

OASIS ebXML Registry Technical Committee -- Query sub-team

Subject: getLevelNumber() method on ClassificationNode class

Version: 1

Author: Len Gallagher -- for the Query sub-team

Date: 5 October 2001

Type: ebRIM Proposal

Introduction

At today's Query teleconference we were trying to decide if the Browse and Drill Down Query Support in ebRS Section 8.1 was 100% superseded by the capabilities of FilterQuery in Section 8.2. I said: "No – not quite". There is still the capability of Browse and Drill Down to take a ClassificationNode instance as input and return all nodes in a specific number of levels below the input node. For example, find all nodes in the next 3 levels beneath the given node.

Right now a ClassificationNodeQuery cannot operate on a specific number of levels because a ClassificationNode instance has no attribute or method to return its level in the hierarchy. We all agreed that we need such a capability. We also all agreed that it was much better to introduce this capability in RIM as a method rather than as an attribute.

I volunteered to write a proposal for a getLevel() method to be included as an additional method for the ClassificationNode class (cf ebRIM section 10.2.4). NOTE: I'm going to use author's prerogative to change its name to getLevelNumber() in order to emphasize that an integer number is returned, not a string!

Proposal

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

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

Return Type	Method
Integer	getLevelNumber() Get the level number of this ClassificationNode in the classification scheme hierarchy. This method returns a positive integer and is defined for every node 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.

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. Since it is upward compatible it is not necessary that we pursue it for version 2.