

OASIS ebXML Registry Technical Committee -- Query sub-team

Subject: getPathElements() method on ClassificationNode class

Version: 1

Author: Len Gallagher

Date: 9 October 2001

Type: ebRIM Proposal

Introduction

At last week's Query teleconference we discussed the FilterQuery proposal and its three different options for a HasPathBranch element. The reasons why three "overlapping" alternatives are defined is given in my earlier email to this list:

<http://lists.oasis-open.org/archives/regrep-query/200109/msg00079.html>

We've all agreed that the third alternative, PathElementsFilter, is not yet complete in that it requires a new method in ebRIM that would specify just exactly what the "path elements" are. Once we know what they are, they can be queried by our existing Clause syntax specified in Section 8.2.10 of ebRS v1.0. Recall that this Clause syntax provides logical connectors (AND, OR) and comparison predicates for numeric and string data types. However, it does NOT define comparison predicates on non-primitive data types, so it cannot be used as currently specified for XPATH type expressions, or for set, list, or sequence data types.

The purpose of a new getPathElements() method is to represent the "path" of a ClassificationNode instance as a set of instances of a derived class, call it PathElements. This derived class will have attributes defined with primitive data types so that they can be queried by existing Clause syntax.

The proposal below defines a getPathElements() method in ebRIM that acts on a given ClassificationNode instance to return the set of ClassificationNode instances that precede the given node in the classification scheme hierarchy. Technically, each such instance would have all of the attributes and methods of any other ClassificationNode instance, but for now the FilterQuery proposal only assumes that one attribute, i.e. "code", and one method, i.e. getLevelNbr(), are visible.

The PathElementsFilter in ebRS renames the result of the getLevelNbr() method as "level" and renames the code attribute as "value". It assumes that both "level" and "value" are strings. There is no real reason for these name changes; they could have been called "levelNumber" and "code" instead, with levelNumber treated as an integer type.

Proposal

1. In ebRIM version 1.1, section 10.2.4, Method Summary

Add the following row to the table Method Summary of ClassificationNode:

ReturnType	Method
Collection	<code>getPathElements()</code> Get the collection of ClassificationNode instances that precede the given node in the classification scheme hierarchy. This method returns an empty collection if the given node has no parent node in the classification scheme hierarchy. The classification scheme itself is not a member of this collection since it is not a ClassificationNode instance.

Further Discussion

I think the above is sufficient as a definition, but if people want we could include a more formal mathematical definition as follows:

A classification scheme is a pair (S, \leq) where S is a set of nodes and \leq is a partial ordering over S , with the additional requirement that the set of predecessors of every node is linearly ordered by the partial ordering and has a unique first element. Every node $x \in S$ is assigned a level number by the expression $\text{Level}(x) = \text{Card}(\text{Pred}(x)) + 1$, where $\text{Pred}(x)$ is the set of predecessors of x under the partial ordering and $\text{Card}(\text{Pred}(x))$ is the cardinality of that set. All nodes that have no node predecessors are at level 1.

It is sometimes helpful to think of S as the Root of the classification scheme hierarchy, but it is not itself a node and is not a participant in the partial ordering, so one must be very careful when visualizing S as playing the role of root of a classification scheme hierarchy. If one chooses to visualize S attached at the top of the hierarchy, then it would be consistent to say it is at level zero. But it is still not a node and methods on the ClassificationNode class cannot be applied to it. In particular, S is NOT returned as a node in the `getPathElements()` method, even though the “parent” attribute of other ClassificationNode instances may or may not reference it!

Our model allows a classification scheme definer to name the classification scheme, name each node in the scheme, and assign a code value for each node in the scheme. With the above definition of levels, each node has a level. Some classification schemes allow the scheme definer to also name the levels (e.g. Genus/Species and UNSPSC). At present our model does not allow scheme definers to name the levels of a classification scheme. I see this level naming capability as a very desirable upward compatible extension that we should consider for version 3 of ebRIM. In that case we may want a PathElementsFilter to have three queryable attributes, namely: levelNumber, levelName, and value (or code). Since this potential extension to named levels is upward compatible, it is not necessary that we pursue it for version 2.