OASIS ebXML Registry

Proposal: Service and Service Binding as Registry Objects

Category: New functionality to draft specifications

Date: September 25, 2001

Author: Sanjay Patil

Version: 0.3

Status of this Document

Date	Version Description
9/4/01	Submitted draft proposal to RAWS sub team soliciting additional input.
9/15/01	Update the document with outcome of RAWS sub team discussion on the proposal in 9/7/01 con-call

1 Abstract

This document outlines solution for inclusion of Business Service and its binding to particular technical interface as first class objects in the Registry.

2 Motivation

In order to be readily useful, a registry needs to have first class support for the common use cases. Business software architectures commonly make use of Service interfaces for software components that are either part of the same system or some other business entity. The dynamics of such Service oriented e-Business software systems largely depends upon an ability to publish and retrieve the Service interfaces and other related objects. In order to enable OASIS ebXML Registry to be readily useful in such scenarios, certain objects need to be inherently supported by the Registry such as Service, Service Binding

of the Service to a particular technical specification along with object holding other runtime information.

2.1 Inherent support for Service and Binding for Services

OASIS ebXML RIM currently provides support for Service by means of the generic ExtrinsicObject model. However, the very common usage of Service and Service Binding objects justifies supporting them as first class objects.

3 Use Cases

3.1 Publishing Business Service and Service Bindings

A client program wants to find out Services that have compatible server side interfaces so that it can automatically communicate with such services. Software runtime components that are prospective caterers to the current client's needs have already published the information about the Services they offer as well as technical specification about how these Services can be invoked in a Registry.

4 Proposed Deliverables

Supporting the Service and Service Bindings require addition of some new objects to the information model. Here is a use case scenario that identifies the different entities involved, so that we can easily formulate the new object types.

A business organization provides an OfficeProductsPurchasing "Service" to its internal users. The service can be accessed by individuals working in the organization using the web interface. The same service can also be accessed by automated purchasing processes in the organization. The web interface relies on a SOAP/WSDL based "Service Binding" of the Purchasing service. The Purchasing service also provides an IIOP based "Service Binding" for the automated business processes in the organization.

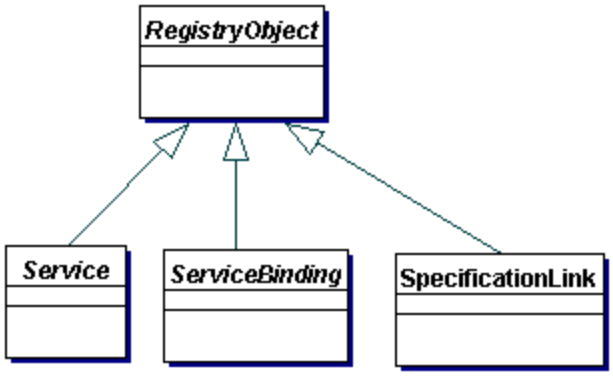
The SOAP/WSDL based service binding makes use of the SOAP, WSDL and HTTP technical "Specifications". Each of these specifications requires a set of runtime parameters ex. HTTP URL. The set of runtime parameters for each technical specification can be perceived as a "Specification Link" between the technical specification and the instance of service binding.

The above use case identifies the following objects that can be added to the RIM.

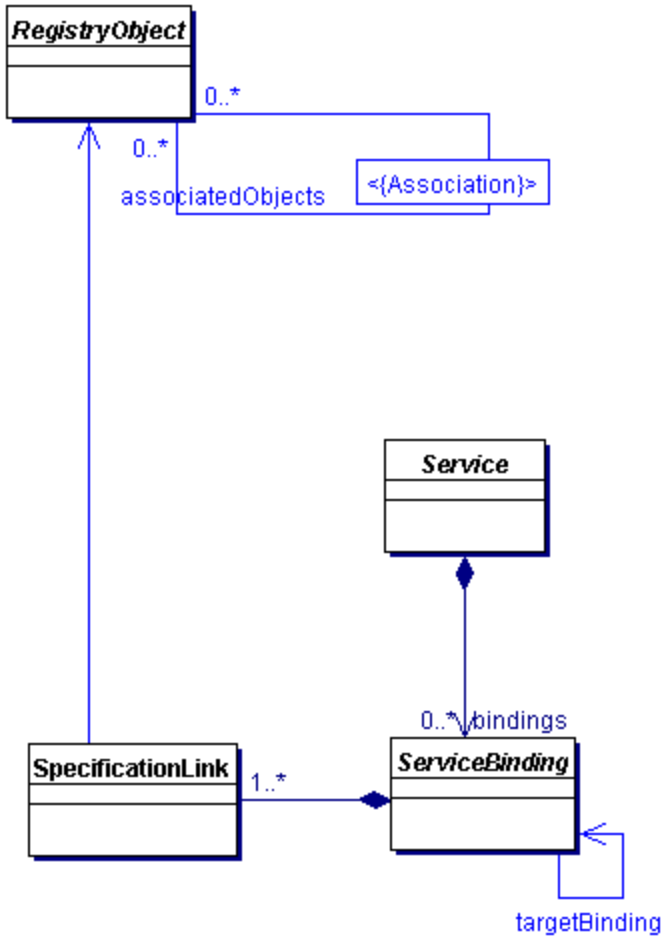
1. Service: ex. Purchasing service
2. Service Binding: ex. Web interface
3. Specification Link: ex. URL, name-value parameters to be specified along with the URL

Based on the above, the following concrete deliverables are proposed:

1. Addition of Service, ServiceBinding and SpecificationLink types to ebXML RIM
2. APIs for publishing and access to the Service and ServiceBinding objects.



As shown in the above diagram, Service, ServiceBinding and SpecificationLink objects can be RegistryObject instances.



4.1.1 Class Service

Service instances are RegistryObjects that provide information on services (e.g. web services).

4.1.1.1 Method Summary

In addition to its attributes, the Service class also defines the following methods.

Method Summary of Package	
`Collection`	**`getServiceBindings`**`()` Get the collection of ServiceBindings instances defined for this Service.

4.1.2 Class ServiceBinding

ServiceBinding instances are RegistryObjects that represent technical information on a specific way to access a specific interface offered by a Service instance. A Service has a Collection of ServiceBindings.

The description attribute of ServiceBinding provides details about the relationship between several specification links comprising the Service Binding. This description can be useful for human understanding such that the runtime system can be appropriately configured by the human being. There is possibility of enforcing a structure on this description for enabling machine processing of the Service Binding, which is however not addressed by the current document

4.1.2.1 Method Summary

In addition to its attributes, the Service class also defines the following methods.

Method Summary of Package	
`Collection`	**`getSpecificationLinks`**`()` Get the collection of SpecificationLink instances defined for this ServiceBinding.

4.1.3 Class SpecificationLink

A SpecificationLink provides the linkage between a ServiceBinding and one of its technical specifications that describes how to use the service using the ServiceBinding. For example, a ServiceBinding may have a SpecificationLink instances that describe how to access the service using a technical specification in form of a WSDL document or a CORBA IDL document.

4.1.3.1 Attribute specificationObject

A SpecificationLink instance must have a specificationObject attribute that provides a reference to a RegistryObject instance that provides a technical specification for the parent ServiceBinding. Typically, this is an ExtrinsicObject instance representing the technical specification (e.g. a WSDL document).

4.1.3.2 Attribute usageDescription

A SpecificationLink instance may have a usageDescription attribute that provides a textual description of how to use the optional usageParameters attribute described next.

4.1.3.3 Attribute usageParameters

A SpecificationLink instance may have a usageParameters attribute that provides a collection of Strings representing the instance specific parameters needed to use the technical specification (e.g. a WSDL document) specified by this SpecificationLink object.

5 Issues

The only issue that was debated in a significant way, and did not get complete consensus was a non-technical issue. At issue was the concern expressed by a couple of team members that this proposal is introducing functionality that would duplicate UDDI functionality and cause confusion in the industry. Other members of the team felt that registration of web services is a common use case that the ebXML Registry must address in a simple, direct and effective manner.