Information Interoperability and Intelligent Documents

Ed Chase and Marc Straat
Adobe Systems Europe
chase@adobe.com; mstraat@adobe.com

Abstract : This document examines how information can be made interoperable by organising the concepts and data models within a community of interest or domain, and by serializing these concepts inside a discovery mechanism like a metadata registry. Highly structured types of documents can be assembled and validated directly from models generated from registry artifacts. Unstructured documents can use descriptive metadata that references from this same foundation of registry concepts. This creates a single source for both data models and metadata vocabularies within a domain. These domain-specific registries can be linked across lines of business and communities of interest. The data models and relationships within them can be mapped to one another, enabling the bridging of data models. Finally, *Intelligent Documents* that can convey both structured and unstructured content can act as a data exchange and presentation platform for a wide range of content types.

1 Information Interoperability

There have always been challenges to sharing information. Before the computer, language differences, as well as differences in language meanings, were the greatest barrier to information exchange. To truly understand any element of information, you need to understand how it relates to a particular scope or context. If I were to send a message that contained the statement: "I was at the bank this afternoon", does this mean that I visited a financial institution, or that I was down by the river? Which afternoon was "this afternoon"? Information needs context to clarify its meaning. Context-specific terminologies can specify concepts within communities of practice and lines of business. Interoperable information needs to have both defined syntax and context.

As a human, one can make logical inferences about information based on the stored knowledge relationships in my brain. A computer application doesn't have this luxury. Without implicit context, a computer application doesn't know a "Bank" as a financial institution, an embankment, the tilt of an aircraft, or any of half-dozen other homonyms. In computer applications, context is typically applicable only within a single application, business process, or specific data type. For computers to meaningfully share information, context rules must be firmly established.

2 Doesn't XML Enable Interoperability Between Systems?

XML was widely heralded as the answer to interoperability issues. Unfortunately, XML by itself does nothing beyond establish a common framework for data. It solves only the most basic issues of what format the data syntax and serialisation will take. Supplemental specifications such as XML Schema can define and validate more complex logical data structures and types. It is still left up to the individual implementers to define their own XML vocabularies. This flexibility is both a strength and a weakness. It allows for the development of meaningful XML taxonomies for any domain, but it does nothing to resolve concepts across these domains. For example, any given implementer's serialisation of a concept like "color" is arbitrary within their scope.

Figure 1 – Describing the Same Concept with Different XML:

```
1. <Color>Red</Color>

2.  <Color>#FF0000</Color>

3. <Color>
        <Red>255</Red><Green>0</Green><Blue>0</Blue>
   </Color>

4. <色>赤い</色>
```

While XML Schema datatypes and Namespaces can separate concepts from different vocabularies, they don't really offer any way to provide any intrinsic meaning for information. A document created by one application may (and often does) have data that could be effectively used by another system, but without an explicit mapping of one XML data model to another, the information can not cross the divide.

3 Focused XML Standards Development

Many communities of interest have developed XML standards to define the information they use and exchange. Standards development groups like MISMO and ACORD have developed data models and XML Schemas for the mortgage banking and insurance sectors. UN/CEFACT and OASIS have fostered the development of a number of general and specialised XML standards. Government agencies have begun to serialise their data models in XML.

The great benefit of this process is that the experts in a particular subject matter - the people who best understand their data and processes - are cooperating to define their data models and XML vocabularies. Organisations within a community of interest are able to exchange XML messages based on their industry standard with a great degree of interoperability. However, the problems with this approach start to appear when you look at interoperability beyond a specific scope. For example, law enforcement groups need a data model for the concept of a person. Postal services also use names in the context of locating their customers. In the course of an investigation, law enforcement officers may need to draw on information from external sources. How does data get from the postal XML for the definition of a “CustomerInfo” to the law enforcement XML for “SuspectData”?

Figure 2 – Stovepipe Standards

• A Postal Data Model
<CustomerInfo>
<ResidentName>Ed</ ResidentName >
<ResidentAddress>125 Main Street</ ResidentAddress >
</CustomerInfo >

• A Law Enforcement Data Model
<SuspectData>
<LegalName>Ed</LegalName >
<HomeAddress>125 Main Street</HomeAddress>
</SuspectData>

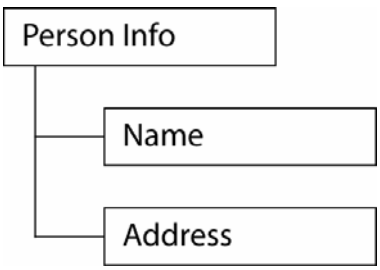
Many of the same concepts, terms, and values are going to be re-used in different contexts by different groups. Data model harmonisation is difficult enough within one community, it's nearly impossible across many. The problem starts to look like “stove-pipe” standards. Industry standard XML vocabularies seem to have solved issues within their domains, but have simply pushed the interoperability issues from a system-to-system level to the line-of-business and industry level, and too a greater level of complexity. How can we move past this without compromising the requirements of each community? The compatible concepts in these data models need to be relatable at a higher level.

4 Organising Information

By organising concepts, we can create logical relationships about them. This means that by providing a very precise meaning to information, it can be found and used by many systems, not just those designed specifically around it.

There are two distinct methodologies for information modeling - *Contextual* and *Non-contextual*. Contextual modeling assumes that the meaning of the information contained in a document is established by the nature of the document. The format that a "Purchase Order" document takes might vary, but its context is firmly established. Contextual modeling can be reflected by hierarchal data models serialized in an XML Schema. Contextual data models are very efficient within a known scope, and for fixed transactions involving structured data (i.e. purchase orders and shipping documents).

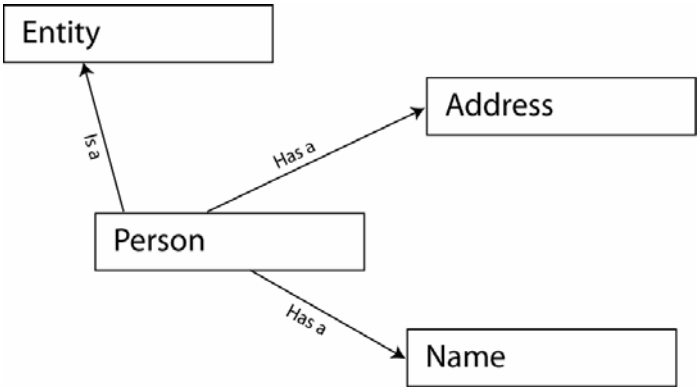
Figure 3 – Contextual Modeling



Non-contextual modeling is much more flexible. It can be used to describe much more complex concept relationships. Where contextual modeling is based on hierarchies, non-contextual modeling is largely based on associations. Non-contextual models can describe

logical relationships beyond class/subclass/instance and can provide information context beyond the scope of a particular document type. Non-contextual models are often used for metadata vocabularies and can be serialized in RDF XML. Unfortunately RDF statements are not as easily machine-processable as hierarchal XML. RDF also has several different serialization possibilities, making it somewhat difficult to automatically validate with many XML tools.

Figure 4 – Non-Contextual Modeling



Both contextual and non-contextual models are necessary to effectively deal with the full range of possible information types. Fortunately, we can encompass them all within the scope of an ontology. Ontologies are knowledge representation tools that capture concepts and the relationships among them. Relationships to other concepts are what provide meaning to any particular concept. Relationships in ontologies can be both hierarchal and associative, allowing the serialization of both contextual and non-contextual data models. An ontology can contain the types of contextual structures represented by XML schema, as well as the non-contextual relationships of RDF statements.

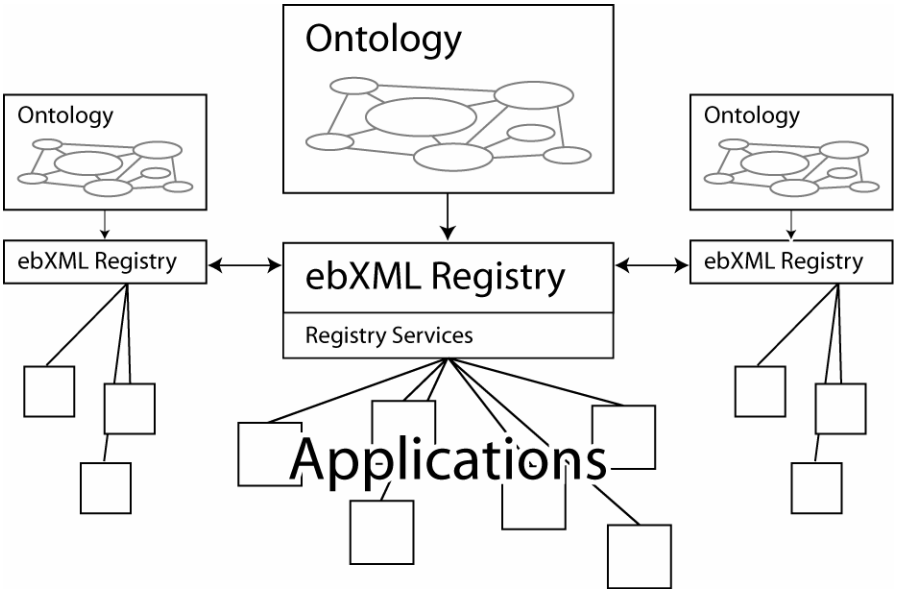
Ontologies are intrinsically extensible. Since they are based on associations, separate ontologies for different domains can be combined without adversely affecting one another in the process. Different government agencies, law enforcement and military organizations, banking, real-estate, insurance, publishing, imaging, and construction organizations in different countries each have different data and metadata models. Information needs to be able to flow between all of these groups, but directly harmonizing data models among these communities would be extraordinarily difficult. Representing these information concepts in an ontology is the first step to creating interoperability across domains.

5 From Ontology to Registry

How can the people and systems that work with information benefit from an ontology? By itself, the expression of any ontology (usually in OWL [Web Ontology Language]) is a framework. The concepts stored in an ontology need to be made programmatically available to the applications that process information. This allows these applications to find, assemble, manage, and process the information intelligently, based on the common conceptual framework. An ebXML Registry (ebXMLR) is a business tool for storing and organizing information about objects, classifications, and relationships. It stores metadata about objects (or "artifacts"). Registries can be used to serialize the ontology relationships for use by applications. ebXML Registries can also be federated. Registry federation means that two or more registries function as a single entity. Federation enables cross-domain relationships to be

established between the ontologies of multiple communities of interest.

Figure 5 – Serializing an Ontology in an ebXML Registry



The framework of the ebXML registry is the Registry Information Model (RIM). The RIM is a set of extensible metadata associations that can be used to categorize classes and objects and the relationships among them. These associations can be among any objects - XML Schemas, Core Components, code lists, web service descriptions, or any other artifact. The RIM provides a basic set of associations, but can be extended to include any of the relationships of an OWL ontology.

Applications can use the registry as a discovery mechanism. They can build, deconstruct, and process data structures based on registry metadata relationships. The registry provides the programming and reference interface to an application or developer seeking to create interoperable content or to map information between applications or domains. For example, XML Schemas for building structured documents can be dynamically assembled, and XML instances can be dynamically validated using the registry. RDF statements can also be built from the registry concepts. Information can pass from contextual to non-contextual data models since they share a single reference in the embedded ontology framework. Cross-domain XML vocabularies become transparent through their associations at the ontology level. Larger pieces of data can be re-used without custom mapping between concepts.

6 Information Structures

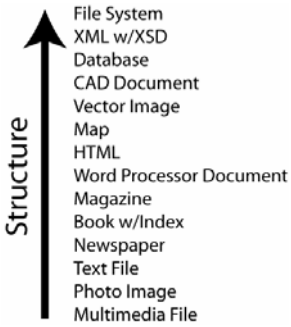
How are the relationships in a registry used to create interoperable documents? First, we need to look at how information is structured at a very basic level. In a computing context, information is typically contained in files and documents. A document can be a word processor file or a spreadsheet, a database or XML file, an image or multimedia clip, or a combination of multiple formats. Document types can be characterized by levels of structure.

Highly structured documents typically use rigorously specified data-types. For example - a "Date" field in a structured document can be expected to contain a calendar date in a very specific format (like DDMMYYYY). Structured data can be found in database-type

applications and electronic forms. It can be definitively categorized and searched, and it has a high level of machine-readability.

While structured information is ideal for interaction with and among computers, much of the data that people interact with lacks the formal rigidity of forms and databases. Images and multimedia files are particularly free from any logical interpretation other than that of a (human) viewer. Machine processing of the information in unstructured documents is difficult. Computers can store and render images and video clips, but the content in those files can not be intelligently searched, sorted, or categorized. Fortunately, even the most unstructured document types have some clues - a title, filename, or location within a file hierarchy that can provide some rudimentary description of a document. And increasingly, unstructured documents have some categorization or metadata attached to them, making it not "Unstructured" but actually more "Semi-structured" information.

Figure 6 – Levels of Relative Information Structure



Semi-structured information provides some description or structure for unstructured content. Semi-structured data can be anything from a document or image with embedded metadata to a video clip stored in a content management system or a logically tagged eBook. Metadata can be as simple as a document creation date, author, or other document-level data, or as comprehensive as an index, logical tagging, or object-level metadata.

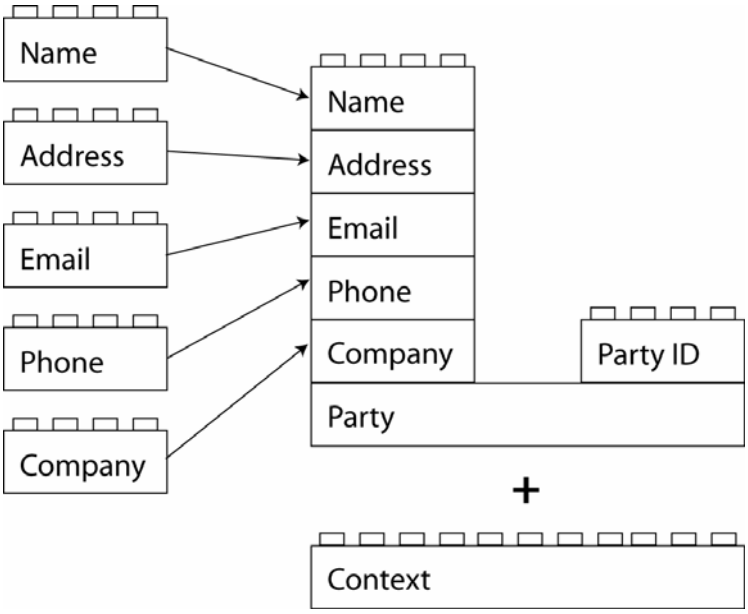
7 Structured Data - ebXML Core Components and XML Schema

The OASIS ebXML Core Components Technical Specification (CCTS) provides "A way to identify, capture and maximize the re-use of business information to support and enhance information interoperability across multiple business situations." The Core Components approach provides a means of developing XML-based vocabularies that can be cleanly mapped to other CCTS-based vocabularies by de-constructing data into syntax and context-neutral atomic elements. By developing their individual XML vocabularies using the CCTS, communities of interest can build data models that will be more seamlessly interoperable at a higher level. Mapping between CCTS-based vocabularies requires less manual harmonization and lends itself better to automation.

Core Components are context-neutral. A context mechanism is used to qualify CCs for a specific purpose, allowing them to be refined and customized for the needs of different domains. It's both this separation and association of context to data elements that facilitates better interoperability between Core Components-based vocabularies. From our previous example: A law enforcement XML vocabulary may contain a <SuspectInfo> with properties like <LegalName> and <HomeAddress>. A postal XML vocabulary may contain an

<CustomerInfo>, with properties like <ResidentName> and <ResidentAddress>. Both of these vocabularies might meet the specific needs of their communities, but bridging between them will require mapping each element and concept directly. In contrast, if they were developed with CCTS, they would use common elements for the data and structures of their <SuspectInfo> and <CustomerInfo> concepts. Functionally and logically, they are identical. Industry specific context allows these models to meet the needs of their domains, but they are readily interoperable because they share the same foundations.

Figure 7 – Core Components



Core components have one more advantage when it comes to interoperability. They are designed to be used with an ebXML registry. By deriving industry data models from an ontology and serializing them with Core Components in federated ebXML Registries, communities of interest can provide for a very high level of interoperability between similar concepts in different domains without the need for comprehensive data-mapping or middleware.

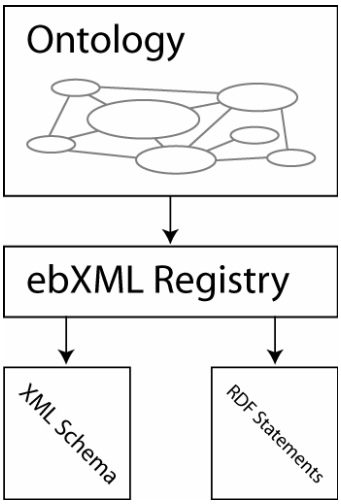
However, not all data and documents are able to benefit from directly from CCTS methods. Unstructured data is as critical as its structured counterpart in business and government, yet can not be readily interpreted into something like XML Schema. Any system for providing information interoperability needs to address all types of information - structured, unstructured, and anywhere between. Fortunately, using ontologies and metadata registries as a foundation for structured data models has also given us a means of interoperating with less-structured documents.

8 Unstructured Data - Organization for Semi-structured Documents

A similar interoperability mechanism is needed for unstructured documents. It's actually much simpler. Since OWL is an RDF-based language for defining ontologies, the ontology-driven relationships in a registry are built on the properties and associations of the RDF statements in OWL. The resulting extended Registry Information Model is a direct reflection of the RDF statements from the ontology. RDF metadata vocabularies can be derived from the ontology concepts in the registry and referenced by instance documents to bring meaning to their

content. In this way, the same concepts in the registry that are used to assemble schemas for structured documents are used to build and associate RDF metadata vocabularies for semi-structured documents. Applications that can reference and extract this metadata are able to make more intelligent searches, accelerate workflows, re-use information, and provide intelligent content analysis.

Figure 8 – Defining Different Levels of Structure



9 The Last Mile of Interoperability

The final step in information interoperability is to provide universal access to all types of content. We have established methods for creating and understanding interoperable data and metadata, but how do we ensure that users across domains have access to data and documents from different applications, platforms, and different specialized XML vocabularies?

Organizations including Adobe, Xerox PARC, Forrester Research, and the META Group have individually begun to refine the concept of Intelligent Documents. Intelligent Documents "…blend structured and unstructured content, processing information, are independent of the system by which they were created, support high fidelity multimodal imaging, offer security and rights management, and support document and object-level metadata."

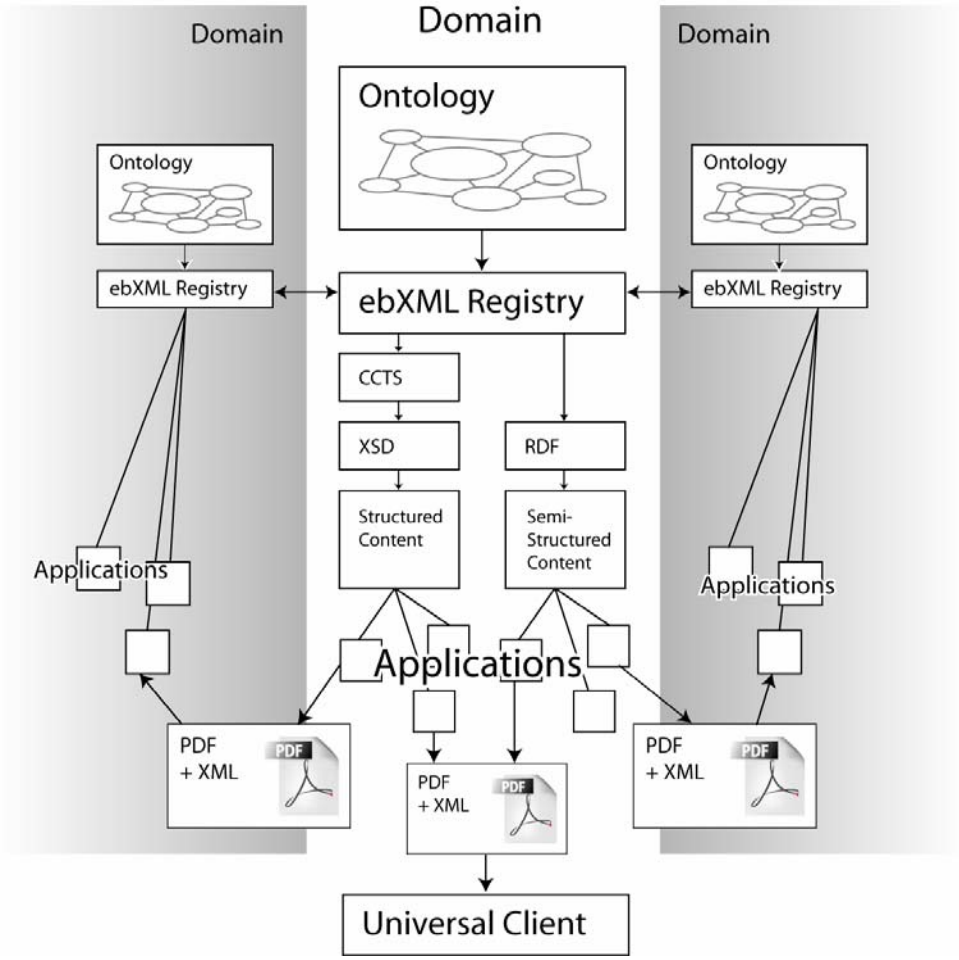
Compatibility with both XML and RDF is required. Document portability, consistency, and rich content need to be addressed. Rich content and proprietary and/or binary spreadsheets and word processing files, images, multimedia and CAD-type drawings need to be supported. Intelligent Documents need to be both machine and human readable, and support both highly structured and rich semi-structured content.

PDF is an Intelligent Document format that meets all of these requirements. PDF can convey both structured and unstructured information. It can contain and render embedded XML data from any Schema, and it can be wrapped in XML (through the XDP specification) and presented in XML (using the XFA specification). For metadata, PDF supports an RDF subset through Adobe's XMP. The PDF v1.6, XDP, XFA, and XMP specifications are all publicly available. Adobe publishes and maintains the PDF format and its XML extensions, and there are hundreds of third-party developers and thousands of applications built on them. There are also several existing and emerging ISO standards based on PDF (PDF/X, PDF/A, PDF/E, PDF/Access). PDF files can contain rich content from multiple applications. They have

multimedia and accessibility support, as well as comprehensive security and rights management features. The PDF client - the Adobe Reader - is freely distributed and available on all major operating systems and platforms.

PDF can support structured data from any XML schema, and it can act as a platform for structured XML data generated from concepts in an ontology-based ebXML registry. PDF forms can be used for XML data collection and exchange. Their self-contained nature means that they can pass beyond the borders of enterprise applications, yet retain their content, authenticity, and security. PDF also works exceptionally well with semi-structured data and rich content. PDF files can capture exact content from any application, and they can contain RDF-based XMP document and object metadata. Unstructured content can be organized and tagged with PDF to add structure and accessibility. PDF rich content support includes audio, video, 3D, CAD drawings. Virtually any type of content can be precisely and accurately represented in PDF, and it can convey the structured and semi-structured concepts derived from an underlying ontology.

Figure 9 – Information Interoperability



By using PDF as a universal document format, the concepts and relationships that provide intelligent structure to data, metadata, and content are preserved and conveyed beyond the scope and domain of the originating systems. Different communities of interest can exchange PDF documents that contain the data and metadata concepts contained in their ontologies.

Federated registries ensure that the domain-specific relationships among concepts found in the documents are interoperable with those of each registry in the federation, and each community or organization that participates.

10 Conclusion

Let us look at how this type of model could work with an existing standard. UN eDocs is a standards initiative managed by UN/CEFACT. The goal of eDocs is to provide an open standard for electronic shipping and trade documents. The specifications include a data model, XML schemas, paper/eForm layout rules, code lists, and web services. For governments and shipping-related companies to fully benefit from the eDocs standards, these groups would need map their existing data models and systems to the eDocs data specifications when producing trade documents.

Recently, the UN eDocs XML standards have been re-engineered with CCTS methods. There is now the potential to manage the eDocs standards with an ebXML registry, allowing eDocs implementers to map other line-of-business standards to eDocs through registry associations. Companies ordering goods will be able to submit eDocs-based purchase orders from their enterprise systems. Shippers and trade facilitators can automatically populate their own internal forms and data systems from the same information. Customs agencies will be able to rapidly gather highly accurate data and intelligently verify shipments. Regulators and inspectors can easily access and collate trade data. Financial institutions in different countries can securely exchange and verify letters of credit and bills of lading without exchanging paper documents. Law enforcement groups will be able to use data from multiple sources to quickly track and locate suspicious items. Non-eDocs file attachments can take advantage of registry concepts to generate accurate and context-specific metadata.

All of these functions - ordering, purchasing, shipping, customs, financial, and security - involve data and documents crossing between domains with different standards. By using intelligent documents and a universal client, the barriers to implementing integrated electronic solutions for small and medium organizations are greatly reduced. Even those groups that would still need paper will have the advantage of a consistent rendering in the same document. An Intelligent Document PDF with embedded eDocs XML will become usable by any person or system. The PDF can be viewed by anyone with the Adobe Reader, and the data in it can be extracted and used by any systems using an eDocs registry to map their own data models and vocabularies.

Many of the concepts in this document actually align with the direction of emerging semantic web technologies. The ebXML registry specification is moving toward support for the import of OWL ontologies. RDF support is advancing into the mainstream with implementations like RSS and Adobe's XMP. CCTS is being used in XML vocabularies from RosettaNet and UBL. What's proposed here is a convergence of several of these already accepted paths to interoperability. These practices can be implemented in the normal development of community and industry standards.

Many tangible benefits can be realized from these methods well in advance of large-scale cross-domain interoperability. Even at the agency/ministry level within government, metadata registries can provide a re-usable way to cross lines of business and XML vocabularies. Robust and meaningful document metadata can provide even a single organization with vastly improved access to information. PDF files containing both structured XML and rich content

can provide a common interface to all types of information. Ontology-driven data modeling can help groups develop information models that protect against rapid obsolescence. The short term benefits of these technologies are immediately evident, and the long term potential or true information interoperability is incredible.

References:

• XML 1.0 Recommendation. Extensible Markup Language (XML) 1.0 (Third Edition). T. Bray, J. Paoli, C. M.. Sperberg-McQueen, E. Maler, editors, 4 Feb 2004. http://www.w3.org/TR/REC-xml

• XML Schema Recommendations. H. Thompson, D. Beech, et. al., Part 0-2. http://www.w3.org/TR/xmlschema-0/, http://www.w3.org/TR/xmlschema-1/, http://www.w3.org/TR/xmlschema-2/

• RDF Recommendation. Resource Description Framework (RDF): RDF/XML Syntax Specification, D. Beckett, editor, 10 Feb 2004. http://www.w3.org/TR/rdf-syntax-grammar/

• Web Ontology Language OWL Overview, D. McGuinness, F. van Harmelen, editors, http://www.w3.org/TR/owl-features/

• The Semantic Web, M. Daconta, L. Obrst, K. Smith, 2003, Wiley Publishing, inc.

• ebXML, http://www.ebxml.org/

• Enhancing ebXML Registries to Make them OWL Aware, A. Dogac, Y. Kabak, G. Laleci, C. Mattocks, F. Namji, J. Pollock, www.srdc.metu.edu.tr/webpage/ publications/2004/_DAPD_ebXML-OWL.pdf

• XML Data Package (XDP) Specification 2.0, October 2004, Adobe Systems, Inc., http://partners.adobe.com/public/developer/en/xml/xdp_2.0.pdf

• XML Forms Architecture (XFA) Specification 2.2 Draft, October 2004, Adobe Systems, Inc., http://partners.adobe.com/public/developer/en/xml/xfapecification_2.2_draft.pdf