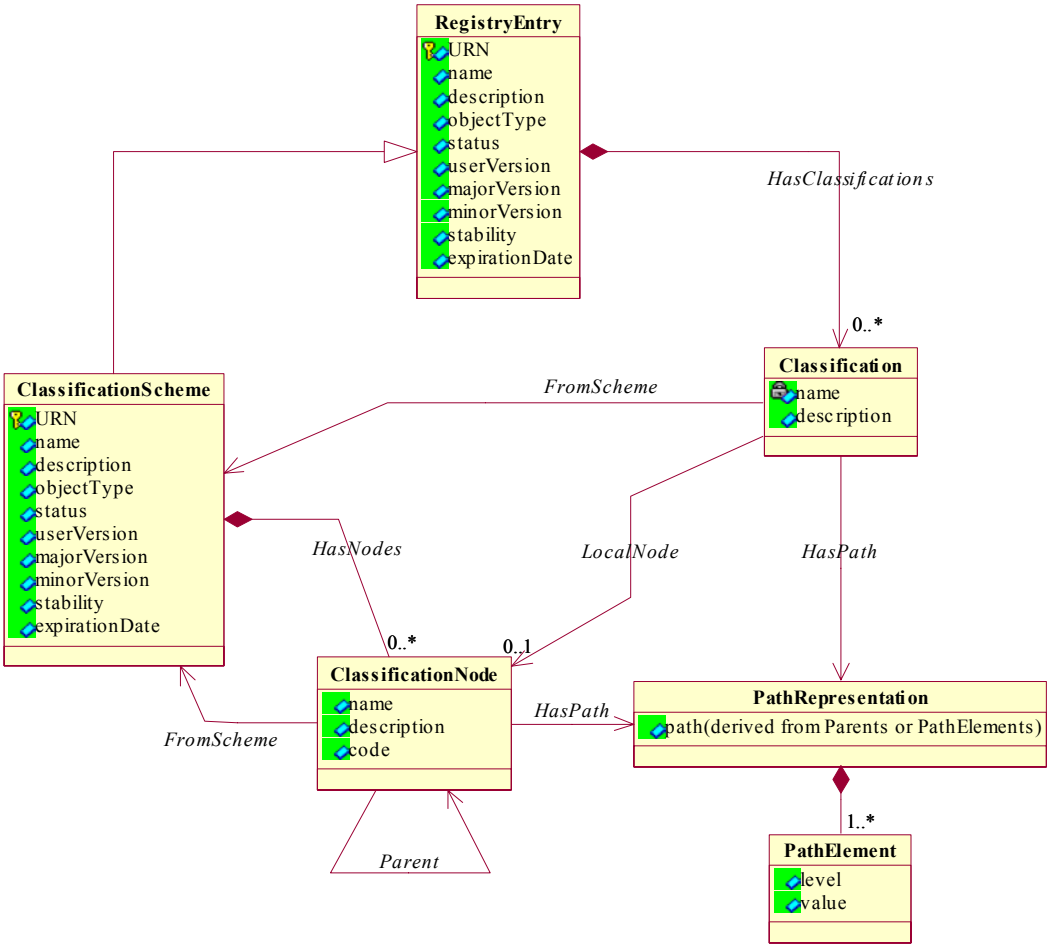
UML Query View for both Internal and External Classifications – 10 Sept 2001



Assumptions

1) A Classification can be either an InternalClassification or an ExternalClassification, but the query writer need not know which it is. An InternalClassification will have a LocalNode association to ClassificationNode; an ExternalClassification will not.
2) The name attribute of Classification has no role to play in this presentation, so it is "locked", i.e. not visible.
3) If the HasNodes association is empty, then the ClassificationScheme instance describes an external classification scheme; otherwise, it describes an internal classification scheme.
4) Every RegistryEntry instance has a required URN that is a unique identifier for that instance. If the RegistryEntry points to a repository item in an external Registry, then the RegistrationAuthority will enforce a requirement that the local Registry use the same URN as does the external Registry.
5) The Classification class has a method defined, e.g. FromScheme(), that identifies a ClassificationScheme instance in the local Registry. For internal classifications, FromScheme() is derived from LocalNode.FromScheme().
6) Every Classification and every ClassificationNode maps to a set of PathElement instances that determine the full path of a node from some external or internal classification scheme. The set of PathElement instances is represented as a "path" character string visible as a "special" attribute of ClassificationNode and of Classification.
7) The Classification class has a method defined, e.g. HasPath(), that identifies a "path" to represent the node. For internal classifications, HasPath() is derived from LocalNode.HasPath().
8) The special "path" node in both Classification and ClassificationNode can be queried by XPATH syntax, which is not yet specified in RIM or RS. Alternatively, the "path" can be queried by a PathElementFilter, which is also not yet specified in RIM or RS.

Conclusions

If the above assumptions hold, then it is possible to specify a RegistryEntryQuery whose XML HasClassificationBranch element (cf ebRS Section 8.2.2) is identical for both ExternalClassification and InternalClassification instances.

If the above assumptions hold, then I think it is possible to represent all of the Browse and Drill Down queries (cf ebRS Section 8.1) as a revised and extended ClassificationNodeQuery (cf ebRS Section 8.2.4).

Revised ebRS Syntax

In the syntax for querying repository items that have been classified (e.g. ebRS Section 8.2.2), I propose that the HasClassificationBranch element be replaced by the following (or equivalent XML schema) in accord with the above UML view.

```
<!ELEMENT  HasClassificationBranch
   (       ClassificationFilter?,
           FromSchemeBranch?,
           HasPathBranch?,
           LocalNodeBranch?          )>

<!ELEMENT  FromSchemeBranch
   (       ClassificationSchemeQuery          )>

<!ELEMENT  HasPathBranch
   (       PathFilter | XPATHnodeQuery  |  PathElementFilter          )>

<!ELEMENT  PathFilter  ( Clause )>

<!ELEMENT  XPATHnodeQuery   TO BE SPECIFIED as special XPATH synatx >

<!ELEMENT  PathElementFilter  ( Clause )>

<!ELEMENT  LocalNodeBranch
   (       ClassificationNodeFilter    )>
```

Examples

It doesn't matter if NAICS is an internal or external classification scheme. A RegistryEntryQuery can retrieve all RegistryEntry instances classified as NAICS with code=33611 as follows:

```
<RegistryEntryQuery>
  <HasClassificationBranch>
     <FromSchemeBranch>
        <ClassificationSchemeQuery>
           <RegistryEntryFilter>
                URN EQUAL "urn:ntis-gov:naics:1997"
           </RegistryEntryFilter>
        </ClassificationSchemeQuery>
     </FromSchemeBranch>
     <HasPathBranch>
        <PathFilter>
                path = "33611"
        </PathFilter>
    </HasPathBranch>
  </HasClassificationBranch>
</RegistryEntryQuery>
```

The above RegistryEntryQuery doesn't include any shortcuts. However, if we assume the existence of two derived attributes in a Classification instance, i.e. path and schemeURN, then the above query could be shortened to:

```
<RegistryEntryQuery>
  <HasClassificationBranch>
    <ClassificationFilter>
          schemeURN EQUAL "urn:ntis-gov:naics:1997"
          AND
          path = "33611"
    </ClassificationFilter>
  </HasClassificationBranch>
</RegistryEntryQuery>
```

In general, the judicious specification of popular derived attributes in many of the RIM classes could substantially shorten the specification of many filter queries.

The advantage of the longer, fully spelled out query, is that it provides much more flexibility. For example, suppose someone wishes to identify all items that have been classified by a classification scheme owned by OASIS, but they have no idea what the URN or name of the classification scheme is. The following should do the trick:

```
<RegistryEntryQuery>
  <HasClassificationBranch>
     <FromSchemeBranch>
       <ClassificationSchemeQuery>
         <SubmittingOrganizationBranch>
           <OrganizationFilter>
              URN EQUAL "urn:org:oasis-open"
           </OrganizationFilter>
         </SubmittingOrganizationBranch>
       </ClassificationSchemeQuery>
     </FromSchemeBranch>
  </HasClassificationBranch>
</RegistryEntryQquery>
```

Suppose the query writer doesn't know the "code" or "path" attribute of the classification scheme, and instead wants to find all repository items classified by a "local" classification scheme that has any of the words "truck" or "tractor" in the name attribute of its corresponding ClassificationNode instance. The following should do it:

```
<RegistryEntryQuery>
  <HasClassificationBranch>
     <LocalNodeBranch>
       <ClassificationNodeFilter>
             name CONTAINS "truck"
             OR
             name CONTAINS "tractor"
       </ClassificationNodeFilter>
     </LocalNodeBranch>
  </HasClassificationBranch>
</RegistryEntryQquery>
```

NOTE: The above query would not work in general for external classifications because the local Registry implementation may not know how to access the external classification scheme. It knows the path of the classification but not necessarily the spelling of the various node names represented by codes in that path.