

Meeting Minutes 3 April 2001

Metadata Issues

Author:

Melanie Nelson (Physiome Sciences Inc.)

Contributor:

Warren Hedley (Bioengineering Research Group, University of Auckland)

1 Introduction

Work on developing a recommendation for the representation of non-reference, non-person metadata in CellML highlighted the following three general issues:

- Should we use the `rdf:parseType` attribute with a value of "Resource" to eliminate some of the `<rdf:Description>` elements in our RDF?
- How should we use RDF containers to represent multivalued metadata?
- How should we implement controlled vocabularies of terms?

These meeting minutes address each of the above issues, and then apply the conclusions to the non-reference and non-person metadata types. Section 5 provides examples of the recommended definition of all non-reference and non-person metadata types. These recommendations will be included in the draft metadata specification, contingent upon acceptance by the rest of the CellML development team.

2 The `rdf:parseType` Attribute

The use of the `parseType` attribute is buried on page 31 of the [RDF syntax specification](#)¹. This states that the element content of an RDF element with an `rdf:parseType` attribute value of "Resource" "must be treated as if it were the content of a `<rdf:Description>` element."

The use of this attribute can decrease the verbosity of RDF. For instance, Figure 1 and Figure 2 both represent the same piece of metadata: a limitation annotation, created by Betty Smith on 28 March 2001.

There only disadvantage of using the `rdf:parseType` attribute in this manner is that it does not exactly follow the recommendations of the Dublin Core Data Model working group for representing qualified Dublin Core in RDF (see [the working group's draft proposal](#)²). They recommend using the `<rdf:Description>` element. However, since the meaning of the two methods is precisely equivalent, this does not seem to be a major concern. Furthermore, the document produced by the working group remains a draft proposal, and is not published by the official Dublin Core Metadata Initiative site. Therefore, its status is uncertain.

Another possible concern is that the need to support both the use of `<rdf:Description>` elements and the `rdf:parseType` attribute is an added burden on CellML processing software. This is technically true, but only for software that chooses to support CellML metadata by implementing only the recommended metadata definitions in the CellML metadata specification. Any software that chooses to support CellML metadata by becoming fully RDF compliant will have to support both the `<rdf:Description>` element and the `rdf:parseType` attribute regardless of what we recommend.

If the metadata specification does not recommend use of the `rdf:parseType` attribute, metadata-compliant CellML processing software would be required to recognize both of the following two uses of the `<rdf:Description>` element:

¹<http://www.w3.org/TR/REC-rdf-syntax/>

²<http://www.ukoln.ac.uk/metadata/resources/dc/datamodel/WD-dc-rdf/>

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cmeta="http://www.cellml.org/2001/03/metadata#"
  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>
      <rdf:Description>
        <cmeta:annotation_type>limitation</cmeta:annotation_type>
        <rdf:value>
          This component is only valid for temperatures above 20 degrees C
        </rdf:value>
        <dc:creator>Betty Smith</dc:creator>
        <dc:date>
          <rdf:Description>
            <dcq:dateScheme>W3C-DTF</dcq:dateScheme>
            <dcq:dateType>created</dcq:dateType>
            <rdf:value>2001-03-28</rdf:value>
          </rdf:Description>
        </dc:date>
      </rdf:Description>
    </cmeta:annotation>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 1: Original definition of annotation metadata. Note that the contents of the `<dc:creator>` element are provided for example only. Recommendations for how CellML metadata should handle information about people will be provided in a separate document.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cmeta="http://www.cellml.org/2001/03/metadata#"
  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 rdf:parseType="Resource">
      <cmeta:annotation_type>limitation</cmeta:annotation_type>
      <rdf:value>
        This component is only valid for temperatures above 20 degrees C
      </rdf:value>
      <dc:creator>Betty Smith</dc:creator>
      <dc:date rdf:parseType="Resource">
        <dcq:dateScheme>W3C-DTF</dcq:dateScheme>
        <dcq:dateType>created</dcq:dateType>
        <rdf:value>2001-03-28</rdf:value>
      </dc:date>
    </cmeta:annotation>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 2: Recommended definition of annotation metadata using the `rdf:parseType` attribute. Note that the contents of the `<dc:creator>` element are provided for example only. Recommendations for how CellML metadata should handle information about people will be provided in a separate document.

- If an `<rdf:Description>` element defines an `about` attribute, its contents store metadata about a resource identified by the value of the `about` attribute, or about the entire document (if the `about` attribute is given an empty value).
- If an `<rdf:Description>` element does not define an `about` attribute, its contents create a new anonymous resource, and store metadata about it.

If the metadata specification recommends the use of the `rdf:parseType` attribute, software will have to recognize both of the following two methods for storing metadata about a resource:

- If an `<rdf:Description>` element defines an `about` attribute, its contents store metadata about a resource identified by the value of the `about` attribute, or about the entire document (if the `about` attribute is given an empty value).
- If any metadata element defines an `rdf:parseType` attribute value of `Resource`, its contents create a new anonymous resource, and store metadata about it.

The second option does create a slight additional burden on processing software, because in addition to recognizing that the `<rdf:Description>` element refers to a resource, it must also recognize that any element with an `rdf:parseType` attribute with a value of `"Resource"` creates an anonymous resource. However, this is not a large burden, and it is balanced by the savings in filesize produced by using the more compact notation. Furthermore, there are several publicly available Java classes and other general tools to parse RDF. It is likely that CellML processing software will use these tools to become metadata compliant by becoming fully RDF capable.

Recommendation: use the `rdf:parseType` attribute to create a new anonymous resource.

3 Use of RDF Containers

RDF provides three types of containers for multivalued metadata: bags, sequences, and alternatives. The RDF specification defines the meaning of each of these. A bag is used if a metadata element has unordered, multivalued content. An example is the students in a class. A sequence is used if a metadata element has ordered, multivalued content. An example is the authors on a scientific paper. An alternative is used if a metadata element has single-valued content that is chosen from a list of alternatives. An example is the title of a document provided in multiple languages.

Warren is concerned that not every type of RDF container may be relevant for each metadata element in the CellML metadata specification, and wonders if we need to restrict the use of containers. This should not be necessary. The RDF specification already defines the meaning of each type of container. The recommendations in the CellML metadata specification will demonstrate how to use containers to represent the types of metadata that CellML metadata compliant software must recognize. If software chooses to become CellML metadata compliant by implementing only the types of metadata defined in the CellML metadata specification, it need only recognize the containers shown in the recommended definitions in the specification. If software chooses to become CellML metadata compliant by becoming fully RDF capable, the RDF specification will define the meaning of the different containers for us.

4 Use of Controlled Vocabularies of Terms

There are several possible methods for dealing with controlled vocabularies of terms in RDF metadata:

- List the terms and their meanings in the CellML metadata specification, and leave it to processing software to restrict the allowed values of the relevant metadata elements.

- Use the `<rdfs:range>` property in the RDF schema to limit the allowed values for an element to a specified set. (The terms and their meanings will also be listed in the CellML metadata specification).
- Use the `rdf:resource` attribute on the metadata element to point to a namespace URL.

The first two options produce the same RDF in the CellML document, an example of which is shown in Figure 3. This is because the control of the vocabulary is handled in a separate document (the CellML metadata specification for option 1 and the CellML RDF Schema for option 2). The second option is preferable to the first because it produces machine-understandable limits on the allowed values of the element. Since these allowed values would also be defined in the specification, the second option does not produce any additional demands on CellML processing software: software could still ignore the RDF Schema and implement only the recommendations in the CellML metadata specification.

The third option produces the RDF shown in Figure 4. The metadata content in the two figures is identical.

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cmeta="http://www.cellml.org/2001/03/metadata#">

  <rdf:Description about="some_element_id">
    <cmeta:math_problem rdf:parseType="Resource">
      <cmeta:math_problem_scheme>GAMS</cmeta:math_problem_scheme>
      <rdf:value>I1a</rdf:value>
    </cmeta:math_problem>
  </rdf:Description>
</rdf:RDF>

```

FIGURE 3: An example of RDF metadata produced by controlling vocabularies of terms in the specification and the RDF schema.

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cmeta="http://www.cellml.org/2001/03/metadata#">

  <rdf:Description about="some_element_id">
    <cmeta:math_problem rdf:parseType="Resource">
      <cmeta:math_problem_scheme
        rdf:resource="http://www.cellml.org/metadata/terms#GAMS/" />
      <rdf:value>I1a</rdf:value>
    </cmeta:math_problem>
  </rdf:Description>
</rdf:RDF>

```

FIGURE 4: An example of RDF metadata produced by controlling vocabularies of terms using the `rdf:resource` attribute.

One important difference between the options presented in the two figures is the method by which modellers can use their own terms as values for the RDF element. If the terms in the vocabulary are controlled by the RDF schema and the CellML metadata specification (Figure 3), modellers would need to place the metadata element in their own namespace in order to use a term not in the CellML controlled vocabulary, as shown in Figure 5. If the terms in the vocabulary are controlled using the `rdf:resource` attribute, modellers would need to supply their own URL as a value for this attribute, as shown in Figure 6.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cmeta="http://www.cellml.org/2001/03/metadata#"
  xmlns:app="http://www.bozo.com/my_app/">

  <rdf:Description about="some_element_id">
    <cmeta:math_problem rdf:parseType="Resource">
      <app:problem_scheme>my classification scheme</app:problem_scheme>
      <rdf:value>system of ODEs</rdf:value>
    </cmeta:math_problem>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 5: An example of using a term not in the CellML controlled vocabulary if the controlling vocabularies of terms are defined in the specification and the RDF schema.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cmeta="http://www.cellml.org/2001/03/metadata#">

  <rdf:Description about="some_element_id">
    <cmeta:math_problem rdf:parseType="Resource">
      <cmeta:math_problem_scheme
        rdf:resource="http://www.bozo.com/my_app/terms#sillymath/" />
      <rdf:value>ODE</rdf:value>
    </cmeta:math_problem>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 6: An example of using a term not in the CellML controlled vocabulary if the controlling vocabularies of terms are defined using the `rdf:resource` attribute.

Recommendation: Control vocabularies in the RDF schema, and include an explanation of all terms in the CellML metadata specification (this results in RDF as shown in Figure 3 and Figure 5). This method is marginally preferable to the method shown in Figure 4 and Figure 6, because that method mixes RDF abbreviated syntax with the full syntax. Mixing the two syntaxes is legal in RDF, but using such a mixed syntax would increase the demands on CellML processing software.

5 New Metadata Implementation

The following subsections demonstrate the application of the recommendations in the previous three sections to all non-reference and non-person metadata. Note that the names of the mathematical problem type metadata elements have changed slightly, to remove a potential source of confusion. This is discussed in Section 5.11.

The discussion presented here assumes that the reader is familiar with the use of Dublin Core elements and Dublin Core qualifiers. These are discussed in the 26 March 2001 meeting minutes.

5.1 Alternative Names

Alternative name metadata provides human-readable names for CellML elements. One of these names can be considered to be the preferred name, equivalent to the old “displayname” concept in CellML99. This preferred name could be used by software whenever it needs to display a human-readable name. The use of this metadata allows us to limit the values of `name` attributes on CellML elements to enable efficient code generation, without worrying about whether or not the name will be sufficiently meaningful to human readers.

Alternative name metadata is defined with the Dublin Core title element, `<dc:title>`.

One element may have multiple alternative names. Only one should be considered the preferred human-readable name. The preferred name should be stored in an unqualified `<dc:title>` element. Additional names should be stored in `<dc:title>` elements that are qualified by setting the title type (`<dcq:title>`) to “alternative”.

Figure 7 shows the definition of alternative name metadata. The element referenced by `#cellml_element_id` is given two human-readable names. The preferred one is “EGF-EGFR complex”.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.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="#cellml_element_id">
    <dc:title>EGF-EGFR complex</dc:title>
    <dc:title rdf:parseType="Resource">
      <dcq:titleType>alternative</dcq:titleType>
      <rdf:value>
        epidermal growth factor-epidermal growth factor receptor complex
      </rdf:value>
    </dc:title>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 7: Recommended definition of alternative name metadata.

5.2 Model Builder

Model builder metadata stores information about the person or persons who coded the model into CellML. 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.

Model builder metadata is defined using the Dublin Core creator element, `<dc:creator>`. Repeating this element for a given CellML element indicates that the people listed worked independently on the model. This definition is shown in Figure 8. Listing multiple people in the `<dc:creator>` element using an `<rdf:Bag>` container indicates that the group of people worked together on the model, and that they are all considered equal contributors. This definition is shown in Figure 9. Listing multiple people in the `<dc:creator>` element using an `<rdf:Seq>` container indicates that the group of people worked together on the model, and that their contributions are ordered (the first member of the list is first author, the second member is second author, and so on). This definition is shown in Figure 10. The CellML metadata specification will not include the use of an `<rdf:Alt>` container with this type of metadata. Metadata authors are free to use this container (as long as they produce valid RDF). However, CellML metadata compliant software is not required to be able to consistently interpret the meaning of an `<rdf:Alt>` container in this context. Note that in all of the examples shown here, the basic vCard “name” construct is used to store the name of the model builder. This and other vCard constructs will be discussed in a later document.

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.0/"
  xmlns:vCard="http://www.w3.org/2001/vcard-rdf/3.0#">

  <rdf:Description about="#cellml_element_id">
    <dc:creator rdf:parseType="Resource">
      <vCard:N rdf:parseType="Resource">
        <vCard:Family>Flintstone</vCard:Family>
        <vCard:Given>Fred</vCard:Given>
      </vCard:N>
    </dc:creator>
    <dc:creator rdf:parseType="Resource">
      <vCard:N rdf:parseType="Resource">
        <vCard:Family>Brown</vCard:Family>
        <vCard:Given>Charlie</vCard:Given>
      </vCard:N>
    </dc:creator>
    <dc:creator rdf:parseType="Resource">
      <vCard:N rdf:parseType="Resource">
        <vCard:Family>Doo</vCard:Family>
        <vCard:Given>Scooby</vCard:Given>
      </vCard:N>
    </dc:creator>
  </rdf:Description>
</rdf:RDF>

```

FIGURE 8: Recommended definition of model builder metadata in which multiple people worked independently on the model.

5.3 Species

Species metadata refers to the biological species (such as human, dog, pig, etc.) for which an element 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.

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.0/"
  xmlns:vCard="http://www.w3.org/2001/vcard-rdf/3.0#">

  <rdf:Description about="#cellml_element_id">
    <dc:creator>
      <rdf:Bag>
        <rdf:li rdf:parseType="Resource">
          <vCard:N rdf:parseType="Resource">
            <vCard:Family>Flintstone</vCard:Family>
            <vCard:Given>Fred</vCard:Given>
          </vCard:N>
        </rdf:li>
        <rdf:li rdf:parseType="Resource">
          <vCard:N rdf:parseType="Resource">
            <vCard:Family>Brown</vCard:Family>
            <vCard:Given>Charlie</vCard:Given>
          </vCard:N>
        </rdf:li>
        <rdf:li rdf:parseType="Resource">
          <vCard:N rdf:parseType="Resource">
            <vCard:Family>Doo</vCard:Family>
            <vCard:Given>Scooby</vCard:Given>
          </vCard:N>
        </rdf:li>
      </rdf:Bag>
    </dc:creator>
  </rdf:Description>
</rdf:RDF>

```

FIGURE 9: Recommended definition of model builder metadata in which multiple people worked together on the model, and all are considered equal contributors.

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.0/"
  xmlns:vCard="http://www.w3.org/2001/vcard-rdf/3.0#">

  <rdf:Description about="#cellml_element_id">
    <dc:creator>
      <rdf:Seq>
        <rdf:li rdf:parseType="Resource">
          <vCard:N rdf:parseType="Resource">
            <vCard:Family>Flintstone</vCard:Family>
            <vCard:Given>Fred</vCard:Given>
          </vCard:N>
        </rdf:li>
        <rdf:li rdf:parseType="Resource">
          <vCard:N rdf:parseType="Resource">
            <vCard:Family>Brown</vCard:Family>
            <vCard:Given>Charlie</vCard:Given>
          </vCard:N>
        </rdf:li>
        <rdf:li rdf:parseType="Resource">
          <vCard:N rdf:parseType="Resource">
            <vCard:Family>Doo</vCard:Family>
            <vCard:Given>Scooby</vCard:Given>
          </vCard:N>
        </rdf:li>
      </rdf:Seq>
    </dc:creator>
  </rdf:Description>
</rdf:RDF>

```

FIGURE 10: Recommended definition of model builder metadata in which multiple people worked together on the model, but all are not considered equal contributors. In this example, Fred Flintstone is the first author, Charlie Brown is the second author, and Scooby Doo is the third author.

Species metadata is defined with a CellML-specific metadata element, `<cmeta:species>`, as shown in Figure 11. The content of this metadata must be a valid scientific name for a species or group of species. Notwithstanding recent arguments among taxonomists about the impact of genomic data on species classifications, scientific names are considered to be sufficiently standard to obviate the need to use a formal controlled vocabulary. Constructing such a vocabulary would be a daunting task! However, the CellML metadata specification will refer to the [NCBI's Taxonomy Browser](http://www.ncbi.nlm.nih.gov/Taxonomy/TaxonomyBrowser)³ as a good resource for scientific names. 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. This was chosen over the use of RDF containers because it is simpler, and there was no need to differentiate between different possible meanings of multiple values for the species metadata. Multiple values for the species metadata will always mean that the CellML element is relevant for any one of the species listed. Relevance to all of the species as a group would imply some sort of population dynamics model, which is outside of the scope of CellML.

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cmeta="http://www.cellml.org/2001/03/metadata#">

  <rdf:Description about="#cellml_element_id">
    <cmeta:species>Mammalia</cmeta:species>
    <cmeta:species>Xenopus laevis</cmeta:species>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 11: Recommended definition of species metadata. The element referenced by "#cellml_element_id" is relevant for all mammals and the African clawed frog, *Xenopus laevis*.

5.4 Sex

Sex metadata refers to the sex for which a CellML element is relevant. A given element may be relevant for more than one sex.

Sex metadata is defined with the CellML-specific element, `<cmeta:sex>`, as shown in Figure 12. The valid content of this 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).

5.5 Creation Date

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.

³<http://www.ncbi.nlm.nih.gov/Taxonomy/taxonomyhome.html>

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cmeta="http://www.cellml.org/2001/03/metadata#">

  <rdf:Description about="#cellml_element_id">
    <cmeta:sex>male</cmeta:sex>
  </rdf:Description>
</rdf:RDF>

```

FIGURE 12: An example of the use of sex metadata

Creation date metadata is defined using the fully-qualified form of the Dublin Core date element, `<dc:date>`. The fact that the date is a creation date is indicated by setting the date type qualifier (`<dcq:dateType>`) to “created”. The encoding scheme for the date is named in the date scheme qualifier (`<dcq:dateScheme>`). The allowed values of the encoding scheme qualifier are a controlled vocabulary from the Dublin Core (see [the Dublin Core Qualifiers document](#)⁴.) The definition of creation date metadata is demonstrated in Figure 13.

```

<rdf:RDF
  xmlns:rdf="http://www.w3.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="#cellml_element_id">
    <dc:date rdf:parseType="Resource">
      <dcq:dateScheme>W3C-DTF</dcq:dateScheme>
      <dcq:dateType>created</dcq:dateType>
      <rdf:value>2000-10-05</rdf:value>
    </dc:date>
  </rdf:Description>
</rdf:RDF>

```

FIGURE 13: Recommended definition of the creation date metadata.

5.6 Last Modified Date

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.

The last modified date metadata is defined with the fully-qualified Dublin Core date element. Its definition is exactly the same as that of the creation date metadata, except that the value of the date type qualifier (`<dcq:dateType>`) is “modified”. The definition of creation date metadata is demonstrated in Figure 14.

⁴<http://dublincore.org/documents/dcmes-qualifiers/>

```

<rdf:RDF
  xmlns:rdf="http://www.w3.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="#cellml_element_id">
    <dc:date rdf:parseType="Resource">
      <dcq:dateScheme>W3C-DTF</dcq:dateScheme>
      <dcq:dateType>modified</dcq:dateType>
      <rdf:value>2000-10-05</rdf:value>
    </dc:date>
  </rdf:Description>
</rdf:RDF>

```

FIGURE 14: Recommended definition of the last modified date metadata.

5.7 Annotations

There are four types of annotations that will be recognized by the CellML metadata specification. Model authors are free to create additional types. However, CellML metadata compliant software will not be required to recognize any annotation types except for the following four:

- **comment:** free-form comment of the person who coded the model into CellML.
- **limitation:** brief description of the limitations/scope of the content of the CellML element.
- **modification:** description of a change made to the content of the CellML element.
- **validation:** description of the level of validation of the content of the CellML element. This may be a code. Note that validation codes are unlikely to be interoperable.

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

Annotation metadata is defined using a CellML-specific element, **<cmeta:annotation>**. This element is qualified to include a type (**<cmeta:annotation_type>**) that indicates which type of annotation is included. The content of the **<cmeta:annotation_type>** element is a vocabulary controlled by the CellML RDF specification, with four valid values: comment, limitation, modification, validation. If a model author wishes to use a different value, he/she must place the **<annotation_type>** element in an application-specific namespace.

The author metadata associated with an annotation is defined exactly as the model builder metadata (Section 5.2), and creation date metadata associated with an annotation is defined exactly as the general creation date metadata (Section 5.5).

Figure 15 demonstrates the definition of comment and limitation annotations. Figure 16 demonstrates the definition of modification and validation annotations.

5.8 Biological Entity

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, either as a complete group or as a list of alternatives.

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cmeta="http://www.cellml.org/2001/03/metadata#"
  xmlns:dc="http://purl.org/dc/elements/1.0/"
  xmlns:dcq="http://purl.org/dc/qualifiers/1.0/"
  xmlns:vCard="http://www.w3.org/2001/vcard-rdf/3.0#">

  <rdf:Description about="#cellml_element_id">
    <cmeta:annotation rdf:parseType="Resource">
      <cmeta:annotation_type>comment</cmeta:annotation_type>
      <rdf:value>This model does not include the data of Jones, et al.
      about the corresponding pathway in canine.</rdf:value>
      <dc:creator rdf:parseType="Resource">
        <vCard:N rdf:parseType="Resource">
          <vCard:Family>PowerPuff</vCard:Family>
          <vCard:Given>Bubbles</vCard:Given>
        </vCard:N>
      </dc:creator>
      <dc:date rdf:parseType="Resource">
        <dcq:dateScheme>W3C-DTF</dcq:dateScheme>
        <dcq:dateType>created</dcq:dateType>
        <rdf:value>2001-04-01</rdf:value>
      </dc:date>
    </cmeta:annotation>
    <cmeta:annotation rdf:parseType="Resource">
      <cmeta:annotation_type>limitation</cmeta:annotation_type>
      <rdf:value>
      This component is only valid for temperatures above 20 degrees C
      </rdf:value>
      <dc:creator rdf:parseType="Resource">
        <vCard:N rdf:parseType="Resource">
          <vCard:Family>Doo</vCard:Family>
          <vCard:Given>Scooby</vCard:Given>
        </vCard:N>
      </dc:creator>
      <dc:date rdf:parseType="Resource">
        <dcq:dateScheme>W3C-DTF</dcq:dateScheme>
        <dcq:dateType>created</dcq:dateType>
        <rdf:value>2001-03-28</rdf:value>
      </dc:date>
    </cmeta:annotation>
  </rdf:Description>
</rdf:RDF>

```

FIGURE 15: Recommended definition of comment and limitation annotation metadata.

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cmeta="http://www.cellml.org/2001/03/metadata#"
  xmlns:dc="http://purl.org/dc/elements/1.0/"
  xmlns:dcq="http://purl.org/dc/qualifiers/1.0/"
  xmlns:vCard="http://www.w3.org/2001/vcard-rdf/3.0#">

  <rdf:Description about="#cellml_element_id">
    <cmeta:annotation rdf:parseType="Resource">
      <cmeta:annotation_type>modification</cmeta:annotation_type>
      <rdf:value>changed the equation for the sodium current</rdf:value>
      <dc:creator rdf:parseType="Resource">
        <vCard:N rdf:parseType="Resource">
          <vCard:Family>PowerPuff</vCard:Family>
          <vCard:Given>Bubbles</vCard:Given>
        </vCard:N>
      </dc:creator>
      <dc:date rdf:parseType="Resource">
        <dcq:dateScheme>W3C-DTF</dcq:dateScheme>
        <dcq:dateType>created</dcq:dateType>
        <rdf:value>2001-04-01</rdf:value>
      </dc:date>
    </cmeta:annotation>
    <cmeta:annotation rdf:parseType="Resource">
      <cmeta:annotation_type>validation</cmeta:annotation_type>
      <rdf:value>Physiome level 2</rdf:value>
      <dc:creator rdf:parseType="Resource">
        <vCard:N rdf:parseType="Resource">
          <vCard:Family>Too</vCard:Family>
          <vCard:Given>Shaggy</vCard:Given>
        </vCard:N>
      </dc:creator>
      <dc:date rdf:parseType="Resource">
        <dcq:dateScheme>W3C-DTF</dcq:dateScheme>
        <dcq:dateType>created</dcq:dateType>
        <rdf:value>2001-03-28</rdf:value>
      </dc:date>
    </cmeta:annotation>
  </rdf:Description>
</rdf:RDF>

```

FIGURE 16: Recommended definition of modification and validation annotation metadata.

A CellML element that represents a list of alternative biological entities would probably be a “superclass” component, that will re-used multiple times in a model, each time to represent a different entity on the list of alternatives. For instance, a modeller might define a general “calcium-binding protein” component, and then re-use this component three times in his/her model: once to represent calmodulin, once to represent troponin C, and once to represent parvalbumin. [Note that the component re-use capabilities are not yet defined in CellML. They will be a part of the a future version of CellML. The list of alternative biological entities metadata construct is provided now for use in the future.]

Biological entity metadata is defined using a CellML-specific element, **<cmeta:bio_entity>**. A biological entity may be identified by name, database identifier, or both. Multiple database identifiers may be provided, but all except one must be marked “alternative”. The name of the biological entity is defined exactly as alternative names for CellML elements are defined (with the **<dc:title>** element, which may be qualified by a **<dcq:titleType>** element).

Each database identifier is stored in a **<cmeta:identifier>** element, which must be qualified by a **<cmeta:identifier_scheme>** element that identifies the database. The CellML metadata specification will control names for certain encoding schemes (see below). The **<cmeta:identifier>** element may also be qualified by a **<cmeta:identifier_type>** element. This element should have a value of “alternative” for all **<cmeta:identifier>** elements except for one, which is considered the primary identifier. This addresses a concern about allowing multiple database identifiers that might actually refer to different biological entities. Such an error may still occur, but marking all identifiers except one as “alternative” provides software a method by which to determine which identifier should be given precedence.

The CellML metadata specification will define the following encoding schemes:

- SWISS-PROT (SWISS-PROT protein database)
- GenBank (GenBank nucleic acid database)
- URI (URI for a web resource providing info about the biological entity)

Model authors and authors of processing software are free to define additional encoding schemes, by putting the **<entity_scheme>** element in an application-specific namespace. However, software claiming to be “CellML metadata compliant” is not required to recognize these schemes.

RDF containers can be used to indicate that a given CellML element is relevant for more than one biological entity. An **<rdf:Bag>** element can be used to indicate that the CellML element is relevant for an entire group of biological entities. An **<rdf:Alt>** element can be used to indicate that the CellML element can be relevant for one member of a group of entities. Note that the first member listed in the **<rdf:Alt>** element will be considered the preferred value. The use of the **<rdf:Bag>** element is shown in Figure 17. The use of the **<rdf:Alt>** element would be identical. “CellML metadata compliant” software will be required to recognize RDF containers in biological entity metadata. The use of RDF containers is preferred to simply repeating the **<cmeta:bio_entity>** element because it removes all ambiguity about how the group of biological entities relates to the referenced CellML element.

Figure 17 demonstrates the definition of biological entity metadata.

5.9 Copyright

The copyright metadata refers to the copyright that protects the CellML document, model, model component, or other CellML element. It is defined using the Dublin Core rights element (**<dc:rights>**), and therefore, a given CellML element can technically have multiple copyrights. However, the recommended practice is to include only one copyright for any given element.

Figure 18 demonstrates the definition of the copyright metadata.

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cmeta="http://www.cellml.org/2001/03/metadata#"
  xmlns:dc="http://purl.org/dc/elements/1.0/"
  xmlns:dcq="http://purl.org/dc/qualifiers/1.0/">

  <rdf:Description about="#cellml_element_id">
    <cmeta:bio_entity>
      <rdf:Bag>
        <rdf:li rdf:parseType="Resource">
          <dc:title>calmodulin</dc:title>
          <dc:title rdf:parseType="Resource">
            <dcq:titleType>alternative</dcq:titleType>
            <rdf:value>CaM</rdf:value>
          </dc:title>
          <cmeta:identifier rdf:parseType="Resource">
            <cmeta:identifier_scheme>SWISS-PROT</cmeta:identifier_scheme>
            <rdf:value>CALM_HUMAN</rdf:value>
          </cmeta:identifier>
        </rdf:li>
        <rdf:li rdf:parseType="Resource">
          <dc:title>troponin C</dc:title>
        </rdf:li>
        <rdf:li rdf:parseType="Resource">
          <cmeta:identifier rdf:parseType="Resource">
            <cmeta:identifier_scheme>SWISS-PROT</cmeta:identifier_scheme>
            <rdf:value>PRVA_HUMAN</rdf:value>
          </cmeta:identifier>
        </rdf:li>
      </rdf:Bag>
    </cmeta:bio_entity>
  </rdf:Description>
</rdf:RDF>

```

FIGURE 17: Recommended definition of biological entity metadata. The referenced CellML element represents the following group of proteins: calmodulin, troponin C, and parvalbumin (the protein identified by SWISS-PROT entry PRVA_HUMAN). The calmodulin biological entity has an alternative name and a database entry. The troponin C biological entity is only identified by name. The PRVA_HUMAN protein is only identified by database reference.

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.0/">

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

```

FIGURE 18: Recommended definition of copyright metadata

5.10 Publisher

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.

Publisher metadata is defined with the Dublin Core publisher element (**<dc:publisher>**), as shown in Figure 19. Multivalued publisher metadata is handled exactly as multivalued model builder metadata. Simple repetition of the element indicates that the people or organizations publish the resource independently. The use of RDF bag (**<rdf:Bag>**) or sequence (**<rdf:Seq>**) containers indicates that the people or organization publish the resource as an unordered or ordered group, respectively.

```

<rdf:RDF
  xmlns:rdf="http://www.w3.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 19: Recommended definition of publisher metadata. Note that the empty **about** attribute indicates that this metadata refers to the CellML *document* (as opposed to the model or a specific element in the model).

5.11 Mathematical Problem Type

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](http://gams.nist.gov/)⁵.

Mathematical problem type is defined using a CellML-specific element, **<cmeta:math_problem>**. (Note that earlier documents use an element called **<math_problem_type>**. The name of the element has been changed to avoid confusion with the type element qualifier.) This element is qualified by an

⁵<http://gams.nist.gov/>

encoding scheme element (**<cmeta:math_problem_scheme>**), which provides the name of the controlled vocabulary used for the classification. The allowed values of this element are themselves a vocabulary controlled by the CellML RDF schema. Currently, the only allowed value is “GAMS”, which indicates that the math problem classification is taken from the NIST’s [GAMS classification tree](#)⁶. Modellers are free to use a different controlled vocabulary for the math problem classifications by placing the **<math_problem_scheme>** element in an application-specific namespace. However, CellML metadata compliant software is not required to recognize any classification scheme other than the GAMS tree.

Figure 20 shows the recommended definition of mathematical problem type metadata.

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cmeta="http://www.cellml.org/2001/03/metadata#">

  <rdf:Description about="#cellml_element_id">
    <cmeta:math_problem rdf:parseType="Resource">
      <cmeta:math_problem_scheme>GAMS</cmeta:math_problem_scheme>
      <rdf:value>I1a</rdf:value>
    </cmeta:math_problem>
  </rdf:Description>
</rdf:RDF>

```

FIGURE 20: Recommended definition of the mathematical problem type metadata. The meaning of the value “GAMS” for the encoding scheme will be controlled by the CellML metadata specification.

5.12 Contributor

Contributor metadata indicates that a person contributed to a resource, but did not actually create it (an example of this is an editor).

Contributor metadata is defined using the Dublin Core contributor element, **<dc:contributor>**, as shown in Figure 21. Multivalued contributor metadata is handled exactly as multivalued model builder metadata. Simple repetition of the element indicates that the people contributed to the resource independently. The use of RDF bag (**<rdf:Bag>**) or sequence (**<rdf:Seq>**) containers indicates that the people contributed to the resource as an unordered or ordered group, respectively.

5.13 Description

Description metadata is a short description of the referenced resource.

Description metadata is defined with the Dublin Core description element, **<dc:description>**. This element is qualified by a type element (**<dcq:descriptionType>**), which may have a value of “abstract” or “table of contents”. The “abstract” type will probably be most common in CellML metadata.

Figure 22 shows how to define description metadata.

⁶<http://gams.nist.gov/>

```

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.0/"
  xmlns:vCard="http://www.w3.org/2001/vcard-rdf/3.0#">

  <rdf:Description about="#cellml_element_id">
    <dc:contributor rdf:parseType="Resource">
      <vCard:N rdf:parseType="Resource">
        <vCard:Family>Flintstone</vCard:Family>
        <vCard:Given>Fred</vCard:Given>
      </vCard:N>
    </dc:contributor>
  </rdf:Description>
</rdf:RDF>

```

FIGURE 21: Recommended definition of contributor metadata.

```

<rdf:RDF
  xmlns:rdf="http://www.w3.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="#cellml_element_id">
    <dc:description rdf:parseType="Resource">
      <dcq:descriptionType>abstract</dcq:descriptionType>
      <rdf:value>
        This element uses simple mass-action kinetics to describe the
        A + B <-> C + D reaction.
      </rdf:value>
    </dc:description>
  </rdf:Description>
</rdf:RDF>

```

FIGURE 22: Recommended definition of the Dublin Core description metadata.
