Meeting Minutes 5 October 2000

Author:
Melanie Nelson (Physiome Sciences Inc.)
Contributor:
Warren Hedley (Bioengineering Institute, University of Auckland)

1 Summary

Check it out — new author! OK, there was some post-processing of the document by the usual rabble but this document marks Melanie's first foray into the new world of XML, which is always a big event for a girl. There is this idea that it always hurts a little the first time, and this can be the case, but after a few documents most chicks find that writing XML can be quite a lot of fun. Of course most guys don't think about much else.

First up, we have Melanie's review of the use of the EBI's bibliographic query service data model as a model for citations in CellML. And then there are some notes on grouping, along with a suggestion for a possible re-naming (for the purposes of discussion) of the concept formerly known as geometry. Finally there a few random notes resulting from a discussion about how metadata should be fitted into the XML.

2 Introduction

This document outlines a proposed method for including reference information in CellML. The data model borrows heavily from the EMBL-EBI's Bibliographic Query Service RFP response to the OMG's Life Science Resource group. The response document can be found at

http://cgi.omg.org/cgi-bin/doc?lifesci/00-01-15

We are not advocating simply using this data model as an existing standard for two reasons:

1. There is no XML serialization of the data model.
2. The RFP response defines a method for setting up a bibliographic collection, and methods to query the collection. We do not need this functionality. We only want to be able to refer to references in someone else's collection.

However, we should adhere to the OMG's data model, in order to make it easier to migrate to a complete serialization of this model if one becomes available from an external group.

3 Overview of Important Portions of the OMG Bibliographic Query Service Data Model

Interested parties who do not want to trust this summary should see sections 3 and 4.1 of the OMG BQS response document.

The important parts of the OMG Bibliographic Service (BQS) data model are the Citation class and its subclasses, and the Provider class and its subclasses.

3.1 Citation Class

The Citation class is a superclass for the following classes of citation types:

1. Book

2. Article (itself subclassed into BookArticle and JournalArticle)
3. Journal
4. Patent
5. WebResource
6. Thesis
7. Proceeding
8. TechReport

However, the Citation class is also instantiable, and contains enough attributes to allow it to be used to store information for citation types not on the above list.

The Citation class and its subclasses have all of the attributes you'd expect to allow the storage of reference information, such as: title, date, authors (as an association from the Provider class), etc. It also has an identifier attribute, which consists of two parts: id and kind. The id part can be the unique identifier for the reference in some collection of references. The kind part can be a string identifying the collection of references for which the id is valid.

3.2 Provider Class

The Provider class is subclassed into Person, Organization, and Service. Providers are used as authors, editors, publishers, and contributors to citations.

4 Use of the OMG BQS Data Model in CellML

The portion of the OMG BQS data model that can be adopted by CellML is shown in Figure 1. If all attributes and associations are made optional, this data model will support both citing a reference by a unique identifier and by including complete reference details. If a user wishes to cite the reference by a unique identifier, he/she can provide only the identifier attribute for the Citation class. As explained in Section 3, this identifier has two parts: an id and a kind. The id is the unique identifier from the citation source (such as a Medline UI), and the kind is the name of the source (such as Medline). If a user wishes to include the complete reference information, he/she uses the other attributes and associations of the Citation class to do so. It is not necessary to provide the identifier attribute in this case.

Note that Melanie has no idea what the cost of a WebResource is meant to be, as documentation available from the OMG does not explain it. Also, the above discussion does not imply any specific serialization of the data model into XML. For instance, in our XML serialization, we may choose to split the two parts of the identifier attribute into two XML elements or attributes. An example of a potential XML serialization is shown in the next section.

5 An Example XML Serialization of the Data Model

The XML snippets in Figure 2 and Figure 3 demonstrate how we might use the OMG BQS data model to cite references by a unique identifier such as a Medline UI. Note that we might want to make a small change to the OMG BQS data model and allow the inclusion of a URL for the reference identifier type. However, we should consider this carefully, as any changes to the data model would make it hard to migrate to a more complete external serialization of the OMG BQS data model if one does become available.

The following XML snippet demonstrates how we might use the OMG BQS data model to cite references by including complete reference information.

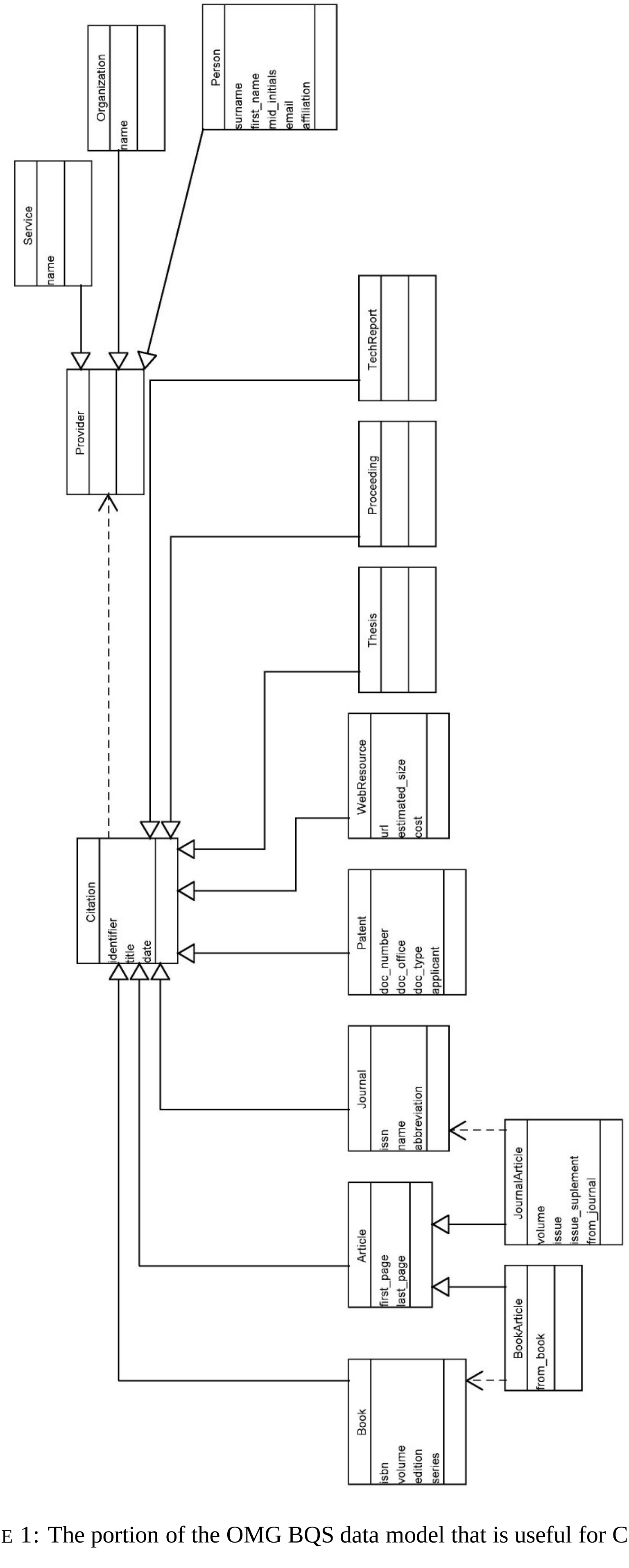

FIGURE 1: The portion of the OMG BQS data model that is useful for CellML.

```
<reference type="primary">
  <identifier>
    <id>20241869</id>
    <kind>Medline</kind>
  </identifier>
</reference>
```

FIGURE 2: An example of how a reference could be cited by an external unique identifier in CellML.

```
<reference identifier="Medline.20241869" type="primary" />
```

FIGURE 3: An equally valid example of how a reference could be cited by an external unique identifier in CellML. This example more closely parallels the OMG BQS IDL serialization.

```
<reference type="secondary">
  <date>2000</date>
  <title>
    Action potential and contractility changes in [Na(+)](i)
    overloaded cardiac myocytes: a simulation study.
  </title>
  <journal_article>
    <volume>78</volume>
    <issue>5</issue>
    <first_page>2392</first_page>
    <last_page>2404</last_page>
    <journal>
      <abbreviation>Biophys J</abbreviation>
    </journal>
  </journal_article>
  <author_list>
    <author>
      <surname>Faber</surname>
      <first_name>G</first_name>
      <mid_initials>M</mid_initials>
    </author>
    <author>
      <surname>Rudy</surname>
      <first_name>Y</first_name>
    </author>
  </author_list>
</reference>
```

FIGURE 4: An example of how complete reference information could be included in CellML. Note that the order of the author list is meaningful, and should be preserved. However, the author is making no statement in the holy war of the relative merits of putting information in XML elements versus attributes.

The reference type attribute in the above examples is storing metadata about whether the reference cited is the primary reference for a model or model component (i.e., the reference in which the model was first described), or a secondary, supporting reference (such as one in which supporting experimental data was published).

6 Geometry??

Aaah ... the concept formerly known as geometry. Basically this word is not really what we want in the specification, because it doesn't really capture what we mean by the "*is-in*" relationship. A quick search of an online thesaurus led to the term "***containment***" as a possible replacement. Feedback welcome.

Note that a group can have exactly one *encapsulation* relationship and/or multiple *containment* relationships, where each containment relationship must have a unique name. Note that the absence of a name is a legal name.

7 Including Metadata in CellML

Previous meetings had established the following rules for metadata:

- A model does not need to contain any metadata.
- Absence of metadata doesn't imply anything (like inheritance of metadata for instance.)
- Any element can have instances of all types of metadata. That is, it doesn't seem worth restricting the modeller from associating certain types of metadata with certain types of CellML object.
- To associate metadata with an element, it goes inside the element.
- Any element can only have one metadata-data-model's worth of metadata. That is, if the data model (still under construction) allows one instance of item A, that is all that can be associated with an element. This prevents multiple documentation sections within a single element.