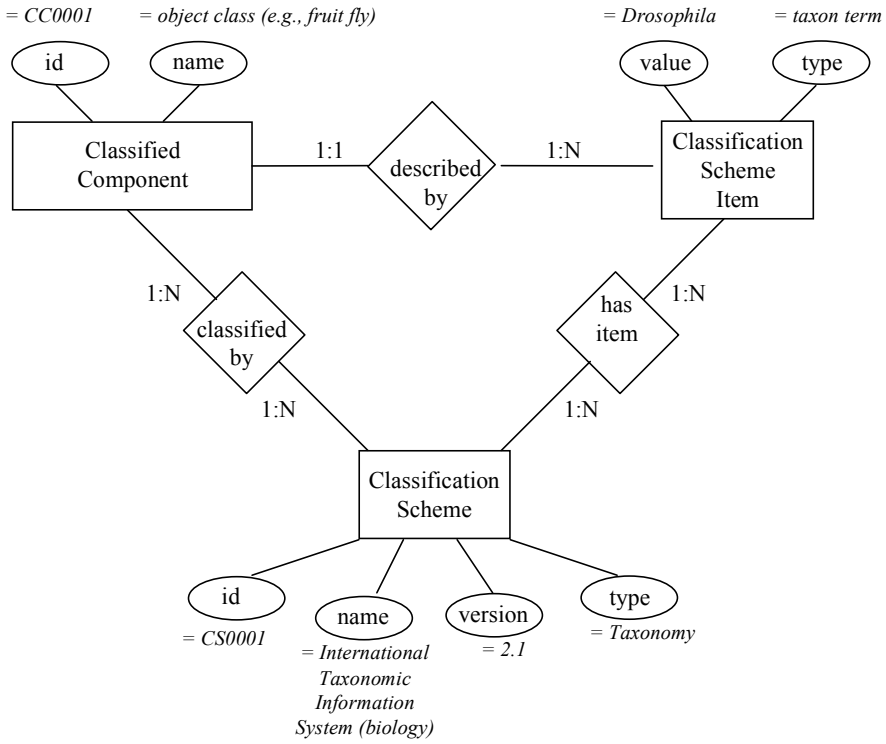
Source: ebXML.

11179

The 11179 registry model also implements classification functionality and supports classification by multiple schemes. The 11179 registry model does not support the concept of context-sensitive classification.

The 11179 registry model provides an example in which a registered object is classified according to a classification scheme using the binomial nomenclature taxonomy. The classified component (registered object) is associated with a classification scheme item (classification node) as shown in Figure 8-4.

Figure 8-4. An Example of Binomial Nomenclature Taxonomy



Source: ISO/IEC.

RECOMMENDATIONS

Table 8-3 is a list of the recommendations, determinations, and issues about associations and classifications that relate to EPA.

Table 8-3. Recommendations, Determinations, and Issues—Associations and Classifications

ID No.	Topic	Type	Description	Time-frame	Comments
8.1	Association Roles	R	Implement simple association roles	ST/NI	
8.2		D	Determine what a “simple” association role is	ST/NI	
8.3		R	Implement more complex association roles	MT	
8.4		D	Determine what a “complex” association role is	MT	
8.5	Classification Schemes	R	Implement simple classification schemes	ST/NI	
8.6		D	Determine what a “simple” classification scheme is	ST/NI	
8.7		R	Implement more complex classification schemes	LT	
8.8		D	Determine what a “complex” classification scheme is	LT	
8.9	Number of Classifications per Registered Object	I	Should there be a limit to the number of classifications a registered object can have?	ST/NI	
8.10	Submission of Classifications	I	Should users be allowed to submit new classifications for registered objects?	MT	
8.11	Submission of Classification Schemes	I	Should users be allowed to submit classification schemes as registered objects?	MT	
8.12	Multiple-Scheme Classification	I	Should a registered object be allowed to be classified by multiple classification schemes?	ST/NI	
8.13	Context-Sensitive Classification	I	Should context-sensitive classification be used?	MT	

Chapter 9
Registry Services

This chapter describes the services of the registry models that are described in this report. The information in this chapter can help EPA define what types of operations the XML registry should support. The term *registry services* means operations that are performed on the metadata and contents in a registry, such as

- registration of an object,
- query of an registered object, and
- submission of a classification scheme.

We discuss additional operations and functionality in detail below.

REGISTRY SERVICES INTERFACE

A registry services interface is used for the automatic or human-initiated execution of functions, such as registration of new objects, modification or removal of registered objects, and retrieval of registered objects, through browse or query functionality. Because each model interfaces differently, we describe the functionality for each separately.

OASIS

The following are examples of requests allowed in the OASIS registry services model:

- *Register Object*: Register a new object in the XML registry. The object itself may or may not be included with the request.
- *Add Association*: Add an association for two registered objects.
- *Delete Association*: Remove an existing association.
- *Add Classification*: Add a new classification for a registered object.
- *Delete Classification*: Delete an existing classification for a registered object.
- *Define Classification Scheme*: Define a new scheme that can be used to classify registered objects.
- *Modify Classification Scheme*: Modify an existing classification scheme to add or delete levels or nodes.

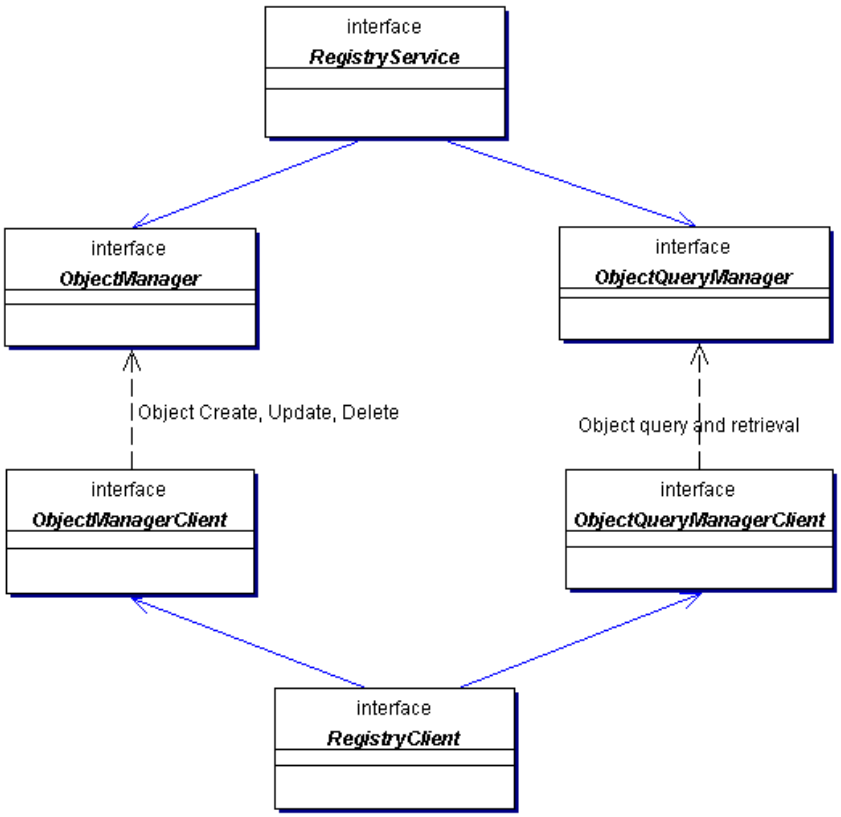
- *Define Registry Package*: Define a single, complete package of related registry entries.
- *Modify Registry Package*: Modify the contents of an existing registry package.
- *Modify Registry Entry*: Modify one or more of the metadata attributes of an existing registry entry, including object identifier, object type, stability, or registration status.

A request may affect one or more registry entries. For example, a request to supersede a registered object with a new registered object will affect the registry entries for both registered objects. The Impact class of the OASIS registry services model captures such effects. All requests are sent to the registry in XML documents.

ebXML

The ebXML registry services model consists of a registry and registry clients. Registry clients communicate with the registry using the ebXML messaging service. Several interfaces exist in this model, as shown in Figure 9-1.

Figure 9-1. ebXML Registry Services Model



Source: ebXML.

Each of these interfaces also may be thought of as a service. The RegistryService interface shown above is the principal registry interface. It is the mechanism through which all registry service requests are carried out. The ObjectManager interface is the interface through which requests similar to those outlined in the OASIS description above are carried out. The ObjectQueryManager interface is the interface through which query requests are carried out. The ObjectManager-Client interface is the mechanism that a registry client uses to connect with the ObjectManager interface. The ObjectManagerClient interface will send service requests to the ObjectManager service, and receive responses (both error and accepted) from this service. The ObjectQueryManagerClient interface is the mechanism that a registry client uses to connect with the ObjectQueryManager interface. The ObjectQueryManagerClient will send service requests to the ObjectQueryManager service, and receive responses (both error and accepted) from this service.

As with the OASIS model, all requests are sent to the registry as XML documents. The ebXML model does not reference effects the same way as the OASIS model. Rather, it has a registry audit trail capability.

In the future, the ebXML model will enable using packages to group objects in the registry so operations can be done on an entire package of objects.

11179

The 11179 registry model does not address registry services. For information retrieval standards, users should refer to standards, such as the following:

- *ISO 8777*: Defines commands and services for searching and retrieving data stored in bibliographic databases.
- *ANSI Z39.50*: Information Retrieval Application Service Definition and Protocol Specification, for online library catalogs and other structured data sets.

QUERY SERVICES

OASIS

The OASIS registry services model also allows query requests, such as the following:

- *GetRegisteredObject*: Obtain one or more registered objects. Objects may be queried by object identifier, association, classification, description, alternate name, Submitting Organization, etc.

- *GetRegistryEntry*: Obtain selected metadata for one or more registry entries. Registry entries may be queried by the same criteria as for "GetRegisteredObject."

The OASIS registry services model also supports queries on the following registry contents:

- *Contact*: Locate a contact by name, organization, e-mail address, etc.
- *Request*: Locate a request by date or time, number, type, etc.
- *Impact*: Discover the effects on a given registry entry by different supported requests.
- *Organization*: Locate the information for an organization that is known to the XML registry.

Currently, the OASIS registry services model does not support ad hoc queries.

ebXML

The ebXML registry services model includes support for three types of queries:

- Browse and drill down
- Filtered
- Ad hoc

We discuss each of these types of queries below.

BROWSE AND DRILL DOWN QUERY

In the browse and drill down query, the user browses registry content according to classification schemes by using a GUI tool known as a registry browser. The user then selects a registered object and "drills down" to view the object details.

FILTERED QUERY

In the filtered query, a user can query on registry entries and registered objects by using "filters" to narrow down the query results. Two types of requests are possible:

- *ReturnRepositoryItem*: Similar to OASIS "GetRegisteredObject" request.
- *ReturnRegistryEntry*: Similar to OASIS "GetRegistryEntry" request.

The following "ReturnRegistryEntry" example filters on item status to obtain registry entries for all objects whose status is "registered":

```
<ReturnRegistryEntry>
    <RegistryEntryQuery>
        <RegistryEntryFilter>
            Status EQ "Registered"
        </RegistryEntryFilter>
    </RegistryEntryQuery>
</ReturnRegistryEntry>
```

Note: A single registry entry can be returned by including a UID as RegistryEntryFilter.

The following "ReturnRegistryEntry" example filters on item object type to obtain registry entries for all registered objects that are registry packages:

```
<ReturnRegistryEntry>
    <RegistryEntryQuery>
        <RegistryEntryFilter>
            objectType EQ "RegistryPackage"
        </RegistryEntryFilter>
    </RegistryEntryQuery>
</ReturnRegistryEntry>
```

The "ReturnRepositoryItem" request is similar to the "ReturnRegistryEntry" request, but returns the actual registered object rather than the registry entry.

AD HOC QUERY

The ad hoc query uses a SQL-based query language for more complex queries than are possible using the other query types. The following example returns all registry entries whose names contain the word "Acme" and whose version is greater than 1.3:

```
SELECT id FROM RegistryEntry WHERE
    name LIKE '%Acme%' AND
    majorVersion >= 1 AND
    (majorVersion >= 2 OR minorVersion > 3);
```

11179

The 11179 registry model does not address query services at this time.

RECOMMENDATIONS

Table 9-1 is a list of the recommendations, determinations, and issues about registry services that relate to EPA.

Table 9-1. Recommendations, Determinations, and Issues—Registry Services

ID No.	Topic	Type	Description	Time-frame	Comments
9.1	Registry Services	R	Implement essential registry services	ST/NI	
9.2		D	Determine what an “essential” registry service is	ST/NI	
9.3	Registry Services	R	Implement non-essential registry services	MT	
9.4		D	Determine what a “non-essential” registry service is	MT	
9.5	Query Services	R	Implement simple queries	ST/NI	
9.6		D	Determine what a “simple” query is	ST/NI	
9.7		I	Should ad hoc queries be implemented?	MT	
9.8		I	Should complex queries be implemented?	LT	

Chapter 10
Registry Administration

This chapter describes functions of the administrative aspects of a registry. These functions include

- submission to the registry,
- object life cycle,
- object stability,
- version control, and
- audit trail.

The information in this chapter can help EPA define the administrative functionality that the XML registry should support.

We discuss the features for each registry model separately.

SUBMISSION TO THE REGISTRY

OASIS

The OASIS model defines a submission as "a collection of requests, in the form of a message, sent from a Submitting Organization to a registry." A request is an invocation of a service in the registry that results in some action being performed. The action may be the registration of a new object, or a request to operate on the metadata for a registered object (for instance, to add a new classification for a registered object). Examples of requests are listed below:

- Add classification
- Add association
- Delete alternate name
- Delete description
- Register Submitting Organization
- Add external data.

In the OASIS model, submissions are grouped into packages. Therefore, a submission package is a collection of submissions, which in turn is a collection of requests. A request in which a new object is registered may include the object itself or it may simply include an object identifier (i.e., the object may reside outside the repository).

ebXML

In the ebXML model, a submission does not contain requests but rather object metadata. As with the OASIS model, the object may or may not be included with the submission.

11179

In the 11179 registry model, the term *submission* pertains to the registration of a data element or administered component.

OBJECT LIFE CYCLE

The term *life cycle* describes the phases that an object in a registry can pass through from the time that it is submitted to the registry to the time that it is removed. Because each of the models specifies different object life cycle functionality, we describe the functionality for each separately.

OASIS

The following registration statuses of the OASIS model represent the life cycle of an object:

- *Submitted*: Object has been submitted to registry.
- *Under review*: Submitted object is under review by panel.
- *Registered*: Object is registered.
- *Deprecated*: Object is deprecated.
- *Withdrawn*: Object is withdrawn.
- *Expired*: Registration of object has expired.

Once an object is submitted to a registry, it does not immediately become a "registered" object (although we use this as the general term for a submitted object throughout this report). First, the object must be reviewed by a group of reviewers. The specifications for the review and the criteria for selecting reviewers are not part of the OASIS specification. A deprecated status is used when the SO indicates that the registered object will soon be replaced or withdrawn. A certain

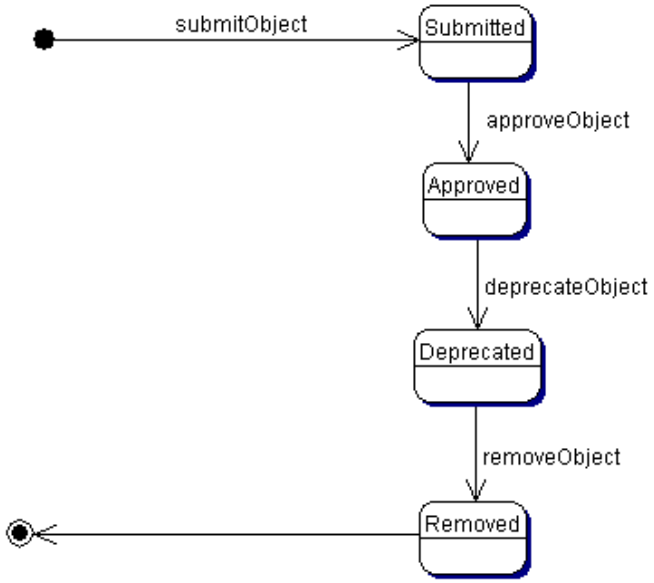
period is allowed before this takes place—the RA's policy usually determines this period. No other metadata changes to the object are allowed during this time. A withdrawn status indicates that the registered object is no longer available, although its metadata will remain in the registry. This status also can be assigned only at the request of the SO.

Each registered object has an expiration date that the RA assigns when the object is registered, using a date suggested by the SO. The RA may accept this expiration date, or may use its own expiration date based on organizational policy. The SO may change the expiration date during the life of the registered object by submitting a new expiration date to the registry. This is known as *reaffirming a registered object.* The new date can be overridden by the RA according to its policy. If the expiration date is reached without reaffirmation by the SO, the RA may initiate an expiration action.

ebXML

Figure 10-1 shows the typical life cycle of an object in an ebXML registry.

Figure 10-1. Life Cycle of an ebXML Registered Object



Source: ebXML.

In the ebXML model, once an object is deprecated no new references to that object (i.e., associations, classifications, external data) can be submitted. However, existing references to that object continue to function normally. The ebXML model, unlike the OASIS model, does not specify a period that an object must be deprecated before it is removed. As with the OASIS model, an object must be deprecated before it can be removed from the registry.

A “removeObject” request may result in the deletion of only the registered object, or both the registry entry and registered object. Deletion of only the registered object allows existing references to the registry entry to remain valid. The ebXML model also states that all references to a registry entry must be removed before the registry entry can be removed from the registry.

11179

The 11179 standard specifies both a registration status and an administrative status for a data element. An administrative status is a designation of the position in the processing life cycle of an RA for handling registration requests. A registration status is a designation of the position in the registration life cycle of an RA. The values for administrative status are assigned by a Registration Authority with the required values to facilitate the management of its registry, such as

- received,
- draft,
- rejected,
- submitted for certification,
- processed, and
- being promoted.

The 11179 standard specifies the following registration statuses for data elements:

- *Incomplete*: Data element does not contain all mandatory attribute values—e.g., name, datatype, minimum and maximum size, permissible values.
- *Recorded*: Data element contains all mandatory attribute values, but may not meet the quality requirements specified in other parts of the 11179 standard.
- *Certified*: Data element has met the quality requirements specified in the 11179 standard.
- *Standardized*: The RA establishes the data element as one that is preferred for use in data interchange and in new or updated applications.
- *Retired*: Data element is no longer in use after having been marked as “phased out” for a period of time as prescribed by the RA.

Separating data elements that are being reviewed from those that are in later life cycle stages may be desirable to ensure that the elements are not used

prematurely. Therefore, two separate registries are possible, one an "official" registry and the other a "work-in-progress" registry. Data elements that are being reviewed remain in the "work-in-progress" registry with an administrative status but no registration status. Once these data elements reach the "official" registry, they have a registration status but no administrative status. Data also may be separated logically by using views.

Table 10-1 is an example of the registration process for a data element in an 11179 registry. The example is a mixture of registration statuses and administrative statuses, with nine status categories. The table describes the actions of the organizations involved in the registration process, and the status that results from each action.

Table 10-1. Example of the 11179 Standard Registration Process

Action	Registration status
Submitting organization proposes data element for registration; responsible organization has not yet verified registry entry as complete.	Draft
RO verifies registry entry as complete; RO has not yet verified registry entry as conforming with acceptable quality requirements.	Provisionally recorded
RA verifies registry entry as complete.	Recorded
RO verifies registry entry as conforming with acceptable quality requirements.	Provisionally certified
RA verifies registry entry as conforming with acceptable quality requirements.	Certified
RA confirms the semantic uniqueness of the data element in the registry.	Provisionally standardized
RO completes standardization review of registry entry.	Standardized

Source: ISO/IEC.

Once a data element is no longer needed, collected, or used by current activities or legacy systems, it is placed in a "phasing out" status and the RO reviews it for retirement. Once the RO completes the retirement review, the data element is placed in a "retired" status.

The RA can reject a submitted data element. This would occur, for example, if essential information is not provided for the data element. In such cases, the RA returns the application to the SO with a clear statement of the reasons for rejecting the submittal. If possible, the RA gives constructive advice so the SO can make the application acceptable.

The above scenario mixes registration and administrative statuses. The following statuses are registration statuses:

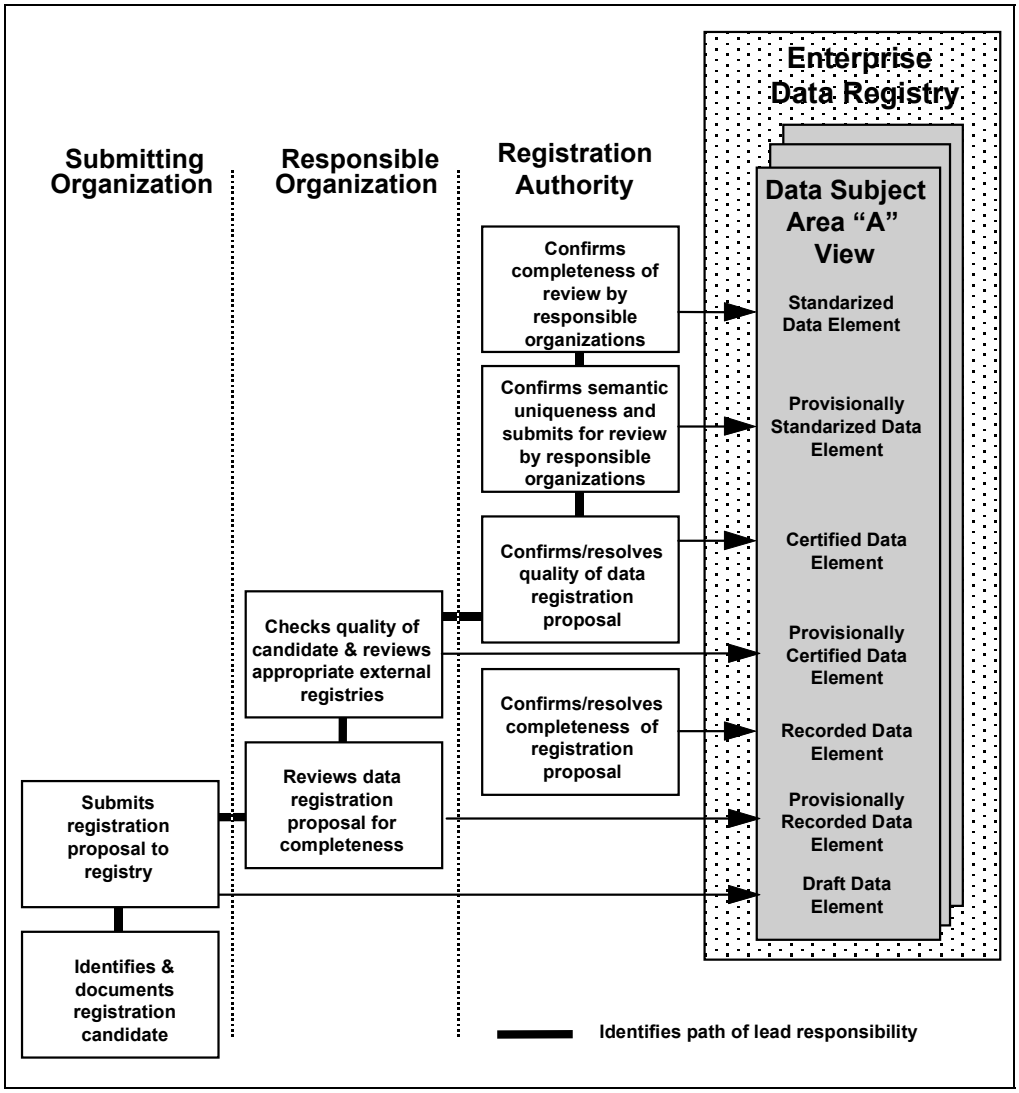
- Recorded
- Certified
- Standardized
- Retired.

The following are administrative statuses that reflect administrative processing stages between the registration statuses:

- Draft
- Provisionally recorded
- Provisionally certified
- Provisionally standardized
- Phasing out.

Figure 10-2 summarizes the entire process described above.

Figure 10-2. The Review Process



Source: ISO/IEC.

OBJECT STABILITY

The ebXML and OASIS models both reference the concept of the stability of a registered object. Stability is the likelihood that the registered object will change in the future. The SO gives this attribute when the object is registered, and may change the attribute during the object life cycle. The following are valid values:

- *Static*: Registered object will not change before expiration.
- *Dynamic*: Registered object may change at any time.
- *Compatible*: Registered object may be replaced only by an upward compatible object.

If an XML document that validates to a given schema also will validate to another schema, the two schemas are said to be *compatible*.

VERSION CONTROL

OASIS

In the OASIS model, the RA does not strictly control versions—i.e., the RA does not assign or maintain version numbers for registered objects. Instead, the SO provides a version number when it submits an object. This version number has an arbitrary format and is used only to help distinguish one registry entry from another having the same name. However, the registry will maintain a record of all effects to existing registry entries of registry service requests by using the Impact object class. For example, a request to supersede a given registered object with a new registered object will affect the registry entries for both registered objects. This will be recorded in the registry by creating an Impact entry.

ebXML

In the ebXML model, the version control also is not strict. Although each registered object in the OASIS model has a single version number, the ebXML model supports two version numbers for an object—a major version and a minor version. The default for the major version of an object is 0, and the default for the minor version is 1. These version numbers also are arbitrary and provided by the SO.

The ebXML model does not reference effects the same way as the OASIS model. Rather, it has a registry audit trail capability.

11179

The 11179 registry model has extensive version control functionality. In the 11179 registry model, the RA controls all versions according to its registration policies. For example, if a data element is submitted for registration and semantic uniqueness issues cannot be resolved through the standard procedures, the final resolution may be to create a new version of a previously standardized data element. In such cases, the previous version of the data element may be proposed for retirement. The version of the data elements also may be based on changes to definition, representation (name, code, measure, etc.), or format.

Each data element in a registry may require a different treatment. For example, a change in permissible values for an employee name may not require a new version, while a change in permissible values for an account type likely will require a version change.

In the 11179 registry model, version changes also apply to classification schemes—i.e., multiple versions of a classification scheme may exist in the registry.

AUDIT TRAIL

OASIS

In the OASIS model, a historical record of all SO modifications of a registry entry is maintained in the Impact object class. Each Impact entry can be associated with a request, which is associated with a submission, which in turn is associated with an organization. All registry entries that are created, deleted, or modified by a request are traceable back to the corresponding submission and its SO. Because the registry stamps each submission with the date and time received, the date and time of the modification also can be obtained by referencing the submission that contained the request. Therefore, modifications made by an organization and their date and time can be determined at any time.

For requests that change a registration status, the status change date in the pertinent registry entry will be updated. The OASIS model also requires "an appropriate historical log of changes to registry content," but does not further specify what "appropriate" means.

ebXML

The ebXML model maintains an audit trail for each object. An entry is made in the audit trail for an object when an auditable event takes place. Auditable events usually result from a client-initiated request, and they often change the life cycle of an object. The following are defined as auditable events:

- Create object
- Delete object
- Deprecate object
- Change object version.

The ebXML registry records the following information for each auditable event:

- Date and time
- Specific user that performed an action resulting in an auditable event.

The submitting organization then can be identified through the specific user.

The ebXML filtered query capability can be used to query on auditable events. The following "ReturnRegistryEntry" example filters using the time stamp to obtain registry entries for all registered objects that were affected in the year 2000:

```
<ReturnRegistryEntry>
    <AuditableEventQuery>
        <AuditableEventFilter>
            Timestamp GE "2000-01-01" AND
            Timestamp LE "2000-12-31"
        </AuditableEventFilter>
    </AuditableEventQuery>
</ReturnRegistryEntry>
```

11179

The 11179 registry model allows for the following information to be maintained in the registry for each data element:

- *Date of receipt*: Date on which the data element was submitted for registration by the SO.
- *Date of last modification*: Date of the latest change in one or more of the attributes of a data element.

The SO also can be determined for each new data element submission and each modification.

RECOMMENDATIONS

Table 10-2 is a list of the recommendations, determinations, and issues about registry administration that relate to EPA.

Table 10-2. Recommendations, Determinations, and Issues—Registry Administration

ID No.	Topic	Type	Description	Time-frame	Comments
10.1	Submissions	D	Determine general policies and procedures for submissions	ST/NI	
10.2		I	Do objects need to be included with requests, or just an identifier, such as the URL?	ST/NI	
10.3		I	Who is allowed to submit?	ST/IM	Should allow wide base of submitters, but not necessarily open to general public Should submitters be only states and laboratories?
10.4		I	Does an SO have to be known to the XML registry before it can submit?	ST/NI	
10.5	Submission Packages	R	Allow submission packages	MT	
10.6		I	Should the number of requests contained in a submission package be limited?	MT	
10.7	Registry Packages	I	Will registry packages be used?	LT	
10.8	Registration Statuses	D	Determine registration statuses to be used in ebXML registry	ST/NI	
10.9		I	Will XML registry support deprecated and removed statuses?	MT	
10.10		D	If deprecated status is allowed, determine the time that a registered object is deprecated before its status becomes removed	MT	
10.11		D	If retirement and phase-out are allowed, need to determine the criteria by which a registered object can be phased out, and criteria for retiring registered objects	MT	
10.12	Review/ Quality Control	D	Determine review and quality control process for submissions to ebXML registry	ST/NI	
10.13	Expiration Dates	I	If expiration dates are used in ebXML registry can the SO change the expiration date of the registered object during its life?	MT	
10.14		I	If expiration dates are used in ebXML registry can the RA override the suggested expiration date?	MT	
10.15		I	If so, is the SO notified?	MT	
10.16	Audit Trail	I	Is more information needed for audit trail in ebXML registry than is specified?	ST/NI	

Chapter 11
Infrastructure

This chapter describes concepts about the infrastructure for the XML registry. We divided the chapter into sections about the following topics:

- Trading partner profiles and agreements
- Messaging service protocol
- Security.

TRADING PARTNER PROFILES AND AGREEMENTS

A trading partner profile (TPP) is a document that describes the business processes that an organization is able to engage in. The TPP describes the specific technological capabilities that a trading partner supports and the specific requirements that must be met to exchange business documents with them. A trading partner agreement (TPA) is a document that defines the conditions under which the two partners will transact business with one another. The TPA outlines concepts, such as

- business scenarios to be used,
- messaging protocol,
- contingency issues, and
- security requirements.

The IMWG Network Blueprint clearly defines the TPA concept in the context of a national environmental information network linking EPA, states, and other interested parties. The following is a summary of the proposed contents of a TPA according to the Network Blueprint:

- *Addenda*: Describes how and if addenda may be added to the agreement.
- *Communication*: Specifies the transport protocols and electronic addresses of the parties.
- *Data definition*: Describes the specific format and structure to be used for exchanging information and the URL for the format.

- *Dispute resolution*: Describes procedures, related to the terms of the agreement, for settling disputes among the partners.
- *Duration*: Identifies the time that the TPA will be in effect.
- *Exchange failure*: Addresses business continuation and identifies actions required by each party if the exchange fails.
- *Identification*: Identifies the organizations involved in the TPA and describes the general purpose of the agreement.
- *Internal systems requirements*: Addresses conditions at the boundary of participating systems.
- *Legal framework*: Includes governance, standing, and applicability issues that apply to the partners.
- *Message exchanges*: Addresses rules for submitting and responding to requests for data and the timing of data exchanges.
- *Parallel paper transactions*: Outlines expectations for exchanging documents on paper, in addition to electronic format, for part of or the duration of the TPA.
- *Performance and reliability*: Specifies the expected availability of participating systems.
- *Quality and stewardship*: Specifies the definitive source for shared data; outlines expectations about timeliness of data entry, error detection, and correction, and other conditions on which acceptability of data is predicated.
- *Record retention*: Addresses issues about transmission logs and requirements for historical data.
- *Roles and responsibilities*: Outlines specific roles and requirements of parties for performance, reliability and use of data.
- *Security*: Identifies the level of network security to be used and the specific parameters, such as certificates used for authentication, nonrepudiation, and digital envelope, and other security issues.
- *Termination*: Specifies conditions for terminating the TPA as a whole, including written notice and the effect of termination on other rights and obligations.
- *Use of data*: Specifies intended routine uses of the data to the extent that they are needed to understand the responsibilities of the parties.

Below, we discuss TPA concepts in relation to the registry models we've described.

ebXML

The ebXML model refers heavily to the concepts of TPPs and TPAs. The ebXML model refers to a TPP as a collaboration protocol profile (CPP), and a TPA as a collaboration protocol agreement (CPA). The ebXML model states the following about TPPs:

- The registration of a profile establishes a mechanism that enables trading partners to find one another and discover the business processes they support.
- The trading partner profile may include contact information, industry classification, supported business processes, and messaging service interface requirements.
- As an option, the trading partner profile may include security and other implementation-specific details.

The ebXML model states the following about TPAs:

- It describes the messaging service and business process requirements that are agreed on by two or more trading partners.
- Trading partners may decide to register their TPAs in an ebXML-compliant registry system, but registration is not mandatory.
- It has an interface to a TPP in that the TPA is derived by mutual negotiation, narrowing the capabilities of the trading partners (TPP) into what the trading partners will do (TPA).

OASIS

Because the OASIS model is based on registry concepts, it does not use trading partner profiles and trading partner agreements.

11179

Because the 11179 registry model is based on registry concepts, it does not use trading partner profiles and trading partner agreements.

MESSAGING SERVICE PROTOCOL

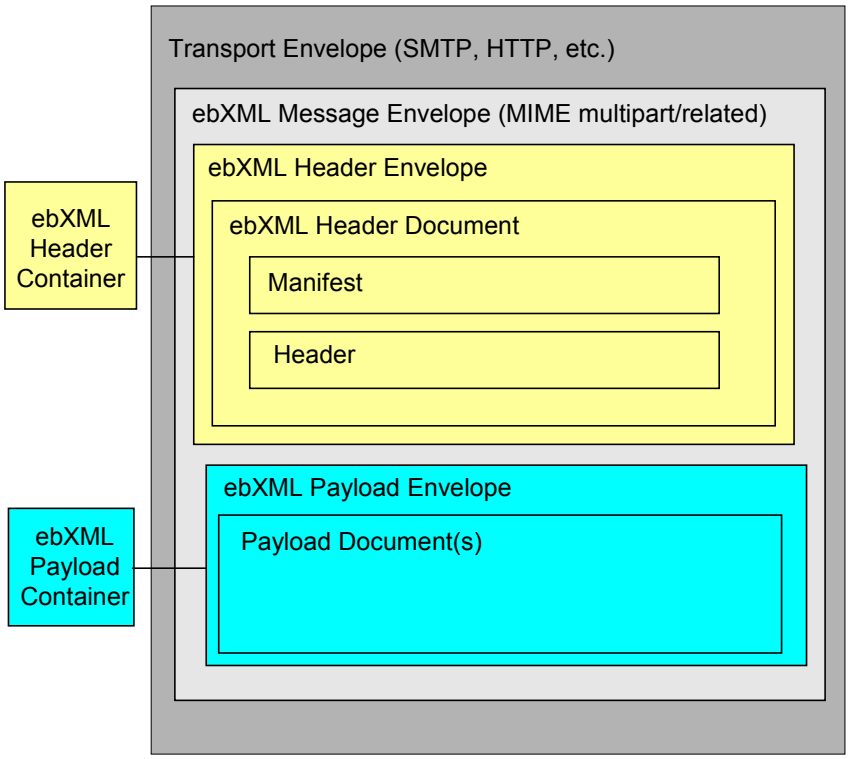
This section describes the protocol for messaging between a registry and registry client. We discuss this concept in relation to the registry models we've described.

ebXML

The ebXML model specifies a highly comprehensive messaging service protocol. The ebXML messaging service is the transport mechanism for all communications into and out of an ebXML registry, and is a standard way to exchange business messages among ebXML trading partners. The messaging service is a reliable means of exchanging business messages without relying on proprietary technologies and solutions.

The format of an ebXML message is shown in Figure 11-1.

Figure 11-1. Format of an ebXML Message



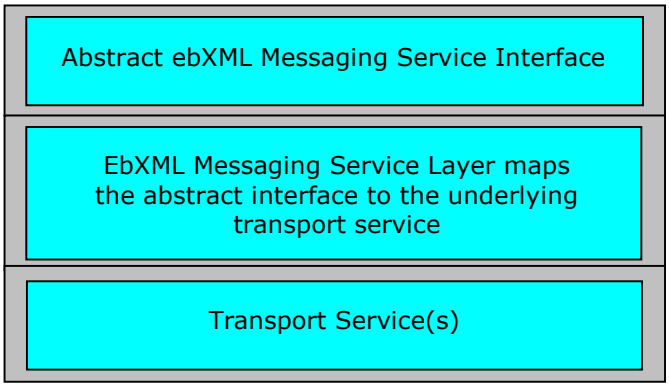
Source: ebXML.

As can be seen in the figure, an ebXML message comprises two sections or containers—a header section (necessary for routing and delivery) and a payload section (which contains the data being transported). The header section contains information, such the XML version used, the number of bytes in the message, date and time stamp, and destination URI. Because requests to and responses from

an ebXML registry are XML documents, the payload section contains the actual XML documents used for the request or response.

The ebXML messaging service is divided into the three layers shown in Figure 11-2.

Figure 11-2. ebXML Messaging Layers



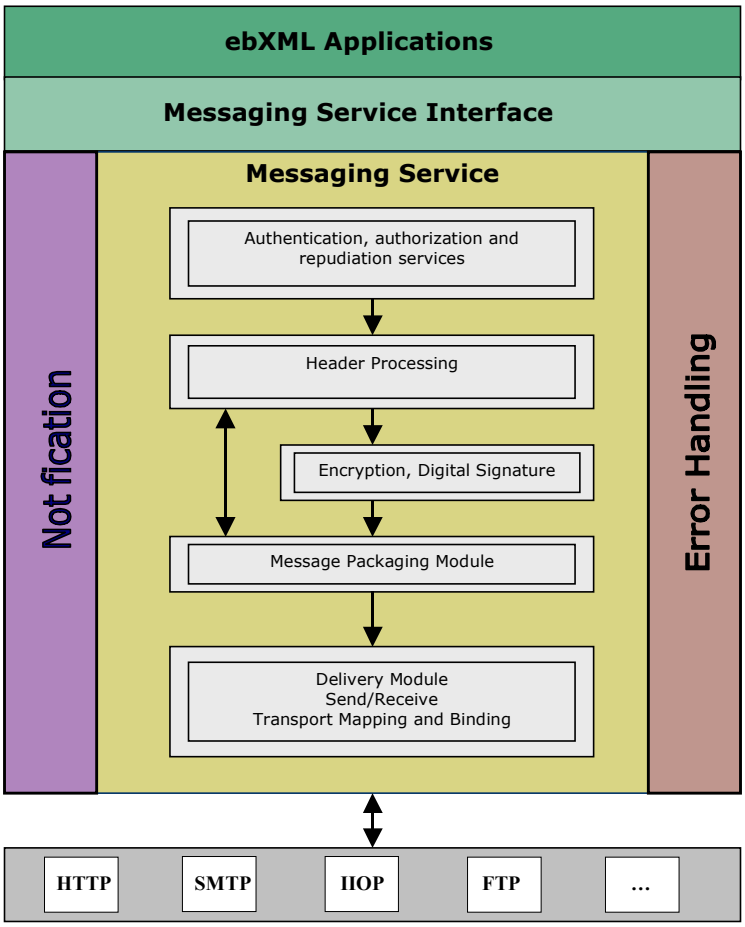
Source: ebXML.

The top layer contains the ebXML messaging service functionality described in this section. The top layer is bound to the underlying transport services by a middle layer, "the ebXML messaging service layer." The transport services may be HTTP, SMTP, FTP, etc. The ebXML messaging service layer also enforces the "rules of engagement" about security and the business process functions for delivering messages as defined by two parties in a TPA.

The vision is that the ebXML message will be constructed by COTS software known as "ebXML-compliant" software, yet to be released. An implementation also may choose to create a custom solution by building the software necessary to implement this functionality in house.

Figure 11-3 depicts the ebXML messaging service architecture, including security services, from a broad perspective.

Figure 11-3. ebXML Messaging Service Architecture



Source: ebXML.

The Simple Object Access Protocol (SOAP) specifications have also been incorporated into the ebXML messaging service. SOAP allows platform-transparent application-to-application communication using HTTP and XML. This interoperability will result in an open, widely adopted global standard for reliably transporting electronic business messages over the Internet and will reduce the cost of implementing products for all companies, regardless of their size.

OASIS

OASIS has recently formed a technical committee to build an open specification for XML message interfaces for business-to-business (B2B) transactions over the Internet. This specification, called the Business Transaction Protocol (BTP), will be designed to complement the ebXML underlying transport mechanism.

11179

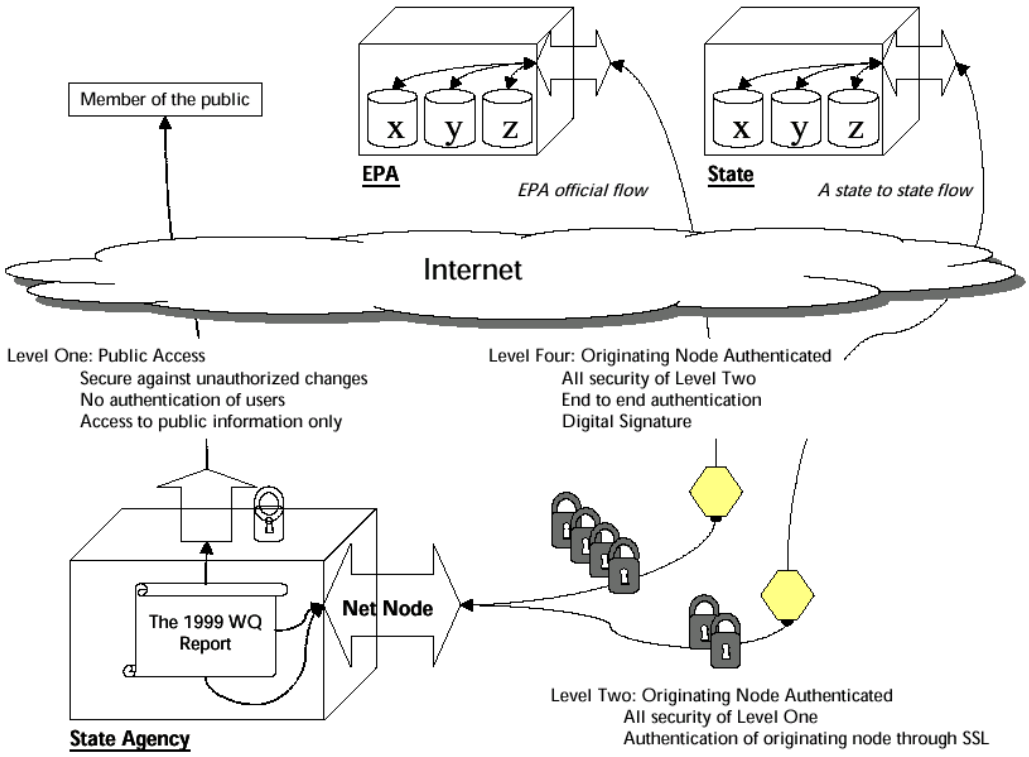
In its current form, the 11179 registry model does not specify machine-oriented mechanisms for interacting with an 11179 registry. Interface services to an 11179 registry are being defined. The services will include functions, such as retrieval, update, and maintenance.

SECURITY

This section is an overview of security considerations for an XML registry.

The IMWG Network Blueprint addresses security concepts, stating that public key infrastructure (PKI) technology (which uses digital signatures and digital certificates) should be considered for verifying and authenticating the validity of partners exchanging information. The secure socket layer (SSL) and Secure Hypertext Transfer Protocol (S-HTTP) are specified for transmitting data securely. An information request may flow over the network under four different security levels, as shown in Figure 11-4.

Figure 11-4. Security Levels for Information Flow



Source: IMWG.

Below, we discuss security considerations in relation to the registry models we've described.

ebXML

The ebXML model specifies comprehensive security features. In general, the ebXML security specification states that the registry, as well as the individual documents in the registry, should have security. The ebXML security specification calls for security needs to be addressed in four areas:

- *Authentication*: Required to identify the ownership of registry content as well as the privileges that may be assigned to a user for the registry content.
- *Authorization*: The registry should have mechanisms for ensuring appropriate access to its contents.
- *Confidentiality*: Not all registry contents will be public. Therefore, the registry should give organizations the ability to publish information that can be seen only by their partners.
- *Integrity*: Because an ebXML registry is global and distributed, the integrity of the registry content is critical to those who use the contents for mission-critical business applications. Mechanisms should exist to ensure that the content submitted by an SO is maintained in the registry without any tampering enroute or in the registry.

The following are some of the requirements stated in the security specification:

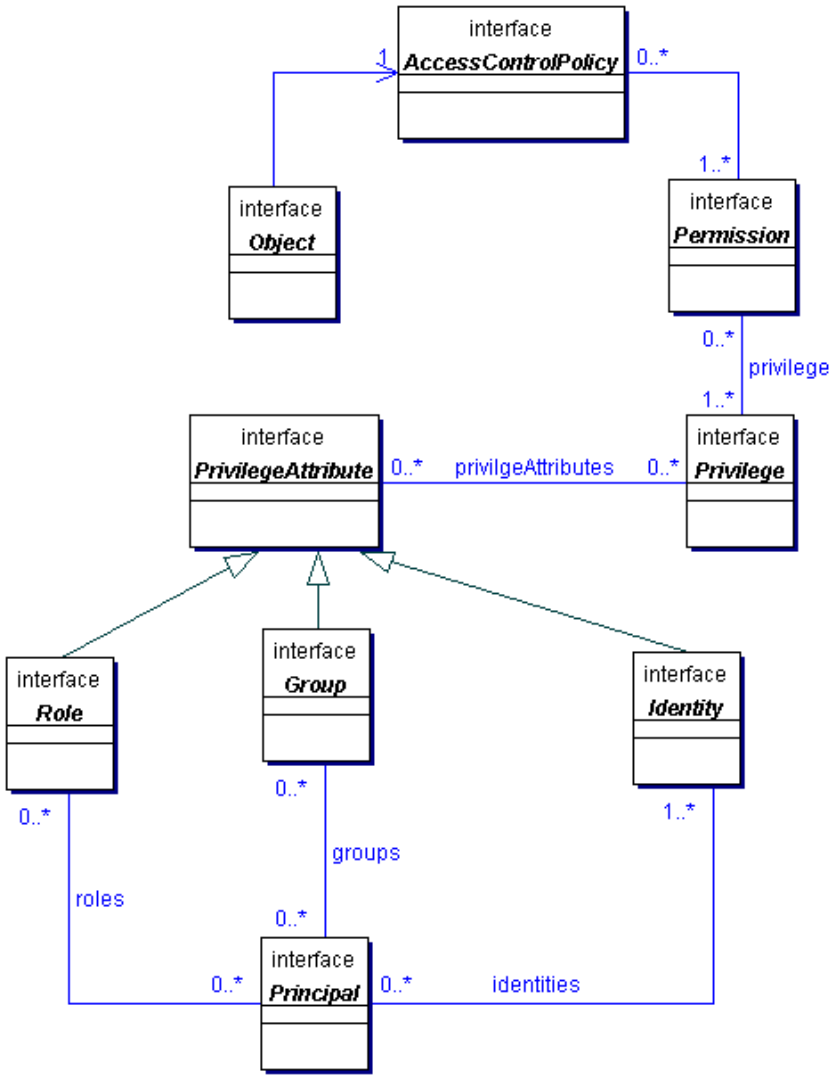
- The registry should implement user-level security and document-level authorization security.
- The authorization scheme implemented in the registry should be flexible enough to have public and private areas in the registry.
- A session-based security scheme can be used to avoid authenticating every message or interaction.
- Messages between registry services and their clients must be confidential.

The initial release of ebXML will use credential-based authorization in the form of digital certificates and digital signatures. All requests that update the registry must be digitally signed using S/MIME, PGP/MIME, or XML-DSIG, and all registry content must be signed. S/MIME and PGP/MIME are approaches used for encrypting and signing MIME messages. PGP ("Pretty Good Privacy") is a public key cryptographic program that protects the privacy of files and electronic mail. The XML-DSIG specification is a W3C candidate recommendation that defines how an XML document may be signed, either in whole or as a selective element by using a transformation, such as XPATH or XSLT.

The ebXML messaging service will use the distinguished name (DN) from the certificate to authenticate the user when the registry receives a request. The DN is the name that is associated with the digital certificate that is being used to authorize a request to the registry. The payload of the message (the section that contains the message) also must be signed, and the registry will store the signature as part of the content. When a client requests contents, the signature will be sent with the contents so the client can verify the integrity of the contents.

The ebXML model includes comprehensive privilege controls, which are shown in the ebXML security information model, Figure 11-5.

Figure 11-5. ebXML Security Information Model



Source: ebXML.

In the figure above, a principal denotes a user, each of which is assigned a role. The ebXML specifications include the following predefined roles for registry users:

- *Content Owner*: The submitter or owner of registry content. Equivalent to the Submitting Organization.
- *Registry Administrator*: "Super" user that is an administrator of the registry. Equivalent to Registration Authority.
- *Registry Guest*: An unauthenticated user of the registry that is browsing the registry. Has read-only access.

A principal also may belong to a group (such as a group of buyers in a purchasing scenario) that is assigned a set of privileges. The principal will have all of the privileges available to members of its group. A principal also may have an identity attribute in the form of a digital certificate. Access to a registered object is controlled by a policy that defines the rules by which operations may be performed on the registered object. The access control policy uses various permissions to enforce these rules.

OASIS

OASIS recently has formed a security services technical committee to complete a single security services standard, the Security Services Markup Language (S2ML). S2ML will create a common language for sharing security information about transactions and end users between companies engaged in B2B transactions over the Internet. The language defines standard XML schemas and an XML request and response protocol for describing authentication and authorization services through XML documents.

The OASIS model also implements privilege controls for requests to ensure that, in many cases, registered objects can only be updated or deleted by their SO.

11179

The 11179 registry model does not address security concepts. All privileges for updates to data elements rest centrally with the RA.

RECOMMENDATIONS

Table 11-1 is a list of the recommendations, determinations, and issues about registry administration that relate to EPA.

Table 11-1. Recommendations, Determinations, and Issues—Infrastructure

ID No.	Topic	Type	Description	Timeframe	Comments
11.1	Trading Partner Profiles	D	Determine the information that should be included in a trading partner profile	ST/NI	
11.2	Trading Partner Agreements	R	Use Network Blueprint trading partner agreement specification	ST/NI	
11.3	Messaging Service	R	Implement ebXML messaging service as transport mechanism for all communications into and out of the XML registry	MT	
11.4	Security	R	Remain consistent with Network Blueprint in security considerations	N/A	
11.5		I	How much of ebXML security specification should be adopted?	ST/NI	
11.6		I	Should privilege controls be implemented for registry requests?	ST/NI	
11.7		I	If so, how should they be implemented?	ST/NI	

Chapter 12
Additional Considerations

This chapter addresses some additional considerations for implementing an XML registry. The following topics are covered:

- *Levels of conformance*: The level at which an implementation of an XML registry conforms to the specifications for a given model.
- *Quality control*: Implementation and execution of procedures to ensure that submissions to an XML registry meet a certain standard of quality.
- *Interoperability*: The ability of an XML registry to interact with another XML registry that is based on the same or different model.
- *Legal liability*: The legal responsibilities of an organization that operates an XML registry.
- *Disaster recovery*: System operational functions that assure that a system (such as an XML registry) can remain in operation if a disaster occurs.

The purpose of this chapter is to raise awareness about the above issues, therefore, we do not make specific recommendations.

LEVELS OF CONFORMANCE

ebXML

ebXML conformance is defined as "conformance to an ebXML system that comprises all the architectural components of the ebXML infrastructure and satisfies at least the minimum conformance requirements for each of the ebXML technical specification documents." The conformance clause in each ebXML specification document specifies all the requirements that must be satisfied to claim conformance to that specification.

The ebXML specification also references the concept of conformance testing, which enables vendors to implement compatible and interoperable systems built on the ebXML foundations. The specification states that "publicly available test suites from vendor-neutral organizations, such as OASIS and NIST, should be used to verify conformance."

OASIS

The OASIS model defines several levels at which an implementation may claim conformance to its specification. The levels of conformance are determined by including specific functionality in the registry implementation. The levels of conformance are defined as follows:

- RegistryOnly
- RegistryRepositoryBasic
- RegistryRepositoryQuery

In general, the RegistryOnly level includes "basic" functionality, such as associations, classifications, and submissions. The RegistryRepositoryBasic level adds the ability to submit classification schemes and implement registry packages. The RegistryRepositoryQuery level adds Impact functionality as well as the ability to query on items, such as registry entries, contacts, requests, impacts, and organizations.

The RegistryRepositoryBasic and RegistryRepositoryQuery levels also may be specified with or without validation. "With validation" means that a registered object that is an XML document, and has a "Validates To" association with another registered object (such as a schema), will be validated upon submission.

11179

The 11179 registry metamodel defines conformance in terms of level of support for mandatory, optional, and extended data element attributes. For example, in the "strictly conforming MDR3 metadata instance" conformance level, all mandatory attributes of data elements exist, some optional data element attributes may exist, and some extended data element attributes may exist.

QUALITY CONTROL

ebXML

The ebXML specifications have deferred addressing quality control to a future phase.

OASIS

The OASIS model does not prescribe definitive quality control measures. The model does state, however, that "if the XML registry provides quality control checking, metadata should be stored regarding what specifications an entity

conforms to and the individual that performed the testing to determine that conformance."

11179

The 11179 registry model places a very heavy emphasis on quality control. This is evidenced by its comprehensive coverage of registration and administrative statuses and registration processes.

INTEROPERABILITY

ebXML

The ebXML specifications state that the primary reason for conforming to ebXML is to increase the probability of the interoperability between XML registry implementations being successful. However, although conformance is a necessary condition, it is not by itself a sufficient condition to guarantee interoperability. Successful interoperability and open interchange is more likely to be achieved if implementations conform to the requirements in the ebXML specifications.

OASIS

The OASIS specification states that work on interoperability issues has been postponed to a second phase.

11179

The 11179 registry model does not address interoperability. In its current form, the 11179 registry model does not specify any machine-oriented mechanism for interacting with an 11179 registry. However, interface services to an 11179 registry are being defined; these will include functions such as retrieval, update, and maintenance.

LEGAL LIABILITY

Issues about accessing information in an XML registry and the potential use of that information must be considered. The concept of legal liability is referenced only by the OASIS specification. The OASIS specification states that "a registry shall have a statement of limitation of legal liability (disclaiming responsibility for the use of the information in the repository, for example)."

DISASTER RECOVERY

The ability to recover from a disaster should be considered when implementing an XML registry. The concept of disaster recovery is referenced only by the OASIS specification. The OASIS specification states that “the complete content of both the registry and repository shall be backed up offsite, and the backup tested. Some plan shall be made for reconstituting the registry and repository from the backup should the original site be rendered inoperable.”

Chapter 13
Summary of Recommendations

This chapter summarizes all recommendations, determinations, and issues from each chapter of this report. The three summaries are sorted by

- topic,
- timeframe, and
- recommendation type.

In addition, we recommend that EPA do the following:

- Participate in the Federal CIO Council's activities because the work of this group will influence the way federal agencies use XML.
- Monitor the ebXML model specification activity to determine if new decisions affect the recommendations made in this report.
- Monitor the activities of other standards bodies, such as the W3C, to determine if the activities affect the recommendations made in this report.

Table 13-1. Recommendations, Determinations, and Issues—Sorted By Topic

ID	Topic	Type	Description	Timeframe	Comments
5.7	Alternate Names	R	Allow alternate names to be submitted for registered objects	ST/NI	
5.8		D	Determine policies and procedures for submitting alternate names	ST/NI	
5.9		I	Should the number of alternate names that can be associated with a registered object be limited?	ST/NI	
4.4	Architecture	R	Implement centralized architecture	ST/NI	
4.5		R	Implement distributed architecture	LT	
4.6		R	Implement combination of static and dynamic architecture	N/A	
8.1	Association Roles	R	Implement simple association roles	ST/NI	
8.2		D	Determine what a "simple" association role is	ST/NI	
8.3		R	Implement more complex association roles	MT	
8.4		D	Determine what a "complex" association role is	MT	

Table 13-1. Recommendations, Determinations, and Issues—Sorted By Topic (Continued)

ID	Topic	Type	Description	Timeframe	Comments
10.16	Audit Trail	I	Is more information needed for audit trail in ebXML registry than is specified?	ST/NI	
6.8	Business Objects	R	Identify business objects for business operational view	LT	
6.9		R	Identify core components for business operational view	LT	
6.6	Business Processes	R	Identify business processes for business operational view	LT	
6.7		R	Identify core processes for business operational view	LT	
6.3	Business Process Creation	D	Determine new business processes that need to be created for using the XML registry	ST/NI	
6.2	Business Process Integration	D	Determine existing business processes that will need to be integrated with the XML registry	ST/NI	
6.1	Business Scenarios	D	Determine business scenarios to be supported by using the XML registry	ST/NI	Reference examples on page 6-2
8.5	Classification Schemes	R	Implement simple classification schemes	ST/NI	
8.6		D	Determine what a “simple” classification scheme is	ST/NI	
8.7		R	Implement more complex classification schemes	LT	
8.8		D	Determine what a “complex” classification scheme is	LT	
8.13	Context-Sensitive Classification	I	Should context-sensitive classification be used?	MT	
6.5	Discovery	D	Determine how trading partners will “discover” the XML registry	MT	
5.14	Expiration Dates	I	Should an expiration date be assigned to registered objects?	MT	
5.15		D	If so, who will assign expiration date?	MT	RA or SO?
10.13		I	If expiration dates are used in ebXML registry can the SO change the expiration date of the registered object during its life?	MT	
10.14		I	If expiration dates are used in ebXML registry can the RA override the suggested expiration date?	MT	
10.15		I	If so, is the SO notified?	MT	
5.10	External Data	R	Allow external data to be submitted for registered objects	MT	
5.11		D	Determine policies and procedures for submitting external data	MT	What types of external data can be submitted, (i.e., white papers, user manuals, etc.)?

Table 13-1. Recommendations, Determinations, and Issues—Sorted By Topic (Continued)

ID	Topic	Type	Description	Timeframe	Comments
5.12		I	Should the amount of external data that can be associated with a registered object be limited?	MT	
6.10	Functional Service View	D	Determine how much functionality in functional service view is applicable to EPA's needs	LT	
11.3	Messaging Service	R	Implement ebXML messaging service as transport mechanism for all communications into and out of the XML registry	MT	
4.3	Multiple Contexts	R	Allow XML tags to be registered in EDR under multiple contexts	MT	
8.12	Multiple-Scheme Classification	I	Should a registered object be allowed to be classified by multiple classification schemes?	ST/NI	
5.3	Naming of Registered Objects	D	Determine standards for naming registered objects	ST/NI	
8.9	Number of Classifications per Registered Object	I	Should there be a limit to the number of classifications a registered object can have?	ST/NI	
9.5	Query Services	R	Implement simple queries	ST/NI	
9.6		D	Determine what a "simple" query is	ST/NI	
9.7		I	Should ad hoc queries be implemented?	MT	
9.8		I	Should complex queries be implemented?	LT	
7.1	Registration Authority	D	Determine criteria for identifying and establishing registration authority, for both immediate and future selection	ST/NI	
7.2		D	Determine who the registration authority will be according to established criteria	ST/NI	
7.3		D	Determine responsibilities of registration authority	ST/NI	
7.4		D	Determine what information an RA needs to provide to the XML registry for record keeping	ST/NI	
7.5		I	Will Registration Authority Identifiers be required of RAs?	ST/NI	
7.6		I	Will hierarchical RAs be used?	LT	
4.2	Registration of Data Elements	R	Create policies and procedures for adding data elements contained in schemas to EDR	ST/NI	
10.8	Registration Statuses	D	Determine registration statuses to be used in ebXML registry	ST/NI	
10.9		I	Will XML registry support deprecated and removed statuses?	MT	

Table 13-1. Recommendations, Determinations, and Issues—Sorted By Topic (Continued)

ID	Topic	Type	Description	Timeframe	Comments
10.10		D	If deprecated status is allowed, determine the time that a registered object is deprecated before its status becomes removed	MT	
10.11		D	If retirement and phase-out are allowed, need to determine the criteria by which a registered object can be phased out, and criteria for retiring registered objects	MT	
5.2	Registry Entry Contents	D	Determine the information that will be contained in a registry entry	ST/IM	
10.7	Registry Packages	I	Will registry packages be used?	LT	
9.1	Registry Services	R	Implement essential registry services	ST/NI	
9.2		D	Determine what an “essential” registry service is	ST/NI	
9.3		R	Implement non-essential registry services	MT	
9.4		D	Determine what a “non-essential” registry service is	MT	
4.7	Repository	D	Determine physical location of repository	ST/IM	
4.8		D	Determine number of repositories required	ST/IM	
7.7	Responsible Organization	D	Determine criteria for identifying and establishing responsible organizations, for both immediate and future selection	ST/NI	
7.8		D	Determine who ROs will be according to established criteria	ST/NI	
7.9		D	Determine responsibilities of ROs	ST/NI	
7.10		D	Determine what information an RO needs to provide to the XML registry for record keeping	ST/NI	
10.12	Review/ Quality Control	D	Determine review and quality control process for submissions to ebXML registry	ST/NI	
11.4	Security	R	Remain consistent with Network Blueprint in security considerations	N/A	
11.5		I	How much of ebXML security specification should be adopted?	ST/NI	
11.6		I	Should privilege controls be implemented for registry requests?	ST/NI	
11.7		I	If so, how should they be implemented?	ST/NI	
10.1	Submissions	D	Determine general policies and procedures for submissions	ST/NI	
10.2		I	Do objects need to be included with requests, or just an identifier, such as the URL?	ST/NI	

Table 13-1. Recommendations, Determinations, and Issues—Sorted By Topic (Continued)

ID	Topic	Type	Description	Timeframe	Comments
10.3		I	Who is allowed to submit?	ST/IM	Should allow wide base of submitters, but not necessarily be open to general public Should submitters be only states and laboratories?
10.4		I	Does an SO have to be known to the XML registry before it can submit?	ST/NI	
8.10	Submission of Classifications	I	Should users be allowed to submit new classifications for registered objects?	MT	
8.11	Submission of Classification Schemes	I	Should users be allowed to submit classification schemes as registered objects?	MT	
10.5	Submission Packages	R	Allow submission packages	MT	
10.6		I	Should the number of requests contained in a submission package be limited?	MT	
5.4	Submission Validation	R	Ensure that all submitted XML documents and schemas are validated before being stored in the XML registry	ST/NI	
5.5		D	Determine procedures for handling submissions that do not validate properly	ST/NI	How to notify submitter?
5.6		I	Should an object that fails validation be stored until the valid version is received, or rejected completely?	ST/NI	May store with a special registration status, such as “invalid—new submission pending” May notify original submitter if new submission not received in a given timeframe
7.11	Submitting Organization	D	Determine criteria for identifying and establishing submitting organizations, for both immediate and future selection	ST/NI	
7.12		D	Determine who SOs will be according to established criteria	ST/NI	
7.13		D	Determine responsibilities of SOs	ST/NI	
7.14		D	Determine what information an SO needs to provide to the XML registry for record keeping	ST/NI	
11.2	Trading Partner Agreements	R	Use Network Blueprint trading partner agreement specification	ST/NI	
11.1	Trading Partner Profiles	D	Determine the information that should be included in a trading partner profile	ST/NI	
5.1	Types of Registered Objects	D	Determine what types of registered objects will be allowed in the XML registry	ST/IM	
6.4	User Access	D	Determine how users will access the XML registry	ST/NI	

Table 13-1. Recommendations, Determinations, and Issues—Sorted By Topic (Continued)

ID	Topic	Type	Description	Timeframe	Comments
5.13	Versions	D	Determine policy for handling versions of registered objects in ebXML registry	ST/NI	How will SO determine version numbers? Should this be left entirely up to the SO?
4.1	XML Tags	R	Allow XML tags to be stored in EDR	ST/IM	In progress

Table 13-2. Recommendations, Determinations, and Issues—Sorted By Timeframe

ID	Topic	Type	Description	Timeframe	Comments
4.1	XML Tags	R	Allow XML tags to be stored in EDR	ST/IM	In progress
4.7	Repository	D	Determine physical location of repository	ST/IM	
4.8		D	Determine number of repositories required	ST/IM	
5.1	Types of Registered Objects	D	Determine what types of registered objects will be allowed in the XML registry	ST/IM	
5.2	Registry Entry Contents	D	Determine the information that will be contained in a registry entry	ST/IM	
10.3	Submissions	I	Who is allowed to submit?	ST/IM	Should allow wide base of submitters, but not necessarily be open to general public. Should submitters be only states and laboratories?
4.2	Registration of Data Elements	R	Create policies and procedures for adding data elements contained in schemas to EDR	ST/NI	
4.4	Architecture	R	Implement centralized architecture	ST/NI	
5.3	Naming of Registered Objects	D	Determine standards for naming registered objects	ST/NI	
5.4	Submission Validation	R	Ensure that all submitted XML documents and schemas are validated before being stored in the XML registry	ST/NI	
5.5		D	Determine procedures for handling submissions that do not validate properly	ST/NI	How to notify submitter?
5.6		I	Should an object that fails validation be stored until the valid version is received, or rejected completely?	ST/NI	May store with a special registration status, such as “invalid—new submission pending” May notify original submitter if new submission not received in a given timeframe
5.7	Alternate Names	R	Allow alternate names to be submitted for registered objects	ST/NI	
5.8		D	Determine policies and procedures for submitting alternate names	ST/NI	

Table 13-2. Recommendations, Determinations, and Issues—Sorted By Timeframe (Continued)

ID	Topic	Type	Description	Timeframe	Comments
5.9		I	Should the number of alternate names that can be associated with a registered object be limited?	ST/NI	
5.13	Versions	D	Determine policy for handling versions of registered objects in ebXML registry	ST/NI	How will SO determine version numbers? Should this be left entirely up to the SO?
6.1	Business Scenarios	D	Determine business scenarios to be supported by using the XML registry	ST/NI	Reference examples on page 6-2
6.2	Business Process Integration	D	Determine existing business processes that will need to be integrated with the XML registry	ST/NI	
6.3	Business Process Creation	D	Determine new business processes that need to be created for using the XML registry	ST/NI	
6.4	User Access	D	Determine how users will access the XML registry	ST/NI	
7.1	Registration Authority	D	Determine criteria for identifying and establishing registration authority, for both immediate and future selection	ST/NI	
7.2		D	Determine who the registration authority will be according to established criteria	ST/NI	
7.3		D	Determine responsibilities of the registration authority	ST/NI	
7.4		D	Determine what information an RA needs to provide to the XML registry for record keeping	ST/NI	
7.5		I	Will Registration Authority Identifiers be required of registration authorities?	ST/NI	
7.7	Responsible Organization	D	Determine criteria for identifying and establishing Responsible Organizations, for both immediate and future selection	ST/NI	
7.8		D	Determine who ROs will be according to established criteria	ST/NI	
7.9		D	Determine responsibilities of ROs	ST/NI	
7.10		D	Determine what information an RO needs to provide to the XML registry for record keeping	ST/NI	
7.11	Submitting Organization	D	Determine criteria for identifying and establishing submitting organizations, for both immediate and future selection	ST/NI	
7.12		D	Determine who SOs will be according to established criteria	ST/NI	
7.13		D	Determine responsibilities of SOs	ST/NI	
7.14		D	Determine what information an SO needs to provide to the XML registry for record keeping	ST/NI	

Table 13-2. Recommendations, Determinations, and Issues—Sorted By Timeframe (Continued)

ID	Topic	Type	Description	Timeframe	Comments
8.1	Association Roles	R	Implement simple association roles	ST/NI	
8.2		D	Determine what a “simple” association role is	ST/NI	
8.5	Classification Schemes	R	Implement simple classification schemes	ST/NI	
8.6		D	Determine what a “simple” classification scheme is	ST/NI	
8.9	Number of Classifications per Registered Object	I	Should there be a limit to the number of classifications a registered object can have?	ST/NI	
8.12	Multiple-Scheme Classification	I	Should a registered object be allowed to be classified by multiple classification schemes?	ST/NI	
9.1	Registry Services	R	Implement essential registry services	ST/NI	
9.2		D	Determine what an “essential” registry service is	ST/NI	
9.5	Query Services	R	Implement simple queries	ST/NI	
9.6		D	Determine what a “simple” query is	ST/NI	
10.1	Submissions	D	Determine general policies and procedures for submissions	ST/NI	
10.2		I	Do objects need to be included with requests, or just an identifier, such as the URL?	ST/NI	
10.4		I	Does an SO have to be known to the XML registry before it can submit?	ST/NI	
10.8	Registration Statuses	D	Determine registration statuses to be used in ebXML registry	ST/NI	
10.12	Review/ Quality Control	D	Determine review and quality control process for submissions to ebXML registry	ST/NI	
10.16	Audit Trail	I	Is more information needed for audit trail in ebXML registry than is specified?	ST/NI	
11.1	Trading Partner Profiles	D	Determine the information that should be included in a trading partner profile	ST/NI	
11.2	Trading Partner Agreements	R	Use Network Blueprint trading partner agreement specification	ST/NI	
11.5	Security	I	How much of ebXML security specification should be adopted?	ST/NI	
11.6		I	Should privilege controls be implemented for registry requests?	ST/NI	
11.7		I	If so, how should they be implemented?	ST/NI	
4.3	Multiple Contexts	R	Allow XML tags to be registered in EDR under multiple contexts	MT	
5.10	External Data	R	Allow external data to be submitted for registered objects	MT	

Table 13-2. Recommendations, Determinations, and Issues—Sorted By Timeframe (Continued)

ID	Topic	Type	Description	Timeframe	Comments
5.11		D	Determine policies and procedures for submitting external data	MT	What types of external data can be submitted, (i.e., white papers, user manuals, etc.)?
5.12		I	Should the amount of external data that can be associated with a registered object be limited?	MT	
5.14	Expiration Dates	I	Should an expiration date be assigned to registered objects?	MT	
5.15		D	If so, who will assign expiration date?	MT	RA or SO?
6.5	Discovery	D	Determine how trading partners will "discover" the XML registry	MT	
8.3	Association Roles	R	Implement more complex association roles	MT	
8.4		D	Determine what a "complex" association role is	MT	
8.10	Submission of Classifications	I	Should users be allowed to submit new classifications for registered objects?	MT	
8.11	Submission of Classification Schemes	I	Should users be allowed to submit classification schemes as registered objects?	MT	
8.13	Context-Sensitive Classification	I	Should context-sensitive classification be used?	MT	
9.3	Registry Services	R	Implement non-essential registry services	MT	
9.4		D	Determine what a "non-essential" registry service is	MT	
9.7	Query Services	I	Should ad hoc queries be implemented?	MT	
10.5	Submission Packages	R	Allow submission packages	MT	
10.6		I	Should the number of requests contained in a submission package be limited?	MT	
10.9	Registration Statuses	I	Will XML registry support deprecated and removed statuses?	MT	
10.10		D	If deprecated status is allowed, determine the time that a registered object is deprecated before its status becomes removed	MT	
10.11		D	If retirement and phase-out are allowed, need to determine the criteria by which a registered object can be phased out, and criteria for retiring registered objects	MT	
10.13	Expiration Dates	I	If expiration dates are used in ebXML registry can the SO change the expiration date of the registered object during its life?	MT	

Table 13-2. Recommendations, Determinations, and Issues—Sorted By Timeframe (Continued)

ID	Topic	Type	Description	Timeframe	Comments
10.14		I	If expiration dates are used in ebXML registry can the RA override the suggested expiration date?	MT	
10.15		I	If so, is the SO notified?	MT	
11.3	Messaging Service	R	Implement ebXML messaging service as transport mechanism for all communications into and out of the XML registry	MT	
4.5	Architecture	R	Implement distributed architecture	LT	
6.6	Business Processes	R	Identify business processes for business operational view	LT	
6.7		R	Identify core processes for business operational view	LT	
6.8	Business Objects	R	Identify business objects for business operational view	LT	
6.9		R	Identify core components for business operational view	LT	
6.10	Functional Service View	D	Determine how much functionality in functional service view is applicable to EPA's needs	LT	
7.6	Registration Authority	I	Will hierarchical RAs be used?	LT	
8.7	Classification Schemes	R	Implement more complex classification schemes	LT	
8.8		D	Determine what a "complex" classification scheme is	LT	
9.8	Query Services	I	Should complex queries be implemented?	LT	
10.7	Registry Packages	I	Will registry packages be used?	LT	
4.6	Architecture	R	Implement combination of static and dynamic architecture	N/A	
11.4	Security	R	Remain consistent with Network Blueprint in security considerations	N/A	

Table 13-3. Recommendations, Determinations, and Issues—Sorted By Recommendation Type

ID	Topic	Type	Description	Timeframe	Comments
4.7	Repository	D	Determine physical location of repository	ST/IM	
4.8		D	Determine number of repositories required	ST/IM	
5.1	Types of Registered Objects	D	Determine what types of registered objects will be allowed in the XML registry	ST/IM	
5.2	Registry Entry Contents	D	Determine the information that will be contained in a registry entry	ST/IM	
5.3	Naming of Registered Objects	D	Determine standards for naming registered objects	ST/NI	
5.5	Submission Validation	D	Determine procedures for handling submissions that do not validate properly	ST/NI	How to notify submitter?
5.8	Alternate Names	D	Determine policies and procedures for submitting alternate names	ST/NI	
5.11	External Data	D	Determine policies and procedures for submitting external data	MT	What types of external data can be submitted, (i.e., white papers, user manuals, etc.)?
5.13	Versions	D	Determine policy for handling versions of registered objects in ebXML registry	ST/NI	How will SO determine version numbers? Should this be left entirely up to the SO?
5.15	Expiration Dates	D	If an expiration date is to be assigned to registered objects, who will assign expiration date?	MT	RA or SO?
6.1	Business Scenarios	D	Determine business scenarios to be supported by using the XML registry	ST/NI	Reference examples on page 6-2
6.2	Business Process Integration	D	Determine existing business processes that will need to be integrated with the XML registry	ST/NI	
6.3	Business Process Creation	D	Determine new business processes that need to be created for using the XML registry	ST/NI	
6.4	User Access	D	Determine how users will access the XML registry	ST/NI	
6.5	Discovery	D	Determine how trading partners will “discover” the XML registry	MT	
6.10	Functional Service View	D	Determine how much functionality in functional service view is applicable to EPA’s needs	LT	
7.1	Registration Authority	D	Determine criteria for identifying and establishing registration authority, for both immediate and future selection	ST/NI	
7.2		D	Determine who the registration authority will be according to established criteria	ST/NI	

Table 13-3. Recommendations, Determinations, and Issues—Sorted By Recommendation Type (Continued)

ID	Topic	Type	Description	Timeframe	Comments
7.3		D	Determine responsibilities of registration authority	ST/NI	
7.4		D	Determine what information an RA needs to provide to the XML registry for record keeping	ST/NI	
7.7	Responsible Organization	D	Determine criteria for identifying and establishing responsible organizations, for both immediate and future selection	ST/NI	
7.8		D	Determine who ROs will be according to established criteria	ST/NI	
7.9		D	Determine responsibilities of ROs	ST/NI	
7.10		D	Determine what information an RO needs to provide to the XML registry for record keeping	ST/NI	
7.11	Submitting Organization	D	Determine criteria for identifying and establishing submitting organizations, for both immediate and future selection	ST/NI	
7.12		D	Determine who SOs will be according to established criteria	ST/NI	
7.13		D	Determine responsibilities of SOs	ST/NI	
7.14		D	Determine what information an SO needs to provide to the XML registry for record keeping	ST/NI	
8.2	Association Roles	D	Determine what a “simple” association role is, so that simple association roles may be implemented	ST/NI	
8.4	Association Roles	D	Determine what a “complex” association role is, so that complex association roles may be implemented	MT	
8.6	Classification Schemes	D	Determine what a “simple” classification scheme is, so that simple classification schemes may be implemented	ST/NI	
8.8	Classification Schemes	D	Determine what a “complex” classification scheme is, so that complex classification schemes may be implemented	LT	
9.2	Registry Services	D	Determine what an “essential” registry service is, so that essential registry services may be implemented	ST/NI	
9.4	Registry Services	D	Determine what a “non-essential” registry service is, so that non-essential registry services may be implemented	MT	
9.6	Query Services	D	Determine what a “simple” query is, so that simple queries may be implemented	ST/NI	
10.1	Submissions	D	Determine general policies and procedures for submissions	ST/NI	
10.8	Registration Statuses	D	Determine registration statuses to be used in ebXML registry	ST/NI	

Table 13-3. Recommendations, Determinations, and Issues—Sorted By Recommendation Type (Continued)

ID	Topic	Type	Description	Timeframe	Comments
10.10	Registration Statuses	D	If deprecated status is allowed, determine the time that a registered object is deprecated before its status becomes removed	MT	
10.11	Registration Statuses	D	If retirement and phase-out are allowed, need to determine the criteria by which a registered object can be phased out, and criteria for retiring registered objects	MT	
10.12	Review/ Quality Control	D	Determine review and quality control process for submissions to ebXML registry	ST/NI	
11.1	Trading Partner Profiles	D	Determine the information that should be included in a trading partner profile	ST/NI	
5.6	Submission Validation	I	Should an object that fails validation be stored until the valid version is received, or rejected completely?	ST/NI	May store with a special registration status, such as "invalid—new submission pending" May notify original submitter if new submission not received in a given timeframe
5.9	Alternate Names	I	Should the number of alternate names that can be associated with a registered object be limited?	ST/NI	
5.12	External Data	I	Should the amount of external data that can be associated with a registered object be limited?	MT	
5.14	Expiration Dates	I	Should an expiration date be assigned to registered objects?	MT	
7.5	Registration Authority	I	Will Registration Authority Identifiers be required of RAs?	ST/NI	
7.6		I	Will hierarchical RAs be used?	LT	
8.9	Number of Classifications per Registered Object	I	Should there be a limit to the number of classifications a registered object can have?	ST/NI	
8.10	Submission of Classifications	I	Should users be allowed to submit new classifications for registered objects?	MT	
8.11	Submission of Classification Schemes	I	Should users be allowed to submit classification schemes as registered objects?	MT	
8.12	Multiple-Scheme Classification	I	Should a registered object be allowed to be classified by multiple classification schemes?	ST/NI	
8.13	Context-Sensitive Classification	I	Should context-sensitive classification be used?	MT	
9.7	Query Services	I	Should ad hoc queries be implemented?	MT	
9.8		I	Should complex queries be implemented?	LT	

Table 13-3. Recommendations, Determinations, and Issues—Sorted By Recommendation Type (Continued)

ID	Topic	Type	Description	Timeframe	Comments
10.2	Submissions	I	Do objects need to be included with requests, or just an identifier, such as the URL?	ST/NI	
10.3		I	Who is allowed to submit?	ST/IM	Should allow wide base of submitters, but not necessarily be open to general public Should submitters be only states and laboratories?
10.4		I	Does an SO have to be known to the XML registry before it can submit?	ST/NI	
10.6	Submission Packages	I	Should the number of requests contained in a submission package be limited?	MT	
10.7	Registry Packages	I	Will registry packages be used?	LT	
10.9	Registration Statuses	I	Will XML registry support deprecated and removed statuses?	MT	
10.13	Expiration Dates	I	If expiration dates are used in ebXML registry can the SO change the expiration date of the registered object during its life?	MT	
10.14		I	If expiration dates are used in ebXML registry can the RA override the suggested expiration date?	MT	
10.15		I	If so, is the SO notified?	MT	
10.16	Audit Trail	I	Is more information needed for audit trail in ebXML registry than is specified?	ST/NI	
11.5	Security	I	How much of ebXML security specification should be adopted?	ST/NI	
11.6		I	Should privilege controls be implemented for registry requests?	ST/NI	
11.7		I	If so, how should they be implemented?	ST/NI	
4.1	XML Tags	R	Allow XML tags to be stored in EDR	ST/IM	In progress
4.2	Registration of Data Elements	R	Create policies and procedures for adding data elements contained in schemas to EDR	ST/NI	
4.3	Multiple Contexts	R	Allow XML tags to be registered in EDR under multiple contexts	MT	
4.4	Architecture	R	Implement centralized architecture	ST/NI	
4.5		R	Implement distributed architecture	LT	
4.6		R	Implement combination of static and dynamic architecture	N/A	
5.4	Submission Validation	R	Ensure that all submitted XML documents and schemas are validated before being stored in the XML registry	ST/NI	
5.7	Alternate Names	R	Allow alternate names to be submitted for registered objects	ST/NI	
5.10	External Data	R	Allow external data to be submitted for registered objects	MT	

Table 13-3. Recommendations, Determinations, and Issues—Sorted By Recommendation Type (Continued)

ID	Topic	Type	Description	Timeframe	Comments
6.6	Business Processes	R	Identify business processes for business operational view	LT	
6.7		R	Identify core processes for business operational view	LT	
6.8	Business Objects	R	Identify business objects for business operational view	LT	
6.9		R	Identify core components for business operational view	LT	
8.1	Association Roles	R	Implement simple association roles	ST/NI	
8.3		R	Implement more complex association roles	MT	
8.5	Classification Schemes	R	Implement simple classification schemes	ST/NI	
8.7		R	Implement more complex classification schemes	LT	
9.1	Registry Services	R	Implement essential registry services	ST/NI	
9.3	Registry Services	R	Implement non-essential registry services	MT	
9.5	Query Services	R	Implement simple queries	ST/NI	
10.5	Submission Packages	R	Allow submission packages	MT	
11.2	Trading Partner Agreements	R	Use Network Blueprint trading partner agreement specification	ST/NI	
11.3	Messaging Service	R	Implement ebXML messaging service as transport mechanism for all communications into and out of the XML registry	MT	
11.4	Security	R	Remain consistent with Network Blueprint in security considerations	N/A	

Appendix A
Glossary

11179 registry:	A registry based on the ISO/IEC 11179 registry model.
11179 registry metamodel (MDR3):	A conceptual data model that specifies the structure of a metadata registry and defines basic attributes for specifying administered components.
Accredited Standards Committee (ASC) X12:	A committee that links together the standards for multiple industries and sets the norm for a more effective exchange of information.
Activity diagram:	A flow diagram that models business workflow.
Ad hoc query:	An ebXML query type that uses a SQL-based query language for more complex queries than are possible using other ebXML query types.
Administered component:	A component for which administrative information is recorded. 11179 registry metamodel term for a registered object.
Administration record:	11179 registry metamodel term for a registry entry.
Administrative status:	An indication of the position of a registration request in the processing life cycle of a Registration Authority.
Alternate name:	A name that is associated with a registered object that is valid in a particular context. By using alternate names, a registered object can be considered in multiple contexts.
American National Standards Institute (ANSI) X3.285 standard:	A standard that specifies the structure of a data registry as a conceptual data model and provides the attributes for identifying the characteristics of data that are necessary to clearly describe, inventory, analyze, and classify data.

Analysis phase:	A phase of the business operational view of the ebXML model in which activity and sequence diagrams are created that describe the business processes.
Association:	An XML registry concept that represents a relationship between two registered objects.
Association role:	An attribute that describes the type of association between two registered objects.
Attribute:	A characteristic of an object or entity.
Audit trail:	A historical record that enables tracing system activity.
BizTalk:	A Microsoft XML application integration and e-commerce framework that enables organizations to produce consistent XML schemas.
Browse and drill down query:	An ebXML query type in which the user browses registry content according to classification schemes, selects a registered object, and looks down ("drills down") to view the object details.
Business library:	A collection of business processes and business process components.
Business object:	A conceptual object used to describe and represent a business concept or purpose.
Business operational view (BOV):	One of two views used in the ebXML model to describe the relevant aspects of business transactions. The BOV deals with high-level business operational issues that apply to the business needs of ebXML trading partners.
Business process:	A collection of business transactions between business partners. Also may be an internal activity in one business.
Business scenario:	XML versions of the business processes and associated business messages that an organization is able to engage in.

Business transaction:	A clearly defined exchange of business messages (in the form of business documents) resulting in a new legal or commercial state between two trading partners.
Business Transaction Protocol (BTP) Specification:	A developing OASIS-based specification for XML message interfaces for business-to-business (B2B) transactions over the Internet.
Candidate recommendation:	A World Wide Web Consortium review stage indicating that the technical community has significantly reviewed a proposal but has not yet approved it for general use.
Certificate authority (CA):	An authority that issues and manages security credentials and public keys for encrypting messages as part of a public key infrastructure.
Class diagram:	A diagram that shows the existence of object classes and their relationship in the logical view of a system.
Classification:	An arrangement or division of objects into groups that are based on the objects' common characteristics, e.g., origin, composition, structure, application, or function.
Classification scheme:	A specification of a hierarchy of values, names, and codes on which a classification is based.
Collaboration diagram:	A diagram that illustrates aspects of business partner collaboration.
Collaboration protocol agreement (CPA):	ebXML term for a trading partner agreement.
Collaboration protocol profile (CPP):	ebXML term for a trading partner profile.
Common business object:	A conceptual object that exists in more than one business domain.
Common Object Request Broker Architecture (CORBA):	An open, vendor-independent architecture and infrastructure created by the Object Management Group. CORBA enables objects to interact in a heterogeneous, distributed environment, independent of the platforms on which they reside and the techniques used to implement them.

Contact:	A person, role, or other entity in an organization that has some relationship to a registered object. The contact may be the person who submitted the object initially for registration.
Context-sensitive classification:	A technique for associating a classification in multiple contexts and for which an additional classification node is used to clarify the context for each case.
Core component:	A reusable, low-level data structure that occurs in many areas of industry or business information interaction and captures information about a real-world business concept, and the relationship between that concept and other business concepts. Core components together comprise business objects.
Core library:	A collection of core processes and components. A core library contains data and process definitions, including relationships and cross-references, expressed in business terminology that may be tied to an accepted industry classification scheme or taxonomy.
Core process:	A set of business actions independent of industry specifics. Because core processes are generic, they can be reused with specific context and business rules in different vertical industries.
Data element:	A unit of data for which the definition, identification, representation, and permissible values are specified by means of a set of attributes.
Data steward:	An individual or organizational element responsible for the accuracy, reliability, and currency of the metadata for a registered object. Also referred to as a *responsible organization*.
Definition phase:	A phase of the business operational view of the ebXML model in which a business problem is described by using use-case diagrams and descriptions.
Deprecated status:	An ebXML and OASIS registration status that is used when the Submitting Organization indicates that a registered object will soon be replaced or withdrawn.

Design phase:	A phase of the business operational view of the ebXML model in which object-oriented principles may be used to generate collaboration diagrams and possibly a state diagram.
Digital certificate:	An electronic "credit card" issued by a certificate authority that establishes a user's credentials on the Internet.
Digital signature:	An electronic signature that can be used to authenticate the identity of the sender of a message or of the signer of a document. A digital signature also can be used to ensure that the original content of the message or document that has been conveyed is unchanged.
Disaster recovery:	System operation functions that assure that a system (such as an XML registry) can remain operational if a disaster occurs.
Discovery and retrieval phase:	A phase of the functional service view of the ebXML model that covers all aspects of discovering ebXML-related resources, such as trading partner profiles, core libraries, business libraries.
Distinguished name (DN):	The name on the digital certificate that is being used to authorize a request to the registry.
Document type definition (DTD):	A document that defines the required structure of an XML document and the constraints on its content.
Electronic business XML (ebXML):	A joint venture between OASIS and UN/CEFACT whose vision is to enable a global electronic marketplace.
Environmental Council of the States (ECOS):	A national non-profit, non-partisan association of state and territorial environmental commissioners whose goal is to improve the environment of the United States.
Environmental Data Registry (EDR):	EPA's comprehensive authoritative source of reference about environmental data. The EDR is an implementation of a 11179 registry.

Environmental Information Exchange Network:	A network designed by the Information Management Work Group whose goal is to dramatically improve the quality and availability of environmental data to environmental agencies and the public.
Extensible Markup Language (XML):	A markup language for documents that contain structured information. XML is a project of the W3C.
External data:	Information that is related to a registered object and is supporting information but resides outside the registry. An example is user documentation for a registered XML schema.
Federal CIO Council:	The principal interagency forum for improving practices in the design, modernization, use, sharing, and performance of federal agency information resources. The Federal CIO Council's XML working group hosts the XML.gov website.
File Transfer Protocol (FTP):	A protocol that provides a standard way for computers to copy files between computers on the Internet.
Filtered query:	An ebXML query type in which the user queries on registry entries and registered objects by using "filters" to narrow down the query results.
Functional service view (FSV):	One of two views used in the ebXML model to describe the relevant aspects of business transactions. The FSV centers on functional capabilities, service interfaces, and protocols, and describes how the business operational view is implemented using the selected technology.
Hierarchical Registration Authority:	A concept specified in the 11179 registry model in which multiple registration authorities "report" to a single registration authority.
Hypertext Transfer Protocol (HTTP):	The protocol used by the World Wide Web to format and transmit messages.
Information Management Work Group (IMWG):	A work group comprising EPA and state environmental agencies organized through ECOS, whose mission is to build locally and nationally accessible, cohesive, and coherent information systems.

Internet Assigned Numbers Authority (IANA):	The IANA oversees the allocation of internet protocol addresses to internet service providers.
Impact:	An OASIS object class that captures the effects of registry service requests on existing registry entries. For example, a request to supersede a registered object with a new registered object will affect the registry entries for both registered objects.
Implementation phase:	A phase of the functional service view of the ebXML model that deals specifically with the procedures for creating an application of the ebXML infrastructure.
Information model:	A representation of the components that compose a system and the relationship between those components.
International Electrotechnical Commission (IEC):	A world organization that prepares and publishes international standards for all electrical, electronic, and related technologies.
International Press Telecommunications Council (IPTC):	A group that develops and publishes industry standards for the exchange of news data.
International Standards Organization (ISO):	A worldwide federation of national standards bodies whose mission is to promote the development of standardization and related activities worldwide.
Interoperability:	In the context of XML registry, the ability of an XML registry to interact with another XML registry that is based on the same or different model.
ISO/IEC 11179 registry model:	A standard for registering data elements as described in the ISO/IEC 11179 standard.
ISO/IEC 11179 standard:	A metadata standard for data elements that describes the standardization and registration of data elements to share them and make them understandable.
Level of conformance:	In the context of XML registry, the level at which an implementation of an XML registry conforms to the specifications for a specific XML registry model.
Life cycle:	The phases that an object in a registry can pass through from the time it is submitted to the registry to the time that it is removed.

Metadata:	Data that define and describe other data.
Multipurpose Internet Mail Extension (MIME):	A standard that defines a method of moving multimedia files through mail gateways.
National Technology Transfer and Advancement Act of 1995 (NTTA):	An act that issues guidance to federal agencies for using voluntary standards and conformity assessment protocols.
Network blueprint:	A blueprint for the Environmental Information Exchange Network developed by IMWG. The goal of the Environmental Information Exchange Network is to dramatically improve the quality and availability of environmental data to environmental agencies and the public.
North American Industry Classification System (NAICS):	A standard code system that provides common codes for business establishments and industries in Canada, Mexico, and the United States.
Object class:	In object-oriented programming, a template from which an object is constructed. For example, an "employee" object is constructed from an "employee" class that contains the characteristics of an employee (e.g., name, position, and salary)
Object stability:	The likelihood that a registered object will change in the future.
Object Management Group (OMG):	An organization formed to create a component-based software marketplace by hastening the introduction of standardized object software.
ObjectManager interface:	The ebXML interface through which registry service requests are carried out.
ObjectManagerClient interface:	The ebXML interface that an ebXML registry client uses to connect with the ObjectManager interface.
ObjectQueryManager interface:	The ebXML interface through which query registry services are carried out.
ObjectQueryManagerClient interface:	The mechanism that an ebXML registry client uses to connect with the ObjectQueryManager interface.

Office of Management and Budget (OMB) Circular A-119:	A document that directs agencies to use voluntary consensus standards in lieu of government-unique standards except where using the consensus standards would be inconsistent with law or impractical.
Open-EDI Reference Model (ISO 14662):	A model that describes electronic data interchange among multiple autonomous organizations to accomplish a shared business goal.
Organization for Structured Information Standards (OASIS):	A non-profit international consortium that creates interoperable industry specifications based on public standards, such as XML and standard generalized markup language. ebXML is a joint venture between OASIS and UN/CEFACT.
Pretty Good Privacy (PGP):	A public key encryption program that protects the privacy of files and electronic mail.
Pretty Good Privacy/ Multipurpose Internet Mail Extensions (PGP/MIME):	A digital envelope security based on the PGP standard, integrated with MIME security.
Public key infrastructure (PKI):	The combination of software, encryption technologies, and services that enables organizations to protect the security of their communications and business transactions on the Internet.
Quality control:	Procedures and their use for ensuring a standard of quality for submissions to an XML registry.
Reaffirming:	An OASIS concept in which a submitting organization changes the expiration date during the life of the registered object by submitting a new expiration date to the registry.
Reference document:	In the 11179 registry model, a document that contains pertinent details for consulting about a subject. 11179 registry metamodel term for external data.
Registered object:	An object in an XML registry that an author or producer wants to have visible to the world so that it can be used by a client or customer.
Registration Authority Identifier (RAI):	An internationally recognized organization code specified in the 11179 registry model that is used to construct internationally unique identifiers for each data element in a registry.

Registration authority (RA):	A recognized expert organization that is responsible for populating and maintaining a registry.
Registration status:	Status of a registered object in a registry at a specific time.
Registry:	A facility that stores metadata about registered objects and allows registered objects or their metadata to be operated on.
Registry entry:	Metadata about a registered object.
Registry package:	A set of registered objects that can be operated on as a group.
Registry services:	Operations that are performed on the metadata and contents in a registry, such as registering an object, querying a registered object, and submitting a classification scheme.
Registry services interface:	An interface used for the automatic or human-initiated execution of registry services.
Repository:	A storage facility for registered objects with an access method that enables retrieving individual objects, perhaps with an additional authentication or permission layer.
Repository item:	ebXML term for a registered object.
Responsible organization (RO):	An individual or organizational element responsible for the accuracy, reliability, and currency of the metadata for a registered object. Also referred to as a *data steward*.
Run-time phase:	A phase of the functional service view of the ebXML model in which ebXML messages are exchanged between trading partners using the ebXML messaging service.
Schema:	A document that defines the required structure of an XML document and constrains its content. Similar in concept to a DTD, but with broader functionality.
Secure Hypertext Transfer Protocol (S-HTTP):	An extension to the Hypertext Transfer Protocol that enables exchanging files over the Internet securely.

Secure socket layer (SSL):	A security protocol developed by Netscape for transmitting private documents over the Internet.
Secure/Multipurpose Internet Mail Extensions (S/MIME):	A format and protocol for adding cryptographic signature or encryption services to Internet MIME messages.
Security Services Markup Language (S2ML):	A developing XML-based security standard that will create a common language for sharing security information about transactions and end users between companies transacting B2B over the Internet.
Semantics management:	Refers to maintaining detailed information about the meaning of data elements.
Sequence diagram:	A diagram that is used to model details of objects and the passing of messages between objects.
Simple Mail Transfer Protocol (SMTP):	The Internet's standard host-to-host e-mail transport protocol.
Simple Object Access Protocol (SOAP):	An XML-based protocol that allows platform-transparent application-to-application communication using HTTP and XML.
Slot:	A component of the ebXML registry information model that allows metadata to be dynamically added to a registry entry.
Standard Generalized Markup Language (SGML):	An international standard for describing marked-up electronic text. XML was derived from SGML.
State diagram:	An analysis tool that can be used when a system (or a component of a system) passes through a series of discrete states during its operation.
Structured Query Language (SQL):	A standard programming language developed by IBM for extracting information from and updating a database.
Submission:	Meaning differs depending on model. In the ebXML model, a submission is a collection of object metadata. In the OASIS model, a submission is a collection of requests sent from a submitting organization to a registry.

Submitting organization (SO):	An individual or organizational element designated to identify and report data elements suitable for registration.
Synonymous name:	In the 11179 standard, a single or multiword designation that differs from the data element name, but represents the same data element concept.
Trading partner agreement (TPA):	A document that defines the conditions under which two partners will transact business together. The ebXML specifications refer to a trading partner agreement as a collaboration protocol agreement.
Trading partner profile (TPP):	A document that describes the business processes that an organization is able to engage in. The ebXML specifications refer to a trading partner profile as a collaboration protocol profile.
Trading partner:	A participant in a business-to-business data exchange.
UN/CEFACT:	The United Nations body for trade facilitation and electronic business. ebXML is a joint venture between OASIS and UN/CEFACT.
UN/CEFACT Modeling Methodology (UMM):	A business process modeling methodology developed by UN/CEFACT that uses the Unified Modeling Language as its modeling tool.
Unified Modeling Language (UML):	An object-oriented modeling tool developed by Rational Software and adopted as a standard by the Object Management Group.
Unique identifier (UID):	A unique identifier used for defining the location of an object in a registry.
Universal Description, Discovery and Integration initiative (UDDI):	A collaborative initiative between IBM, Ariba, and Microsoft whose focus is on enabling large organizations to manage their network of smaller business customers through the shared operation of a business registry on the World Wide Web.
Use case diagram:	A diagram of requirements from the perspective of how the user will use the system rather than from the perspective of the features that the system is required to incorporate.

Use case:	A collection of possible sequences of interactions between a system and its users in relation to a particular goal.
World Wide Web Consortium (W3C):	A group that develops specifications to lead the World Wide Web to its full potential as a forum for information, commerce, communication, and collective understanding. XML is a project of the W3C.
XML digital signature (XML-DSIG):	A W3C candidate recommendation that defines how an XML document should be digitally signed.
XML document:	A document that contains data surrounded by XML tags.
XML.gov:	A website hosted by the federal CIO XML working group whose purpose is to facilitate using XML efficiently and effectively through cooperative efforts among government agencies, including partnerships with commercial and industrial organizations.
XML.org:	An open, vendor-neutral website for XML resources hosted by OASIS.

Terms are used throughout this report that differ from one registry model to another. Table A-1 describes these terms.

Table A-1. Terms that Differ

Term	ebXML	OASIS	11179
Registered object	Repository item	Registered object	Administered component
External data	External data	External data	Reference document
Trading partner agreement	Collaboration protocol agreement	N/A	N/A
Trading partner profile	Collaboration protocol profile	N/A	N/A

Appendix B
Bibliography

1. Blueprint for a National Environmental Information Exchange Network, 30 October 2000.

2. ISO/IEC 11179-1, *Specification and Standardization of Data Elements—Part 1: Framework.*

3. ISO/IEC 11179-2, *Information Technology—Specification and Standardization of Data Elements—Part 2: Classification for Data Elements*, Committee Draft 1, March 1999.

4. ISO/IEC 11179-3, *Information Technology—Specification and Standardization of Data Elements—Part 3: Basic Attributes of Data Elements*, June 1994.

5. ISO/IEC 11179-3, *Information Technology—Data Management and Interchange - Metadata Registries (MDR)—Part 3: Registry Metamodel (MDR3),* June 18, 2000.

6. ISO/IEC 11179-4, *Information Technology—Specification and Standardization of Data Elements—Part 4: Rules and Guidelines for the Formulation of Data Definitions,* October 1994.

7. ISO/IEC 11179-5, *Information Technology—Specification and Standardization of Data Elements—Part 5: Naming and Identification Principles for Data Elements*, June 1995.

8. ISO/IEC 11179-6, *Information Technology—Specification and Standardization of Data Elements—Part 6: Registration of Data Elements.*

9. Draft Technical Report, *Concept of Operations for a Data Element Registry*, July 1996.

10. Procedures for Achieving Content Consistency in ISO/IEC 11179 Metadata Registries, Working Paper Draft 3.0, September 1999.

11. Intelligent Transportation Systems Data Registry Functional Operating Procedures, 28 September 1998.

12. OASIS Registry/Repository Technical Specification, Working Draft v1.1, 20 December 2000.

13. ebXML Technical Architecture Specification v1.0.4, February 16, 2001.

14. ebXML Registry Information Model v1.0, May 8, 2001.

15. ebXML Registry Services Specification v1.0, May 10, 2001.

16. ebXML Registry Security Proposal v0.3, December 24, 2000.

17. ebXML Business Process Methodology Guidelines, Guidelines for Applying the UN/CEFACT Unified Modeling Methodology to the ebXML Metamodel and Core Components, Version 0.3a, October 30, 2000.

Reference Websites

1. ebXML: www.ebXML.com

2. OASIS: www.oasis-open.org/index.shtml

3. ISO/IEC 11179 Metadata Registry Implementation Coalition: http://hmrha.hirs.osd.mil/mrc/

Appendix C
Abbreviations

ANSI	American National Standards Institute
ASC	Accredited Standards Committee
BOV	business operational view
BTP	Business Transaction Protocol
B2B	business-to-business
CA	certificate authority
CORBA	Common Object Request Broker Architecture
COTS	commercial off-the-shelf
CPA	collaboration protocol agreement
CPP	collaboration protocol profile
DN	distinguished name
DTD	document type definition
ebXML	electronic business XML
ECOS	Environmental Council of the States
EDI	Electronic Data Interchange
EDR	Environmental Data Registry
FSV	functional service view
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
IANA	Internet Assigned Numbers Authority
IEC	International Electrotechnical Commission
IMWG	State/EPA Information Management Work Group
IPTC	International Press Telecommunications Council
ISO	International Standards Organization
MDR3	registry metamodel
MIME	Multipurpose Internet Mail Extension
NAICS	North American Industry Classification System
NIST	National Institute of Standards and Technology

NTTA	National Technology Transfer and Advancement Act
OASIS	Organization for Structured Information Standards
OMB	Office of Management and Budget
OMG	Object Management Group
PGP	Pretty Good Privacy
PGP/MIME	Pretty Good Privacy/Multipurpose Internet Mail Extension
PKI	Public key infrastructure
RA	registration authority
RAI	Registration Authority Identifier
RO	responsible organization
S/MIME	Secure/Multipurpose Internet Mail Extension
SGML	Standard Generalized Markup Language
S-HTTP	Secure Hypertext Transfer Protocol
SME	Small and Medium-Sized Enterprise
SMTP	Simple Mail Transfer Protocol
SO	submitting organization
SOAP	Simple Object Access Protocol
SSL	secure socket layer
SQL	Structured Query Language
S2ML	Security Services Markup Language
TPA	trading partner agreement
TPP	trading partner profile
UDDI	Universal Description, Discovery and Integration
UID	unique identifier
UML	Unified Modeling Language
UMM	UN/CEFACT Modeling Methodology
W3C	World Wide Web Consortium
XML	Extensible Markup Language
XML-DSIG	XML digital signature