Meeting Minutes 15 March 2001
Metadata Review

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

1 Introduction

The CellML specification defines the basic framework for including metadata in a CellML document, and the actual metadata structure for CellML documents will be defined in a companion specification. We would like to provide a solid draft of this companion specification by the 6 April 2001, CellML freeze. The work on metadata will be divided into two groups: metadata for references and everything else. We'll work on the "everything else" group first. This group currently uses three namespaces: `dc` (for Dublin Core elements), `dcq` (for Dublin Core qualifiers), and `cmeta` (for things we need to invent ourselves).

These meeting minutes begin the process of specifying a metadata structure for CellML documents. The following sections examine the non-reference metadata types. Each of these sections is split into three subsections: description of the information being stored, a summary of the current implementation, and a list of remaining issues to be resolved (or an indication that this type of metadata seems to be fully handled).

2 Alternative Names

2.1 Information Content

The primary name of an element is given by the value of its **`name`** attribute. However, this value must be a valid CellML identifier, which means that it can't have spaces or any non-alphanumeric characters except for underscores. More human-readable names can be provided using the alternative names metadata. This also allows a model author to store aliases for the object. A CellML element may have multiple alternative names.

2.2 Current Implementation

Alternative names are stored using the Dublin Core title element, as shown in Figure 1.

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

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

FIGURE 1: An example of the use of alternative name metadata.

2.3 Remaining Issues

The Dublin Core now has an “alternative” qualifier for the **<dc:title>** element. We should use this. I think this would look like Figure 2, but the Dublin Core has changed its qualifier info since I last looked at it, so this needs some more investigation.

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

FIGURE 2: An example of the use of alternative name metadata.

3 Model Builder

3.1 Information Content

The model builder metadata stores information about the person or persons who coded the model into CellML. (Note that the person or persons who originally authored the model can be stored in the reference metadata.) A given element can have multiple model builders, which may need to be considered as individuals or as members of a group. If they are members of a group, the group may or may not need to be ordered.

3.2 Current Implementation

Model builder metadata can be stored using the Dublin Core creator element, as shown in Figure 3. This example shows extension `cmeta` elements that store information about the model builders. (Note that the use of the **<rdf:Description>** element without an **about** attribute indicates the creation of a new resource, which in this case represents the model builder. All of the elements contained in the second **<rdf:Description>** element contain metadata about this new resource.)

If a CellML element has more than one creator, it is up to the model builder to decide whether to simply repeat the **<dc:creator>** metadata element, or to group the creators into a bag (unordered collection of objects) or sequence (ordered collection of objects). If a modeller wishes to indicate that multiple people worked together to create a model, but does not want to order the authors, he/she should group the creators into a bag. If the modeller wishes to make a statement about the order of the authors, he/she should use a sequence. If the modeller does not wish to indicate that the creators worked together as a group (for instance, if three people worked on a model at three different points in time), he/she should simply repeat the **<dc:creator>** element.

```
<rdf:RDF
    xmlns:rdf="http://www.w3c.org/1999/02/22-rdf-syntax-ns#"
    xmlns:cmeta="http://www.cellml.org/2000/cellml/RDF"
    xmlns:dc="http://purl.org/dc/elements/1.0">

  <rdf:Description about="some_element_id">
    <dc:creator>
      <rdf:Description>
        <cmeta:surname>Doe</cmeta:surname>
        <cmeta:first_name>Jane</cmeta:first_name>
        <cmeta:mid_initials>A</cmeta:mid_initials>
        <cmeta:email>jdoe@physiome.com</cmeta:email>
        <cmeta:affiliation>
          Physiome Sciences, Inc.
        </cmeta:affiliation>
        <cmeta:mail_address
            city="Princeton"
            country="USA"
            postcode="08540"
            street="307 College Road East"
            territory="NJ"
            type="work" />
        <cmeta:phone_number
            number="609-987-1199"
            type="work" />
        <cmeta:phone_number
            number="609-987-9393"
            type="fax" />
      </rdf:Description>
    </dc:creator>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 3: An example of model builder information.

3.3 Remaining Issues

- We need to determine how to handle metadata about people (the contents of the second **<rdf:Description** element in the example). If possible, we should use an existing schema rather than using elements from the `cmeta` schema.
- We need to determine whether or not the second **<rdf:Description>** element is necessary. Can we instead just have the **<dc:creator>** element contain the elements that provide information about the creator.

4 Species

4.1 Information Content

Species metadata refers to the biological species (such as human, dog, pig, etc.) for which a model is relevant. A given CellML element may be relevant for multiple species. It may also be relevant for an entire class of species (such as all mammals).

4.2 Current Implementation

Species metadata is stored using a CellML-specific species metadata element, as shown in Figure 4.

```
<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>Cavia porcellus</cmeta:species>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 4: An example of the use of species metadata

The content of the **<cmeta:species>** element must be either a scientific name for a species or a valid name for a group of species.

4.3 Remaining Issues

We need to determine if we should limit the allowed names for groups of species. We could create a controlled vocabulary, or we may be able to use standard taxonomy names for groups of species.

5 Sex

5.1 Information Content

Sex metadata refers to the sex for which a CellML element is relevant.

5.2 Current Implementation

Sex metadata is stored using a CellML-specific sex metadata element, as shown in Figure 5.

```
<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:sex>male</cmeta:sex>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 5: An example of the use of sex metadata

A given CellML element may contain multiple **<cmeta:sex>** elements. The content of the **<cmeta:sex>** element must be chosen from the following controlled vocabulary:

- male
- female
- hermaphrodite
- undefined (the element is explicitly specified not to have a defined relevance to any particular sex)

5.3 Remaining Issues

There are no remaining issues about sex metadata.

6 Creation Date

6.1 Information Content

The creation date is the date upon which the model or model part was coded into CellML. A given CellML element can have only one creation date.

6.2 Current Implementation

Creation date metadata is stored using an extension of the Dublin Core Date element, as shown in Figure 6.

6.3 Remaining Issues

There are no remaining issues with creation date metadata

```
<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 6: Extension of the Dublin Core Date element to store the model creation date.

7 Last Modified Date

7.1 Information Content

The last modified date is the date upon which the content of a CellML element was last changed. A given CellML element can have only one last modified date.

7.2 Current Implementation

Last modified date metadata is stored using an extension of the Dublin Core Date element, as shown in Figure 7.

```
<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="modified">2000-10-05</dc:date>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 7: Extension of the Dublin Core Date element to store the last modifi ed date.

7.3 Remaining Issues

There are no remaining issues with last modified date metadata

8 Annotations

8.1 Information Content

Currently, there are three types of annotations:

- **model builder comment**: free-form comment of the person who coded the model into CellML
- **limitation**: brief text description of the limitations/scope of the content of the CellML element
- **modification**: text describing a change made to the content of the CellML element

Each annotation also has creator and creation date metadata that refers to it.

8.2 Current Implementation

Annotations are stored using a CellML-specific annotation element. The metadata about the creator of the annotation should be stored in the same way as the metadata about a CellML element creator (model builder) is stored. The metadata about the creation date of the annotation is stored with the same extended form of the Dublin Core date element used for the CellML element creation and last modified dates. Examples of the three current types of annotation metadata are shown in Figure 8. Note that the implementation of this metadata has changed since the last specification. The annotation text is now direct content of the `<cmeta:annotation>` element, while the information about the creator and creation date are stored in a new `<rdf:Description>` element with an `about` attribute value equal to the value of the `id` attribute on the `<cmeta:annotation>` element (indicating that this second `<rdf:Description>` element is providing metadata about the `<cmeta:annotation>` element).

8.3 Remaining Issues

The following three issues need to be resolved:

- The new method for storing annotation metadata should be discussed, and all team members should be sure that they find it acceptable.
- Recent discussions with Chris Penland indicate that it might be useful to add an annotation type to store information about the validation level of a model. Physiome modellers are working on a standard validation protocol. While the CellML metadata should not force others to use the same validation protocol, it should allow Physiome to store information about which steps in its validation protocol a model has passed.
- The metadata about the annotation creator needs to be stored in the same way as the metadata about a CellML element creator is stored (see Section 3).

9 Biological Entity

9.1 Information Content

This area of the metadata will almost certainly be expanded in future versions of CellML. For now, it is simply a name or database unique identifier for a biological entity, such as an ion channel, signaling pathway, or specific cell type, that is represented by the model or model component. A given CellML element can represent multiple biological entities.

```
<rdf:RDF
    xmlns:rdf="http://www.w3c.org/1999/02/22-rdf-syntax-ns#"
    xmlns:cmeta="http://www.cellml.org/2000/cellml/RDF"
    xmlns:dc="http://purl.org/dc/elements/1.0"
    xmlns:dcq="http://purl.org/dc/qualifiers/1.0">

  <rdf:Description about="some_element_id">
    <cmeta:annotation annotation_type="limitation" id="annot1">
      This model is not valid for drug-inhibition studies.
      <rdf:Description about="annot1">
        <dc:creator>
          <rdf:Description>
            <cmeta:surname>Blow</cmeta:surname>
            <cmeta:first_name>Joe</cmeta:first_name>
          </rdf:Description>
        </dc:creator>
        <dc:date
            dcq:dateScheme="W3CDTF"
            dcq:dateType="created">2000-10-05</dc:date>
      </rdf:Description>
    </cmeta:annotation>

    <cmeta:annotation annotation_type="comment" id="annot2">
      The voltage-gating implementation could be improved.
      <rdf:Description about="annot2">
        <dc:creator>
          <rdf:Description>
            <cmeta:surname>Blow</cmeta:surname>
            <cmeta:first_name>Joe</cmeta:first_name>
          </rdf:Description>
        </dc:creator>
        <dc:date
            dcq:dateScheme="W3CDTF"
            dcq:dateType="created">2000-10-05</dc:date>
      </rdf:Description>
    </cmeta:annotation>

    <cmeta:annotation annotation_type="modification" id="annot3">
      changed conductance
      <rdf:Description about="annot3">
        <dc:creator>
          <rdf:Description>
            <cmeta:surname>Blow</cmeta:surname>
            <cmeta:first_name>Joe</cmeta:first_name>
          </rdf:Description>
        </dc:creator>
        <dc:date
            dcq:dateScheme="W3CDTF"
            dcq:dateType="created">2000-10-08</dc:date>
      </rdf:Description>
    </cmeta:annotation>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 8: An example of the three types of annotation metadata.

9.2 Current Implementation

Biological entity metadata is stored using a CellML-specific **<bio_entity>** element, as shown in Figure 9. Note that the **entity_scheme** attribute has been added since the last specification. It can have a value of "name" or "db_ui".

```
<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:bio_entity entity_scheme="name">HERG</cmeta:bio_entity>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 9: An example of biological entity metadata.

9.3 Remaining Issues

We need to decide how to refer to biological entities by a database unique identifier. This requires referring to both the identity of the database (GenBank, SWISS-PROT, Physiome's DB, etc.) and the value of the unique identifier for the entity within the database.

10 Copyright

10.1 Information Content

The copyright metadata refers to the copyright that protects the model, model component, or other CellML element. It is implemented using the Dublin Core Rights element, and therefore, a given CellML element can technically have multiple copyrights. However, the recommended best practice is to include only one copyright for any given element.

10.2 Current Implementation

Copyright metadata is stored using the copyright element from the Dublin Core, as shown in Figure 10.

10.3 Remaining Issues

There are no remaining issues about copyright metadata.

11 Publisher

11.1 Information Content

The publisher is the person or organization responsible for providing the model, model component, or other CellML element. A given CellML element can have multiple publishers.

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

  <rdf:Description about="some_element_id">
    <dc:rights>Physiome Sciences, 2000</dc:rights>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 10: An example of the inclusion of copyright metadata

11.2 Current Implementation

Publisher metadata is stored using the Dublin Core Publisher element, as shown in Figure 11.

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

  <rdf:Description about="">
    <dc:publisher>University of Auckland, Bioengineering Research Group
    </dc:publisher>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 11: An example of the inclusion of publisher metadata

As defined for model builder metadata, publisher metadata can be repeated in three ways. If the **<dc:publisher>** element is simply repeated, it indicates that the resource has been provided independently by multiple publishers. If the **<dc:publisher>** element contains a bag, it indicates that the resource has been provided jointly by a group of publishers, and that all publishers in that group are equal. If the **<dc:publisher>** element contains a sequence, it indicates that the resource has been provided jointly by a group of publishers, and that the publishers in that group are ordered.

11.3 Remaining Issues

There are no remaining issues about publisher metadata.

12 Mathematical Problem Type

12.1 Information Content

The mathematical problem type is a classification of the type of problem encoded in the math associated with the model or model component. It should be specified using some sort of controlled vocabulary, such

as the NIST's GAMS classification tree[1].

12.2 Current Implementation

Mathematical problem type metadata is stored using the CellML-specific math_problem_type element, as shown in Figure 12.

```
<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:math_problem_type
        problem_scheme="GAMS">I1a</cmeta:math_problem_type>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 12: An example of the mathematical problem type metadata.

12.3 Remaining Issues

We need to add a way to provide a URL for the problem classification scheme.

13 Additional Issues to Discuss

The metadata specification will define the metadata elements that processing software must be able to handle if they claim to be "CellML metadata compliant". Software and model authors will be free to use other metadata if appropriate, but this metadata will be less interoperable than metadata defined using the elements and rules in the CellML metadata specification. The decision about whether or not to include a specific metadata element in the specification must balance the desire to provide a complete set of interoperable metadata, and the need to keep the set a manageable size for processing software to implement. The following additional Dublin Core elements should be considered for inclusion in the specification:

- **The `<dc:description>` element**: this element is used to store a short text description of the resource.
- **The `<dc:contributor>` element**: this element is used to indicate a person contributed to a resource, but did not actually create it. The example given is usually an editor.
- **The `<dc:relation>` element**: this element indicates that the current resource is related in some way to another resource. We could use this to indicate that a model or model part has been derived from another model. (This might be a good item to postpone until next version.)

[1] http://gams.nist.gov/