

Meeting Minutes 13 March 2001

Connections and Variable Mapping

Authors:

Melanie Nelson (Physiome Sciences Inc.)

Warren Hedley (Bioengineering Institute, University of Auckland)

Contributors:

Poul Nielsen (Bioengineering Institute, University of Auckland)

Yi Ge (Physiome Sciences Inc.)

Kam Jim (Physiome Sciences Inc.)

1 Introduction

As Warren and Melanie closed in on a draft of the specification for CellML version 1.0, Physiome's software developers actually began implementing code that can write this version of CellML. Yi Ge was in charge of making Physiome's Cell Editor software write CellML. After reading the 2001-03-02 draft version of the CellML specification, he had two suggestions about how we could make CellML easier for him to implement, and more intuitive in general. Now that the draft specification has been made public, the CellML development team can turn its attention to these suggestions. This progress report covers his comments on improving CellML's connection and variable mapping scheme, and the changes that would need to be made to the specification to formally specify this. A later document will cover Yi's comments on the difficulty of defining hierarchies using the current grouping scheme, and a new scheme that addresses these comments.

2 Yi's Comments

Yi pointed out that under the current connection system, it would be easy for a user to inadvertently introduce errors into the mapping of variables, particularly when editing CellML files manually. This is because our method depends entirely on the order of the lists of variables in each component reference. The first variable in the list for the first component is mapped to the first variable in the list for the second component. It would be very easy for someone to map the wrong two variables together. This sort of bug would be hard to find, since the modeller would have to count through the two lists to find out which variables are mapped to each other. An example of the current method of mapping variables is shown in Figure 1.

3 A New Method

Warren proposed a new connection definition syntax, which is based on a `<map_variable>` element, which has two required attributes that name the two variables to be mapped to each other. The `variable_1` attribute names the variable from the component that is referenced by the first `<component_ref>` element in the connection element, and the `variable_2` attribute names the variable that is referenced by the second `<component_ref>` element. The connection defined using the old syntax in Figure 1, is redefined using the new syntax in Figure 2.

Melanie initially thought that it might make sense to change these attribute names to `output_variable` and `input_variable`, and have them refer to the variables based on their role in the mapping, rather than the component to which they belong. However, Kam pointed out that it is necessary to explicitly specify which variable belongs to which component because we must be able to handle the case in which both components declare an output variable with the same name. These two variables cannot be mapped to each other,

```

<connection>
  <component_ref component="my_component_1">
    <variable_ref variable="A" />
    <variable_ref variable="B" />
    <variable_ref variable="C" />
  </component_ref>
  <component_ref component="my_component_2">
    <variable_ref variable="X" />
    <variable_ref variable="Y" />
    <variable_ref variable="Z" />
  </component_ref>
</connection>

```

FIGURE 1: The definition of a connection as proposed in the 2001-03-02 draft of the CellML specification. Note that variable mapping relies on the order of the two lists of variable references.

but could be mapped to other (input) variables in the opposing component. This situation could not be handled if we rely only on the value of the **<variable>** element's **name** and interface attributes to determine to which component a mapped variable belongs. Therefore, it seems best to stick with the **variable_1** and **variable_2** attributes, and continue to require processing software to go to the **<variable>** elements that originally declare the two variables in the mapping to determine which variable is the output variable and which is the input variable.

```

<connection>
  <component_ref component="my_component_1" />
  <component_ref component="my_component_2" />
  <map_variable variable_1="A" variable_2="X" />
  <map_variable variable_1="B" variable_2="Y" />
  <map_variable variable_1="C" variable_2="Z" />
</connection>

```

FIGURE 2: A possible new method for describing connections and variable mappings. Each mapping is described using a single **<map_variable>** element, where the **variable_1** attribute references the name of a variable in the component referenced by the first **<component_ref>** element, and the **variable_2** attribute references the name of a variable in the component referenced by the second **<component_ref>** element.

The scheme shown in Figure 2 has the following advantages:

- It is very easy to see which variables are being mapped together.
- It does not depend on the ordering of the variable references.
- It is fairly concise.

It has the following disadvantages:

- The method it uses to reference variables is different from the method used in other places (such as in the `<reaction>` element).
- It still depends on the order of the `<component_ref>` elements.

4 A Different Solution

The solution outlined in Section 3 addresses Yi's original concerns, but left some members of the Auckland team with a bad taste in their mouths (Melanie thinks this is because they drank too much beer and/or Midori the night before). They argued that the `variable_1` and `variable_2` attributes look too much like a hack.

Well, there is a standard way of referencing things in CellML: use a "`<thing_ref>`" element. Melanie proposed using this scheme here, which will give the XML shown in Figure 3. The `<map_variable>` element now contains two `<variable_ref>` elements. Each `<variable_ref>` element defines a `variable` attribute, (which names the variable being referenced) and a `component` attribute (which names the component to which the referenced variable belongs). The `<component_ref>` elements are retained to allow enforcement of the rule that there be only a single connection between any two components.

```

<connection>
  <component_ref component="my_component_1" />
  <component_ref component="my_component_2" />
  <map_variable>
    <variable_ref component="my_component_1" variable="A" />
    <variable_ref component="my_component_2" variable="X" />
  </map_variable>
  <map_variable>
    <variable_ref component="my_component_1" variable="B" />
    <variable_ref component="my_component_2" variable="Y" />
  </map_variable>
  <map_variable>
    <variable_ref component="my_component_1" variable="C" />
    <variable_ref component="my_component_2" variable="Z" />
  </map_variable>
</connection>

```

FIGURE 3: A second possible new method for describing connections and variable mappings. Each mapping is described using a single `<map_variable>` element, which contains `<variable_ref>` elements that reference the variables that are to be mapped to each other. Note that the mapping no longer depends on ordering of elements at all.

The mapping no longer depends on the order of any of the elements. However, the observant reader will notice that this second scheme is significantly more verbose than the first one, but does not really add any new information. This is its main drawback.

The scheme shown in Figure 3 has the following advantages:

- It does not depend on the order of either the variable or component references.
- All references are made using the same method as is used elsewhere in CellML.

It has the following disadvantages:

- It adds verbosity without adding more information.

5 Which Method to Use?

The Physiome programmers are happy to implement either method (and just wish we'd stop dithering and make up our minds). I'll leave it to a vote of the interested parties.

E-mail questions, criticism, submissions or info to info@cellml.org
Input document last modified : Mon Feb 02 15:25:02 NZDT 2004