OASIS ebXML Registry

Proposal: ebXML Registry as a Web Service

Category: New functionality to draft specifications

Date: July 9, 2001

Author: Farrukh Najmi

Status of this Document

This document is a draft proposal whose purpose is to solicit additional input.

1 Abstract

This document proposes focused enhancements to the ebXML Registry Services specification that will allow the ebXML Registry services to be accessible as a set of abstract web services with concrete normative bindings specified for ebXML Messaging Service and SOAP.

Currently the only normative access to the ebXML Registry is over the ebXML Messaging Service. What is lacking is a clean separation between an abstract service interface specification which can be bound to multiple concrete technology specific bindings (e.g. ebXML Messaging Service)

The proposal allows more flexibility and ease of access to clients by defining a second normative interface to the ebXML Registry that is based on the widely adopted SOAP protocol.

2 Motivation

The primary motivation behind this proposal is to further ebXML Registry adoption. It is our assertion that adoption is furthered by:

1. Enabling ease of use of ebXML Registry by clients
2. Enabling additional technology bindings
3. Aligning with emerging and de facto standards

ebXML Registry adoption may be measured in the number of operational ebXML Registries. Currently this number is one. We would like it to higher.

2.1 Enabling Ease of Use

Currently, an ebXML Registry client must use an ebXML Messaging service to interact with an ebXML Registry. This requires that the client have access to ebXML Messaging Service infrastructure. This may become a barrier to ebXML Registry adoption.

Making ebXML Registry available as an abstract web service with additional technology bindings (e.g. SOAP) gives clients more options to interact with an ebXML Registry.

A normative SOAP binding is proposed since SOAP has considerable mind share and adoption and has in fact been adopted by the ebXML Messaging Service itself. Numerous tools exist that make it very simple for clients to access any SOAP based web service.

2.2 Aligning With Emerging and De Facto Standards

Much has happened in the industry and standards space since ebXML Registry V 1.0 was developed:

1. XML Schema is now a W3C recommendation
2. SOAP Version 1.2 and XML Protocol Abstract Model Working Drafts have been published within W3C
3. WSDL has been submitted as a W3C note
4. Web Services are a de facto standard way of providing services such as the ebXML Registry over the web

The proposal will align ebXML Registry with all of the above standards and trends in the industry and thus further adoption.

3 Proposed Deliverables

The following concrete deliverables are proposed:

1. XML Schema definition for [ebRIM] and [ebRS] with full support for XML namespaces, data types, constraints etc. This replaces Registry.dtd
2. Abstract service definition in WSDL
3. Concrete SOAP binding for abstract services in WSDL

58 # 4 Use Cases

59 ## 4.1 SOAP Based Access of ebXML Registry

60 An IT shop wants to write a client program to use the ebXML Registry. They do
61 not have the knowledge or infrastructure for using an ebXML Messaging service
62 to access the registry. However, they have the knowledge to use raw SOAP to
63 access ebXML Registry over SOAP. They use the SOAP binding to ebXML
64 Registry to write a custom SOAP client for the ebXML Registry.

65 ## 4.2 Automatic Client Stub Generation

66 The same IT shop now has access to a WSDL compiler that can automatically
67 generated stubs for accessing the SOAP based ebXML Registry services. The
68 stubs provide simplified access to the ebXML Registry in C++ or Java. The client
69 programmer does not even need to know SOAP. All SOAP specific details are
70 hidden in the bindings generated by the WSDL compiler.

71 ## 4.3 Support for Other Technology Bindings

72 The ebXML Registry team may define additional technology bindings for the
73 abstract services defined by this proposal beyond ebXML Registry and SOAP.
74 For example, an IIOP binding may be defined. These bindings could be layered
75 easily on top of the abstract service definitions in WSDL.