RDF Difference Models

Representing the Difference between two RDF Models

Arnold deVos
adv@langdale.com.au

1. Introduction

The purpose of this note is to propose a representation for the difference between two RDF [1] models. There are potentially many such representations, but the chief application for the present proposal is to manage the evolution of a base model through multiple revisions.

1.1 Relationship to CIM/XML

The CIM/XML schema [2] and syntax [3] define an RDF vocabulary and syntax for power system models. The present proposal addresses an immediate need in the power industry to communicate changes to CIM/XML models between utilities.

However, the ability to deal with changes to models or differences between models is likely to be part of many RDF-based systems. Therefore, a generic approach is proposed.

2. Use Case

In a simplified use case, the participants are Regional Energy Co. and Network Power Co:

- Each participant has a copy of a power system model, B1.
- Regional Energy Co. updates B1, to reflect forthcoming power system modifications, producing B2.
- Regional Energy Co. sends the differences between B1 and B2 to Network Power Co. as a *difference model*.
- Network Power Co. reviews and validates the difference model.
- Network Power Co. merges the difference model with its copy of model B1, to produce B2.

An alternative would have been for Regional Energy Co. to simply send Network Power Co. a copy of B2. However, B2 is a very large model and it is not feasible to validate it in any reasonable period of time. Validation is not entirely automated, but involves analysis by experts. Indeed, the best validation strategy for B2 may be to compare it to the previously validated B1. This brings us back to the need for a difference model.

A more complicated use case would involve more than two participants. Several peers of Regional Energy Co. would contribute difference models to Network

Power Co. This use case would introduce issues of parallel model changes and concurrency conflict.

3. Requirements

Given two RDF models, B1 and B2, called *base models,* the requirement is for a *difference model* that:

- Represents the differences between the two base models
- Is itself an RDF model (a collection of RDF statements) and therefore can be processed by RDF infrastructure.
- Can be combined with either base model to construct the other.
- Efficiently represents a small difference between two large base models.
- May contain information about itself such as authorship, purpose and date.
- May contain information to protect against conflicts arising when two difference models are created concurrently from the same base model.

4. Definition

Given base RDF models, B1 and B2, a difference model will made up of four groups of statements:

- *Header statements*, consisting of statements about the difference model, itself.
- F*orward difference statements*. The statements found in B2, not found in B1.
- R*everse difference statements*. The statements found in B1, not found in B2.
- P*recondition statements*. A subset of the statements found in both B1 and B2 considered to be dependencies of the difference model in some application defined sense. (See discussion below.)

Any or all of these sets of statements may be empty.

The difference model itself is represented by a resource. It is conventional to use the URI of the document itself for this resource.

The following properties apply to the difference model resource, where the dm namespace is TBD:

- dm:forwardDifferences is a property of the difference model whose value is a collection of statements (ie resources of type rdf:Statement) representing the forward difference statements.
- dm:reverseDifferences is a property of the difference model whose value is the collection of reverse difference statements.
- dm:preconditions is a property of the difference model whose value is the collection of precondition statements.

Header properties also apply to the difference model resource. These may indicate authorship, date and purpose. These properties can be drawn from the

Dublin Core vocabulary or any other convenient schema. The standard header properties for describing a CIM/XML difference model are TBD

4.1 Preconditions and Concurrency

The precondition statements are a subset of both B1 and B2 and carry no difference information. In simple, sequential model revision scenarios they can be omitted.

For a large shared model, sequential revision is not always feasible. Revisions are likely to be constructed concurrently by different participants, without reference to each other. Concurrency issues must be handled, but the conventional database-oriented approach of using locks to detect incompatible concurrent transactions is not feasible on a web-scale.

The precondition statements are an alternative to locks. Informally, they represent the information that would have been read-locked in an equivalent database transaction. Software agents that process difference models can check that the preconditions hold and, if not, warn of incompatible model revisions.

The choice of statements to include as preconditions is application-specific (as is the choice of which information to lock in a database transaction). Preconditions should include statements that would the affect decisions of the agent that produced the model revision.

Note: it is likely that many of the reverse difference statements (statements that were repudiated in the revision) would have affected decisions of the agent in producing the revision. Should there be a way to flag some or all of these statements as preconditions?

5. Syntax

A document or set of documents constitutes a valid representation of a difference model if it:

- Conforms to RDF syntax (possibly with the RDF syntax extension defined below).
- Yields the statements defined in section 3, above, when interpreted according to RDF rules (again, with the possible extension defined below).

The CIM/XML difference model template given below illustrates a specific case of a valid representation.

5.1 Extension

There is no succinct way in standard RDF syntax to represent a set of statements as the object of a property such as dm:forwardDifferences. Therefore we introduce a new rdf:parseType called Statements.

```
<property parseType="Statements">
      <!-- Content: (definition|description)* -->
<property>
```

- The content model of an element with rdf:parseType="Statements" is the same as the content model of the rdf:RDF element.

- The content generates the same RDF statements as if it appeared in an rdf:RDF element.
- The value of a property element with rdf:parseType="Statements" is a collection of resources of type rdf:Statement, representing the generated statements.

This is a proper extension of RDF syntax as anticipated by the RDF specification. However, standard RDF parsers will parse the content as literal XML rather than as statements. To achieve the interpretation specified above this content could be re-parsed by the standard parser in a second pass. Alternatively, a modified RDF parser that directly recognises rdf:parseType="Statements" could be used.

5.2 CIM/XML Difference Model Template

The following is a template for the conventional syntax of a CIM/XML difference model.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
            xmlns:cim="cim-namespace-uri"
            xmlns:dm="difference-model-namespace-uri">
      <rdf:Description about="">
            <!-- Content: (literal-property|resource-property)* -->
            <dm:preconditions parseType="Statements">
                  <!-- Content: (definition|description)* -->
            </dm:preconditions>
            <dm:forwardDifferences parseType="Statements">
                  <!-- Content: (definition|description)* -->
            </dm:forwardDifferences>
            <dm:reverseDifferences parseType="Statements">
                  <!-- Content: (definition|description)* -->
            </dm:reverseDifferences>
      </rdf:Description>
</rdf:RDF>
```

Notes:

- The content models given in comments above are defined in *Simplified RDF Syntax for Power System Model Exchange*.
- The rdf:Description element introduces the resource representing this difference model.
- The about attribute indicates that the difference model URI is the document URI. (Another value could be given.)
- The properties contained in the rdf:Description element form the header statements.
- The core information is divided into three groups of statements, designated by the properties dm:preconditions, dm:forwardDifferences, and dm:reverseDifferences.

6. CIM/XML Naming Issues

Difference models introduce new naming issues to CIM/XML. Previously, an entire CIM/XML model was transferred at once and it was merely required that instance names be internally consistent. (Class, property and enumeration names were required to agree with the CIM RDF schema, of course.)

Now, instance names must agree between two distinct and independently produced models: a base model and a difference model. By extension, the names must agree within an entire family of base models and difference models.

RDF employs a universal naming scheme, the URI, to deal with this type of problem. In CIM/XML it is merely necessary to choose a stable, agreed URI for each power system resource (eg Switch or Transformer). The following guidelines are proposed:

- The owner of the asset is responsible to assign its name. Each utility already has a slab of the universal http namespace from which to allocate names. Names may be allocated from this space or any other pool of URI's that the utility controls, with no danger of conflict with another utility.
- The owner utility may use any conventions they see fit to name assets. They may use the naming systems devised by the CCAPI or IEEE groups, but they are not constrained to do so for CIM/XML interoperability. The use of RDF merely requires the utility to map their chosen naming scheme into URI syntax.
- Utilities are strongly urged to use naming practices that maximise the stability of names over time. It is undesirable to embed any attributes of the asset in its name if those attributes are likely to change.
- Utilities are encouraged to use operational names and not physical names. Utility assets often have two names in common usage. The physical name is typically the serial number found on the asset's nameplate. The operational name designates the role of the asset in the power system. Over time, different physical assets may be placed into service in a particular operational role. For CIM/XML, it is the operational role that matters.

7. Schema for Difference Model

The RDF schema for a difference model is given below. This should be interpreted in conjunction with the difference model definition in section 3.

TBD

8. References

[1] "Resource Description Framework (RDF) Model and Syntax Specification", W3C Recommendation, February 1999, http://www.w3.org/TR/REC-rdf-syntax

[2] "EPRI Common Information Model RDF Schema", December 2000. Available at http://www.langdale.com.au/CIMXML/cimu09a_RDF_001215a_1.zip

[3] "Simplified RDF Syntax for Power System Model Exchange", Arnold deVos November 2000. Available at http://www.langdale.com.au/CIMXML/PSModelExchange.pdf