

# Meeting Minutes 9 February 2001

## Possible Problems With The `<reaction>` Element

Author:

Warren Hedley (Bioengineering Institute, University of Auckland)

Contributor:

Melanie Nelson (Physiome Sciences Inc.)

### 1 Introduction

Over the last week, Melanie and I have been re-writing the reactions and units parts of the specification (yet again) trying to split the rules section down into the smallest possible subsections, refine the format of the rules, and make their specification more consistent. This process resulted in the highlighting of some problems in the reactions specification. This document attempts to explain the basis of those problems, and gives some possible solutions with arguments for and against.

### 2 Things We Missed

The following rule should be added to the **Allowed values of the `role` attribute** section.

- The value of the `role` attribute on a `<role>` element must be unique across all `<role>` elements within the current `<variable_ref>` element.

### 3 Is the delta variable misnamed?

Both Poul Nielsen and Scott Lett have correctly pointed out that the variable we have been informally calling the “delta variable” is misnamed. This variable actually represents the *rate* of change of a chemical species, and would therefore more correctly be named something like the “rate of change of concentration” variable. However, a `rate_of_concentration_change_variable` attribute has a name too unwieldy even for Warren, who is fond of long names.

Warren and Melanie have racked their brains for a better name, to no avail. Unless someone else comes up with a suggestion, the variables representing the rate of change of a chemical species will retain the name “delta variable”. However, a comment will be added to the specification indicating that this is technically a misnomer.

### 4 Implications Of The `reversible` Attribute

The section of the reactions part of the specification with the above name was initially a sub-rule in the reaction element section. It was then moved to its own section (the problem is that there are some rules associated with this section, so it is not clear where to put it). The section text is reproduced below (where the associated figure has been omitted):

If the `reversible` attribute has a value of "yes", it is assumed that all reactants in the forward direction are products in the reverse direction, and vice versa. Similarly, all products in the forward direction are assumed to be reactants in the reverse direction, and vice versa. It is not necessary to explicitly specify this with separate `<role>` elements for the forward and

reverse directions. The two `<reaction>` elements shown in Figure ?? are exactly equivalent. The more compact formulation shown in the first `<reaction>` element is the recommended best practice. In fact, use of the more verbose formulation shown in the second `<reaction>` element is specifically not recommended, because it creates the possibility for inconsistencies between the `delta_variable` and `stoichiometry` attributes specified on the forward roles and those specified on the reverse roles. If the more verbose formulation is used, the values of the `delta_variable` and `stoichiometry` attributes for the forward and reverse reactant and product roles for a given species variable must be identical.

The last line stipulates that if both reactant and product roles are specified in opposing directions for the same species within a reversible reaction, then the associated delta variable and the rate of change must be equal. Let us quickly consider the alternatives, where we use the term “*equivalent*” to refer to reactant and product roles specified in opposing directions for the same species within a reversible reaction.

If two equivalent `<role>` elements can have different values of their `delta_variable` attributes, then it would effectively imply that a single species is changed using a different delta variable in each reaction direction. It might well be that the two different delta variables are passed back to the same species component, but the reaction component does not “know” this. A simple reaction demonstrating the problem is shown in Figure 1. That CellML fragment defines the reversible reaction  $A \rightleftharpoons C$ . In the forward direction, the concentration of A is altered using the delta variable `delta_A`, yet in the reverse direction it is altered using `delta_B`. Because we have no way of forcing the user to wire these variables to the same species component (we don’t define rules where the contents of one component restrict the content of another component, because this would hinder component re-use), we can not be confident that the reaction is really a consistent reversible reaction. So this rule is a good one.

---

```

<reaction reversible="yes">
  <variable_ref variable="A">
    <role
      role="reactant" direction="forward"
      delta_A="delta_A" stoichiometry="1" />
    <role
      role="product" direction="reverse"
      delta_A="delta_B" stoichiometry="1" />
  </variable_ref>
  <variable_ref variable="C">
    <role
      role="product" direction="forward"
      delta_A="delta_C" stoichiometry="1" />
  </variable_ref>
  <variable_ref variable="r">
    <role role="rate" />
    <math> ... </math>
  </variable_ref>
</reaction>

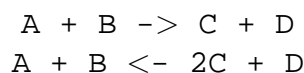
```

FIGURE 1: An example of two equivalent `<role>` elements defined within a single `<variable_ref>` element, where the value of the `delta_variable` attributes differs. The reaction defined is  $A \rightleftharpoons C$ .

**This is not valid CellML!**

---

If two `<role>` elements can have different values of their `stoichiometry` attributes, then it would imply that the following two chemical expressions could be a single reversible reaction.



Chemically, these are two different reactions. Such a system should rightly (and could easily) be defined as two separate reactions, so restricting the **stoichiometry** attribute on equivalent **<role>** elements to the same value is a valid rule.

The only point in defining two equivalent **<role>** elements would then be for the specification of math within each **<role>** element. The upcoming rules on the use of mathematics within **<role>** elements will specify something like: “any mathematics defined inside a **<role>** element with a **role** attribute value of "reactant" or "product" must calculate the change in concentration of the referenced species variable and any required intermediate results.” So we would be left with the same situation as if there were two differing **stoichiometry** attribute values. It makes sense to prevent people from defining mathematics in more than one of any two equivalent **<role>** elements.

At this point, it suddenly becomes obvious that there is no point allowing the definition of equivalent **<role>** elements in the first place, as it is completely pointless duplication of information. Bearing this in mind, it also becomes obvious that **<role>** elements with a **role** attribute value of "reactant" or "product" need (and should) only ever be specified in the "forward" direction. So we add the following rule to the **Allowed use of the direction attribute** section.

- A **<role>** element for which the **role** attribute has a value of "reactant" or "product" may only have a value of "forward" for the **direction** attribute. (This is for clarity, and to prevent the definition of inconsistent chemistry that might occur if model authors were allowed to define equivalent components).

Furthermore, this necessitates the addition of the following rule to the **Proper use of the role attribute** section.

- If a **<role>** element has a **role** attribute value of "reactant", then no other **<role>** element within the same parent **<variable\_ref>** element may have a **role** attribute value of "product", i.e., a species may not be explicitly defined to be both a product and a reactant, although this is implied in a reversible reaction.

## 5 Multiple Reactions In A Component

The current reactions part of the specification allows but discourages the definition of multiple **<reaction>** elements within a single component. This policy dates from November 2000, when the **<reaction>** element only defined the role of each participating species. The result of placing both reactions from the [two reaction model](#)<sup>1</sup> example into a single component element is shown in Figure 2. Note that any chemical species used in more than one reaction must have a unique delta variable for each reaction. For instance, in Figure 2, species **C** has two associated delta variables. It is up to the modeller to decide whether each delta variable should be passed back to the component that controls the concentration variable for this species, or if the multiple delta variables should instead be summed to produce a “global” delta variable for the entire component. If the modeller wishes to sum the various delta variables for the different reactions, he/she must do so explicitly, using MathML.

Multiple reactions are allowed in a single component because no one can think of a compelling reason to disallow this. However, the recommended best practice is to create a component for each reaction, because this makes it easier to reuse reactions.

<sup>1</sup>[http://www.cellml.org/examples/examples/signal\\_transduction\\_models/basic\\_reaction\\_models/two\\_reaction\\_model\\_doc.html](http://www.cellml.org/examples/examples/signal_transduction_models/basic_reaction_models/two_reaction_model_doc.html)

```

<component name="two_reactions">
  <variable name="A" public_interface="in" units="concentration_units" />
  <variable name="B" public_interface="in" units="concentration_units" />
  <variable name="C" public_interface="in" units="concentration_units" />
  <variable name="D" public_interface="in" units="concentration_units" />
  <variable name="E" public_interface="in" units="concentration_units" />
  <variable name="F" public_interface="in" units="concentration_units" />

  <variable name="delta_A" public_interface="out" units="flux_units" />
  <variable name="delta_B" public_interface="out" units="flux_units" />
  <variable name="delta_C_1" public_interface="out" units="flux_units" />
  <variable name="delta_C_2" public_interface="out" units="flux_units" />
  <variable name="delta_D" public_interface="out" units="flux_units" />
  <variable name="delta_E" public_interface="out" units="flux_units" />
  <variable name="delta_F" public_interface="out" units="flux_units" />

  <variable name="r_1" units="flux_units" />
  <variable name="r_2" units="flux_units" />

  <reaction>
    <variable_ref variable="A">
      <role role="reactant" delta_variable="delta_A" stoichiometry="1" />
    </variable_ref>
    <variable_ref variable="B">
      <role role="reactant" delta_variable="delta_B" stoichiometry="1" />
    </variable_ref>
    <variable_ref variable="C">
      <role role="product" delta_variable="delta_C_1" stoichiometry="2" />
    </variable_ref>
    <variable_ref variable="D">
      <role role="product" delta_variable="delta_D" stoichiometry="1" />
    </variable_ref>
    <variable_ref variable="r_1">
      <role role="rate" />
      <math> ... </math>
    </variable_ref>
  </reaction>

  <reaction>
    <variable_ref variable="C">
      <role role="reactant" delta_variable="delta_C_2" stoichiometry="1" />
    </variable_ref>
    <variable_ref variable="E">
      <role role="reactant" delta_variable="delta_E" stoichiometry="1" />
    </variable_ref>
    <variable_ref variable="F">
      <role role="product" delta_variable="delta_F" stoichiometry="1" />
    </variable_ref>
    <variable_ref variable="r_2">
      <role role="rate" />
      <math> ... </math>
    </variable_ref>
  </reaction>

</component>

```

FIGURE 2: A component containing two reaction elements, which define  $A + B \leftrightarrow 2C + D$  and  $C + E \leftrightarrow F$  respectively. **Defining multiple reactions within a component is not good practice!**

## 6 Prevention Of Inconsistent Mathematics Within A Reaction

As they currently stand, the rules in the reactions part of the CellML specification do not sufficiently restrict the modeller's ability to create inconsistent mathematics within a single `<reaction>` element, where these mathematics can be implicitly or explicitly defined. Although these rules would be covered by the catch-all "no inconsistent mathematics" clause in the mathematics part of the specification, it would be polite to explicitly forbid some of the more likely causes of error.

One obvious flaw with the implied mathematics schema is the lack of a rule preventing multiple species from having the same delta variables. The specification already states that `delta_variable` attribute may only be present on reactant or product `<role>` elements. With the role uniqueness rules defined already in this document, we have ensured that a `<variable_ref>` element may only contain one reactant or product `<role>` element, and hence one `delta_variable` attribute. However we have not prevented reactant or product role elements within different species references from referencing the same delta variable. Clearly if an equation is implied every time that a `delta_variable` attribute is defined, then if several `delta_variable` attributes share the same value, multiple equations are defined, where these will duplicate or contradict each other depending on the `role` attribute value of the parent element. The following rule should be added to the **Allowed values of the `delta_variable` attribute** section.

- Every `delta_variable` attribute defined on `<role>` elements within a `<component>` element must have a unique value.

## 7 Rules For Mathematics In `<reaction>` Elements

The 5 February edition of the reactions part of the CellML specification did not even attempt to tackle the use of MathML to define equations for the specification of reaction rate and species concentration change. A rules subsection needs to be added to the specification listing guidelines and rules for this.

The following rules apply to the use of mathematics in `<reaction>` elements:

- Within a `<reaction>` element, mathematics can only be stored in a `<variable_ref>` element. Furthermore, the `<variable_ref>` element must have a `role` attribute value of "rate", "reaction", or "product".
- Mathematics in a `<variable_ref>` element with a `role` attribute value of "rate" must define the kinetic rate equation for the reaction. It may also define any necessary intermediate equations.
- Mathematics in a `<variable_ref>` element with a `role` attribute value of "reactant" or "product" must define the relationship of the rate of change of the species represented by the referenced variable to the general reaction rate. It may also define any necessary intermediate equations.
- The kinetic rate law must always be defined in the forward reaction direction, where the change of concentration of reactants has the same sign as the general reaction rate.
- In a reversible reaction, the rate variable must be calculated to include the contributions of both the forward and reverse reactions. (If this is not possible, the reaction should be split into two irreversible reactions.)

The following guidelines define the recommended best practice for using mathematics in a `<reaction>` element:

- Although model authors are given the freedom to define an arbitrary mathematical relationship between the general rate and the rate of change of the concentration of a chemical species, it is recommended that the `delta_variable` and `stoichiometry` attributes be used instead in all cases

where the mathematics implied by these attributes is sufficient and correct. The CellML development team cannot think of a chemically valid example for which it would be necessary to explicitly define mathematics relating the rate of change of the concentration of a chemical species to the general reaction rate (i.e., we can think of no case where the math implied by the **delta\_variable** and **stoichiometry** attributes would not be correct and sufficient).

- Any explicit mathematical relationship defined between the general reaction rate and the rate of change of a particular chemical species should be defined in the **<variable\_ref>** element that refers to the variable representing that species.
- The mathematical definition of the kinetic rate law for a reaction should be placed in the **<variable\_ref>** element that refers to the variable representing the general reaction rate (i.e., the **<variable\_ref>** element for which the **<role>** element has a **role** attribute value of rate).