The CellML Metadata 1.0 Specification

Working Draft - 16 January 2002

3 Fundamentals

Metadata is defined within an `<rdf:RDF>` element as shown in Figure 1. The recommended best practice is to define the RDF namespace and any namespaces used by the enclosed metadata on the `<rdf:RDF>` element, even if these namespaces are already defined on the ancestor elements of the `<rdf:RDF>` element. This increases the re-usability of the RDF block. Furthermore, RDF processing software that does not recognise the CellML namespace can still parse a CellML document, extract the RDF blocks, and perhaps provide useful functionality with the information described in the RDF.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description rdf:about="#some_element_id">

    ...

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

FIGURE 1: An example of a metadata definition. The metadata about the element referenced by `"some_element_id"` has been left out for now.

An `<rdf:RDF>` element typically contains one or more `<rdf:Description>` elements, each of which defines an `rdf:about` attribute. The value of the `rdf:about` attribute must be a valid Uniform Resource Identifier[1] (URI). Metadata may be associated with the *document* it is defined in by assigning the `about` attribute an empty value (`""`). Metadata may be associated with an element in the current document by defining an attribute of type ID on that element and assigning the `about` attribute on the `<rdf:Description>` element a value equal to the value of the ID attribute preceded by a hash (#). In CellML, the attribute referred to is the `cmeta:id` attribute on any element.

RDF is processed as triples: a *resource* is assigned a *property* of a certain *value*. For instance, in Figure 2, the resource (the element referenced by `"Wilma_Flintstone"`) is assigned a property of `<spouse>` with a value of `Fred Flintstone`.

If you wanted to also indicate that Wilma's husband (resource) has a favourite hobby (property) of bowling (value), you could add an `rdf:id` attribute to the `<toon:spouse>` element and create another triple using a second `<rdf:Description>` element, as shown in Figure 3.

Figure 4 shows an alternative method to describe the two triples shown in Figure 3. The second `<rdf:Description>` element is embedded in the `<toon:spouse>` element to indicate that it is a new resource.

Figure 5 shows yet another way to describe the two triples. This example uses the `rdf:parseType` attribute with a value of `"Resource"` to introduce a new resource (instead of the `<RDF:Description>` element used in the previous example). The CellML Metadata Specification uses this method frequently because it is less verbose than the methods described in the previous two examples.

[1] http://www.ietf.org/rfc/rfc2396.txt

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:toon="http://www.cartoon_times.com/what_namespace">
  <rdf:Description rdf:about="#Wilma_Flintstone">
    <toon:spouse>Fred Flintstone</toon:spouse>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 2: An example of an RDF triple in which the element referred to by `"Wilma_Flintstone"` is the resource, `<spouse>` is the property describing the resource, and `Fred Flinstone` is the property value.

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:toon="http://www.cartoon_times.com/what_namespace/">
  <rdf:Description rdf:about="#Wilma_Flintstone">
    <toon:spouse rdf:id="spouse">Fred Flintstone</toon:spouse>
  </rdf:Description>
  <rdf:Description rdf:about="#spouse">
    <toon:favourite_hobby>bowling</toon:favourite_hobby>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 3: A set of two RDF triples, in which the second `<rdf:Description>` element describes Wilma's spouse, as indicated by the second `rdf:about` attribute.

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:toon="http://www.cartoon_times.com/what_namespace/">
  <rdf:Description rdf:about="#Wilma_Flintstone">
    <toon:spouse>
      <rdf:Description>
        <rdf:value>Fred Flintstone</rdf:value>
        <toon:favourite_hobby>bowling</toon:favourite_hobby>
      </rdf:Description>
    </toon:spouse>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 4: A set of two RDF triples, as in Figure 3. This example shows that a new resource can be created by embedding the `<rdf:Description>` element in the element it is describing.

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:toon="http://www.cartoon_times.com/what_namespace/">
  <rdf:Description rdf:about="#Wilma_Flintstone">
    <toon:spouse rdf:parseType="Resource">
      <rdf:value>Fred Flintstone</rdf:value>
      <toon:favourite_hobby>bowling</toon:favourite_hobby>
    </toon:spouse>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 5: The `rdf:parstType` attribute with a value of `"resource"` is another way to introduce a new resource. In this case the new resource is `<toon:spouse>`.

3.1 Containers

RDF provides the ability to indicate a sequence with the use of containers. The containers `<rdf:Bag>`, `<rdf:Seq>`, and `<rdf:Alt>`, denote an unordered sequence, an ordered sequence, and alternative choices, respectively. Figure 6 demonstrates the use of the `<rdf:Bag>` element. Each family member of the Jetsons is an equal member of the family. They are grouped together using the `<rdf:Bag>` element to show that they all belong to the same family.

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:toon="http://www.cartoon_times.com/what_namespace/">
  <rdf:Description rdf:about="#Jetsons">
    <toon:family_member>
      <rdf:Bag>
        <rdf:li>George</rdf:li>
        <rdf:li>Jane</rdf:li>
        <rdf:li>Judy</rdf:li>
        <rdf:li>Elroy</rdf:li>
        <rdf:li>Astro</rdf:li>
        <rdf:li>Rosie</rdf:li>
      </rdf:Bag>
    </toon:family_member>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 6: The `<rdf:bag>` element demonstrates that each person listed is an equal member of the same family.

The `<rdf:Seq>` element indicates that the members are in a specified order. In the example shown in Figure 7 the `<rdf:Seq>` element is used to list the relative ages of each person.

The `<rdf:Alt>` element indicates that any of the listed items may be chosen, and, generally, the first item listed is the preferred value. Figure 8 shows an example in which a choice is given of two supply companies: `Spacely's Space Sprockets` and `Cogswell's Coggs`.

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:toon="http://www.cartoon_times.com/what_namespace/">
  <rdf:Description rdf:about="#Jetsons">
    <toon:age>
      <rdf:Seq>
        <rdf:li>George</rdf:li>
        <rdf:li>Jane</rdf:li>
        <rdf:li>Judy</rdf:li>
        <rdf:li>Elroy</rdf:li>
        <rdf:li>Astro</rdf:li>
        <rdf:li>Rosie</rdf:li>
      </rdf:Seq>
    </toon:age>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 7: The `<rdf:Seq>` element implies that those listed are in order based on their age.

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:toon="http://www.cartoon_times.com/what_namespace/">
  <rdf:Description rdf:about="#companies">
    <toon:supply>
      <rdf:Alt>
        <rdf:li>Spacely’s Space Sprockets</rdf:li>
        <rdf:li>Cogswell’s Coggs</rdf:li>
      </rdf:Alt>
    </toon:supply>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 8: The `<rdf:Alt>` element implies that any one of the listed values is valid.

3.2 Dublin Core Elements and Qualifiers

The Dublin Core Metadata Elements and their corresponding qualifiers are listed in Table 3.

Each Dublin Core Element is given its own element in the Dublin Core namespace, as shown in Figure 9. The CellML Metadata Specification covers how to use the Dublin Core Qualifiers in later sections.

```xml
<rdf:RDF
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#">
  <rdf:Description rdf:about="#toon_times">
    <dc:title>Toonville Times</dc:title>
    <dc:creator>R.J. Gopher</dc:creator>
    <dc:date>2001-10-18</dc:date>
  </rdf:Description>
</rdf:RDF>
```

FIGURE 9: Each Dublin Core Metadata Element is its own element in the Dublin Core namespace.

DC Metadata Element	**Element Refinement(s)**	**Element Encoding Scheme(s)**
Title	Alternative	-
Creator	-	-
Subject	-	LCSH MeSH DDC LCC UDC
Description	Table of Contents Abstract	-
Publisher	-	-
Contributor	-	-
Date	Created Valid Available Issued Modified	DCMI Period W3C-DTF
Type	-	DCMI Type Vocabulary
Format	Extent	-
	Medium	IMT
Identifier	-	URI
Source	-	URI
Language	-	ISO 639-2 RFC 1766
Relation	Is Version Of Has Version Is Replaced By Replaces Is Required By Requires Is Part Of Has Part Is Referenced By References Is Format Of Has Format	URI
Rights	-	-
Coverage	Spatial	DCMI Point ISO 3166 DCMI Box TGN
	Temporal	DCMI Period W3C-DTF

TABLE 3: The Dublin Core Metadata Element Set and their qualifiers.