

**Title: The RIM should accommodate classification schemes****Author:** Len Gallagher**References**

- 1) ebXML Registry Information Model v1.0 (8 May 2001)  
<http://www.ebxml.org/specs/ebRIM.pdf>
- 2) ebXML Registry Services Specification v1.0 (10 May 2001)  
<http://www.ebxml.org/specs/ebRS.pdf>
- 3) OASIS Registry/Repository technical specification (20 December 2000)  
<ftp://xsun.sdct.itl.nist.gov/regrep/OasisRegrepSpec.pdf>
- 4) OASIS ebXML Registry Issues List ( June 2001)  
Soon to be available from the OASIS ebXML Registry home page.
- 5) Comments on Classifications, Len Gallagher (5 February 2001)  
<http://lists.ebxml.org/archives/ebxml-regrep/200102/pdf00006.pdf>

**Background**

The ebXML Registry Information Model (RIM) supports the notion of classification of registered objects. It does so by allowing registration of each node of a classification scheme as an intrinsic object in the registry and by allowing a registry entry to reference that node through a classification instance (*cf* Figure 1 of [1] page 10). This model also supports classification of registered objects by allowing a registry entry to be associated with a collection of slots that indirectly reference an externally specified classification scheme. Section 11 of [1] presents these two different alternatives for supporting classification of registered objects by standardized taxonomies.

The existing ebXML approach to classification and classification schemes is flawed in several ways. First, there shouldn't be two different ways to handle exactly the same problem. The user of the Registry should not be forced to use two completely different queries just to identify all registry entries of registered objects that are categorized by a node of a classification scheme, some of which may have been classified by classification instances and others may have been classified by a special use of slots. The Slot approach was devised as an attempted fix for problems with classifications [5] discussed last February at the ebXML meeting in Vancouver and identified as issues for resolution [4] in Phase II of the ebXML process.

The following section on Use Cases identifies examples of situations where the current specification of classifications is inadequate. It is followed by a section for a proposed revision of the Classification submodel in the RIM that allows specific registration of classification schemes, combines the two existing approaches to classification in RIM, and avoids the identified problems.

The Proposal section is an outline of the steps that would be necessary to modify references [1] and [2] to support the revised Classification submodel. Completion of these steps would give us a much more flexible method for defining, sharing, and using classification schemes in Registry implementations.

**Use Cases**

Use Case 1 – A user wishes to register their favorite classification scheme and submit it to the Registry's repository for safe-keeping. In the existing ebXML RIM and RS, there is no way to do this in an intuitive easy way. Instead, this user will have to submit and register each node of the classification scheme separately.

Requirement 1a -- The RIM should define a DTD (or XML schema) for a ClassificationScheme so that this user can submit a complete classification scheme to the Registry in a straight-forward manner (*cf* Section 7.5 of [3]).

Requirement 1b -- The RS should specify a service for defining a new classification scheme (*cf* Section 8.7 of [3]).

Use Case 2 -- The above user has managed to register all of the nodes of a favorite classification scheme. By convention, the root node is considered as representative of the entire classification scheme. However, there is no way in the existing RS specification to retrieve the entire classification scheme from the root node as a structured XML document. Instead, the GetClassificationTreeRequest (*cf* Section 8.1.2 of [2] ) returns only a set of ClassificationNode instances as its Response (*cf* Appendix A of [2] lines 2990-2995). The user would have to reconstruct the tree-structure of the original classification scheme from the parent attribute of each classification node in the set.

Requirement 2a -- The RIM should not depend upon the root node of a classification scheme to handle the metadata relevant to the entire classification scheme. Instead, it should be possible to register the entire classification scheme with its own metadata.

Requirement 2b -- The RIM should treat a classification scheme as a repository item with its own registry entry. The contentURI attribute of that registry entry would provide a web resolvable link to the registered classification scheme. Invocation of that URL would return the entire classification scheme as an XML document.

Requirement 2c -- The GetClassificationTreeRequest in the RS specification should be modified to return a tree-structured XML document that validates to a registered ClassificationScheme DTD instead of the existing set of node records.

Requirement 2d -- The ReturnRepositoryItem request in the RS specification (*cf* Section 8.2.8 of [2] and lines 3173-3184) should be modified to return classification schemes as XML structures that validate to a registered ClassificationScheme DTD.

Use Case 3 -- A user wishes to register a new object in this Registry X and then classify it by a classification scheme that exists in some other Registry Y. Right now this cannot be done as a Classification in X without first copying, or re-creating, the nodes one wishes to use from Y. There is no way to create a Classification instance in X unless the nodes of the classification scheme also reside in X.

Requirement 3 -- The RIM should be modified to allow a registry entry to have a Classification instance that references a classification node that exists in some other Registry.

Use Case 4 -- Suppose the United Nations has hired a consultant to help it register the UNSPSC classification scheme (all 10,000+ nodes) in an ebXML Repository. Right now this cannot be done without registering each node separately. Thus each node must carry along all of the required attributes of a RegistryEntry instance, including a 128-bit GUID, and default values for objectType, status, version, stability, and expirationDate. This seems like information overload for a node that consists only of 3 important values, i.e. nodeName, nodeCode, and parent.

Requirement 4 -- The RIM should be modified to relax the requirement that every ClassificationNode instance be a RegistryEntry (*cf* Figure 2 of [2] on page 13). Instead, a classification node should be represented as a simple element within a registered classification scheme.

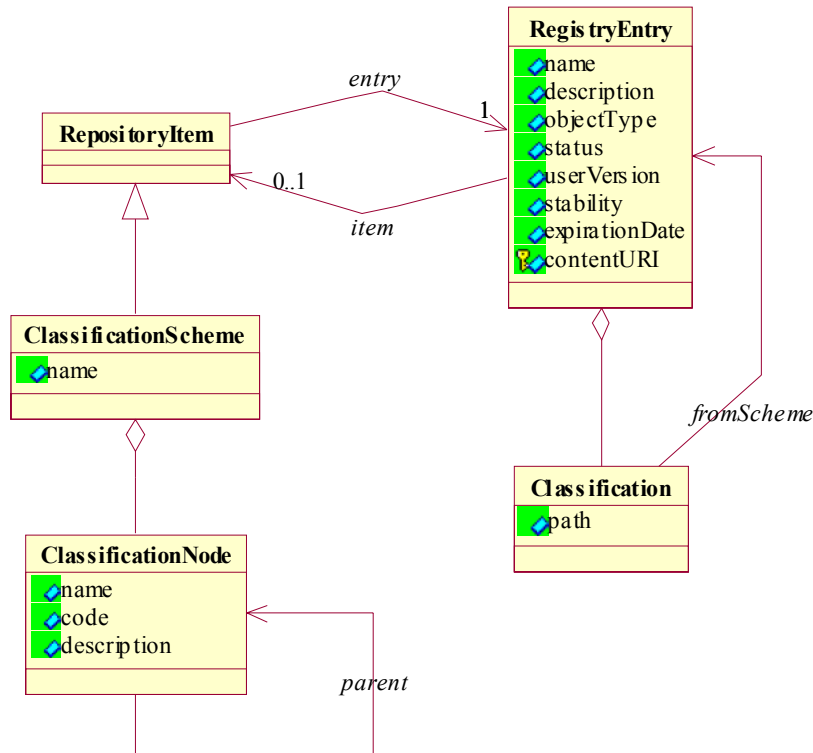
Use Case 5 -- Suppose an industry consortium has identified a small collection of classification schemes that it wants to use to classify the items to be registered in its Registry. The classification nodes are all registered in the Registry and registry entries are created for each repository item. Now they want to create Classification instances for each registry entry. In the existing RIM, each such classification must carry along all of the required attributes of a RegistryEntry instance, including a 128-bit GUID, and default values for objectType, status, version, stability, and expirationDate. This seems like unnecessary information overload, since a classification instance only need consist of a reference to one of the registered classification nodes.

Requirement 5 -- The RIM should be modified to relax the requirement that every Classification instance be a RegistryEntry (*cf* Figure 2 of [2] on page 13). Instead, a classification should be represented as a simple reference to a registered classification scheme and to a specific node of that classification scheme.

## Proposed Classification Submodel

I propose a revision of the ebXML RIM to remove the requirement that a Classification instance be hard-wired to a ClassificationNode instance in the same Registry. Instead, a classification would reference a classification scheme by pointing to a registry entry for that classification scheme. Then the path attribute of the classification would identify a node of the referenced classification scheme. The classification scheme itself could be a repository item in some other Registry.

The following figure is derived from the Classification submodel defined in the old OASIS Registry/Repository specification [3]. It carries along the class names and attributes defined in ebXML RIM, but it avoids nearly all of the problems identified above. The only other modification from ebXML is that we assume the contentURI attribute of the ExtrinsicObject class is moved to the RegistryEntry class.



**Classification Submodel**

1) The path attribute of a classification identifies a classification node of a registered classification scheme in this or in some other registry. For example, when using the UNSPSC classification scheme the path would be the 6-digit UNSPSC code. The classification would point to a local registry entry for the UNSPSC classification scheme and that registry entry would point to the actual classification scheme. The classification scheme itself may reside elsewhere, but its XML representation would always be available via the contentURI attribute of the registry entry.

2) Since a classification scheme is a repository item, it is linked to exactly one registry entry in the same Registry. This is the registry entry that accompanied the classification scheme when it was first submitted to that Registry. Other registry entries, in this or in other Registries, could then point to this classification scheme.

3) A registry entry for a classification scheme is not required to point to a repository item. This is because the classification scheme may be in a different Repository. This is a relaxation of the requirement in [1] that there be a one-to-one correspondence between a registry entry and a repository item (*cf* Section 6.1, last sentence).

4) The *item* link from RegistryEntry to RepositoryItem and the *entry* link from RepositoryItem to RegistryEntry are not directly navigable by an end user. Instead, an end user must use either the contentURI attribute or a Registry Service request to access a repository item.

5) A Classification instance has only the identified attributes. It is not an ImplicitObject instance as required by the current ebXML RIM. A classification need not have a 128-bit identifier and need not carry along all of the implied registration information of a RegistryEntry (*cf* Figure 2, page 13, of [1]).

6) A ClassificationNode instance has only the identified attributes. It is not an ImplicitObject instance as required by the current ebXML RIM. A classification node need not have a 128-bit identifier and need not carry along all of the implied registration information of a RegistryEntry (*cf* Figure 2, page 13, of [1]).

7) A registered classification scheme will always be retrievable either directly, using the contentURI of its registry entry, or indirectly, by sending a Registry Services request to the registry where the classification scheme exists as a repository item. In a cooperating federation of Registries, a Registry Services request made to one Registry could be implemented globally, thereby making the union of all repository items available through a single Registry Services call with the implementation details hidden from the end user.

## Proposal

Make the following modifications to RIM [1] and RS [2]. NOTE: This is just the outline of a complete proposal. The author promises to supply the necessary details if there is general agreement to move in this direction.

- 1) Add a new class, named ClassificationScheme, to RIM. Treat an instance of ClassificationScheme as a repository item that will have a registry entry to present its metadata. Add a new Section 6.x to RIM to describe the semantics of a classification scheme and the classifications that reference it (*cf* Section 2.4 of [3]).
- 2) Modify Section 11.1 of RIM, to describe the attributes of ClassificationScheme and of ClassificationNode (*cf* Section 3.3 of [3]). Re-define LevelValuePair from [3] using appropriate Path terminology.
- 3) Incorporate the Classification Submodel figure discussed above into Figure 1, page 10, of RIM. Modify Section 6.1 of RIM to clarify the relationship between RegistryEntry and RepositoryItem (*cf* Section 2.2 of [3]).
- 4) In RIM figure 2, page 13, remove ClassificationNode and Classification as subtypes of IntrinsicObject. Instead make them subtypes of RegistryObject.
- 5) Modify Section 7.3.2 of RIM to add “ClassificationScheme” as a new objectType. Consider deletion of “ClassificationNode” as a supported objectType.
- 6) Modify Section 11.2.1 of RIM to leverage the semantics of a classification scheme and the classifications that reference it. Modify Section 11.4 of RIM to give examples of how to use classification schemes. Delete Section 11.4.2 since all classifications will now be represented by Classification instances.
- 7) Add a ClassificationScheme DTD to RS (*cf* Section 7.5 of [3]). Add a ClassificationSchemeInstance XML ELEMENT definition to RS (*cf* Sections 6.13 to 6.15 of [3]). Add a new service to RS allowing a user to define a new classification scheme (*cf* Section 8.7 of [3]).
- 8) Delete Section 8.1.1 of RS. Instead, one can get a list of available classification schemes by a direct query on the RegistryEntry class with objectType = “ClassificationScheme”. In addition, to query classification schemes that are repository items in this registry, we could add ClassificationSchemeQuery as a new type of FilterQuery in Section 8.2.1 of RS.
- 9) Modify Section 8.1.2 of RS to ensure that the Result of a GetClassificationTreeRequest is an XML document that validates to the ClassificationScheme DTD defined above. Consider adding other alternatives for navigating a classification scheme hierarchy and returning specified parts of the tree.
- 10) Make minor modifications to Sections 8.2.2 and 8.2.4 of RS to accommodate the revised defn of Classification.