# Meeting Minutes 26 March 2001

# Metadata: Use of Dublin Core Qualifiers

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

## 1 Introduction

As the April 6 freeze date looms ever larger on the horizon, the need to produce a good initial draft of the CellML metadata specification becomes increasingly urgent. Therefore, Melanie has turned her attention to sorting out the remaining issues identified in the March 15 meeting minutes[1]. The first item to be considered is the appropriate way to use Dublin Core qualifiers. This affects the alternative names and the creation and last modified dates. It is discussed in the first section of these meeting minutes.

The second section of these meeting minutes addresses species metadata. A solution to the issue of how to include groups of species (such as “mammals”) is proposed.

## 2 Dublin Core Qualifiers

Dublin Core qualifiers provide a standard way to extend the information that can be encoded in the Dublin Core element set. They are described in this document[2] (provided by the Dublin Core). There are two sorts of qualifiers: *type* and *scheme*. A *type* qualifier classifies the metadata being provided. For instance, date metadata can be of type “created”, “modified”, “valid”, “available”, or “issued”. A *scheme* qualifier identifies the method used to encode the metadata. For instance, date metadata may be encoded using either the DCMI Period method (to refer to a range of dates) or W3C-DTF method (to refer to a single date). The Dublin Core qualifier document defines allowed values for each of these qualifiers for each of the core Dublin Core elements.

### 2.1 Implementation of Dublin Core Qualifiers in RDF

Unfortunately, there is no ratified specification for implementing the Dublin Core qualifiers in RDF. There is, however, a draft proposal from the Dublin Core Data Model working group. It is available online here[3]. Note that this is not the official Dublin Core website. A thorough search of the archives of both the Dublin Core Data Model mailing list and the general Dublin Core mailing list indicates that this is the most current recommendation, but leaves its exact status unclear.

The recommendation in the draft proposal is slightly more verbose than the implementation given in the March 15 meeting minutes (an example of which is shown in Figure 1). However, it adheres to the recommendations of the RDF Model and Syntax Specification[4]. The addition of qualifiers into the data model makes the date metadata a resource in its own right, and makes the relationship between the date resource and its value non-binary. Specifically, the value of the date metadata is only valid in relationship to the information provided by the qualifier. The appropriate way to store non-binary relationships in RDF is discussed in section 7.3 of the RDF specification.

[1] http://www.cellml.org/private/progress_reports/20010315_meeting_minutes.html

[2] http://dublincore.org/documents/dcmes-qualifiers

[3] http://www.ukoln.ac.uk/metadata/resources/dc/datamodel/WD-dc-rdf/

[4] http://www.w3.org/TR/REC-rdf-syntax

```
<rdf:RDF
    xmlns:rdf="http://www.w3c.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.0"
    xmlns:dcq="http://purl.org/dc/qualifiers/1.0">

  <rdf:Description about="some_element_id">
    <dc:date
        dcq:dateScheme="W3CDTF"
        dcq:dateType="created">2000-10-05</dc:date>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 1: Extension of the Dublin Core Date element to store the model creation date, using the method of the March 15 meeting minutes.

Adhering to the RDF specification and the "Dublin Core in RDF" draft proposal requires making the following changes to the implementation:

- An additional `<rdf:Description>` element must be added to indicate that the date metadata is now a resource. This `<rdf:Description>` element should not have an about attribute, because it is indicating the creation of a new resource (see page 8 of the RDF specification.)
- The Dublin Core qualifiers must be elements, rather than attributes.
- The actual date must be enclosed in an `<rdf:value>` element.

Making these changes produces the metadata shown in Figure 2.

```
<rdf:RDF
    xmlns:rdf="http://www.w3c.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.0"
    xmlns:dcq="http://purl.org/dc/qualifiers/1.0">

  <rdf:Description about="some_element_id">
    <dc:date>
      <rdf:Description>
        <dcq:dateScheme>W3C-DTF</dcq:dateScheme>
        <dcq:dateType>created</dcq:dateType>
        <rdf:value>2000-10-05</rdf:value>
      </rdf:Description>
    </dc:date>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 2: Extension of the Dublin Core Date element to store the model creation date, using the method recommended in the "Dublin Core in RDF" draft proposal.

This method is more verbose, but it is also more correct. It should be adopted.

## 2.2 Additional Examples

The Dublin Core qualifiers are used for last modified date and alternative name metadata in addition to the creation date metadata shown above. Figure 3 shows the new implementation of the last modified date. Figure 4 shows the new implementation of alternative names.

```
<rdf:RDF
    xmlns:rdf="http://www.w3c.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.0"
    xmlns:dcq="http://purl.org/dc/qualifiers/1.0">

  <rdf:Description about="some_element_id">
    <dc:date>
      <rdf:Description>
        <dcq:dateScheme>W3C-DTF</dcq:dateScheme>
        <dcq:dateType>modified</dcq:dateType>
        <rdf:value>2000-10-05</rdf:value>
      </rdf:Description>
    </dc:date>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 3: Extension of the Dublin Core Date element to store the last modified date, using the method recommended in the “Dublin Core in RDF” draft proposal.

```
<rdf:RDF
    xmlns:rdf="http://www.w3c.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.0"
    xmlns:dcq="http://purl.org/dc/qualifiers/1.0">

  <rdf:Description about="some_element_id">
    <dc:title>
      <rdf:Description>
        <dcq:titleType>alternative</dcq:titleType>
        <rdf:value>Insulin signaling pathway model</rdf:value>
      </rdf:Description>
    </dc:title>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 4: An example of the use of alternative name metadata.

One interesting consequence of the new method for encoding alternative names is that it would allow us to reintroduce the “display name” concept. One of the instances of `<dc:title>` metadata could be left unqualified. This instance could be considered the preferred alternative name, i.e., the display name. All other instances would be qualified as “alternative”. An example of this is shown in Figure 5.

```
<rdf:RDF
    xmlns:rdf="http://www.w3c.org/1999/02/22-rdf-syntax-ns#"
    xmlns:dc="http://purl.org/dc/elements/1.0"
    xmlns:dcq="http://purl.org/dc/qualifiers/1.0">

  <rdf:Description about="some_element_id">
    <dc:title>EGF-EGFR complex</dc:title>

    <dc:title>
      <rdf:Description>
        <dcq:titleType>alternative</dcq:titleType>
        <rdf:value>epidermal growth factor-epidermal growth factor receptor complex</rdf:valu
      </rdf:Description>
    </dc:title>

  </rdf:Description>
</rdf:RDF>
```

FIGURE 5: An example of the use of alternative name metadata to provide a “display name”.

## 2.3 Recommendations

The following recommendations will be implemented in the draft metadata specification in the absence of complaints from other CellML team members:

- Dublin Core qualifiers will be used to indicate the type and encoding scheme of date metadata.
- If an alternative name should be considered a “display name”, it should be stored in an unqualified `<dc:title>` element.
- All other alternative names should be stored in a `<dc:title>` element that is qualified to be of type “alternative”, as shown in Figure 4.
- The new, more verbose syntax introduced in this document should be used for qualified Dublin Core metadata.

Note that the “Dublin Core in RDF” draft proposal is ambiguous about how to indicate that the value of a qualifier element is being taken from a controlled vocabulary (such as the one provided in the Dublin Core qualifiers document[5]). The examples in these meeting minutes do not do anything to indicate that the values come from a controlled vocabulary. However, this can reasonably be inferred from the use of the Dublin Core qualifier elements and the exact match of the values of these elements with entries in the DC qualifiers controlled vocabulary.

# 3 Allowing Groups of Species

Modellers must be able to indicate that a model or component is relevant for an entire group of species (such as “eukaryotes” or “mammals”) as well as for a single species (such as “Cava porcellus”, a.k.a. the guinea pig).

[5] http://dublincore.org/documents/dcmes-qualifiers

It is possible to support this requirement within the current proposed implementation. Groups of species can be identified by their valid scientific name, as found in resources such as the NCBI's Taxonomy Browser[6]. For instance, mammals can be referred to as "Mammalia". This is shown in Figure 6.

```
<rdf:RDF
    xmlns:rdf="http://www.w3c.org/1999/02/22-rdf-syntax-ns#"
    xmlns:cmeta="http://www.cellml.org/2000/cellml/RDF">

  <rdf:Description about="some_element_id">
    <cmeta:species>Eukaryota</cmeta:species>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 6: An example of the use of species metadata to indicate that a CellML element is valid for all eukaryotes.

The following points will be made in the metadata specification:

- If a modeller needs to refer to a discontinuous group of species (i.e., one that cannot be specified by a single scientific name) he/she can include multiple `<cmeta:species>` elements, as shown in Figure 7.
- The first word in a scientific name must be capitalized.
- Scientific names are deemed to be sufficiently standard that there is no need to refer to a controlled vocabulary for them. However, the the NCBI's Taxonomy Browser is a good resource for scientific names.

```
<rdf:RDF
    xmlns:rdf="http://www.w3c.org/1999/02/22-rdf-syntax-ns#"
    xmlns:cmeta="http://www.cellml.org/2000/cellml/RDF">

  <rdf:Description about="some_element_id">
    <cmeta:species>Cava porcellus</cmeta:species>
    <cmeta:species>Mus musculus</cmeta:species>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 7: An example of the use of species metadata to indicate that a CellML element is valid for guinea pigs and mice. Note that an RDF bag could have been used to group the species together. However, using an RDF bag does not add any information in this case (in contrast to the case of model creators), nor is it necessary to allow references to the entire collection (which would necessitate grouping the species together). Therefore, simply repeating the `<cmeta:species>` element is preferred for conciseness.

[6] http://www.ncbi.nlm.nih.gov/Taxonomy/taxonomyhome.html