The Hodgkin Huxley Squid Axon Model 1952

Authors:
Warren Hedley (Bioengineering Institute, University of Auckland)
Geoff Carden (Bioengineering Institute, University of Auckland)
Autumn Cuellar (Bioengineering Institute, University of Auckland)
Contributors:
Greg Sands (Bioengineering Institute, University of Auckland)
David Nickerson (Bioengineering Institute, University of Auckland)

1 Introduction

In a series of papers published in 1952, A.L. Hodgkin and A.F. Huxley presented the results of a series of experiments in which they investigated the flow of electric current through the surface membrane of the giant nerve fibre of a squid. In the summary paper of the Hodgkin and Huxley model, the authors developed a mathematical description of the behaviour of the membrane based upon these experiments, which accounts for the conduction and excitation of the fibre. The form of this description has been used as the basis for almost all other ionic current models of excitable tissues, including Purkinje fibres and cardiac atrial and ventricular muscle.

The summary paper is:

A Quantitative Description of Membrane Current and its Application to Conduction and Excitation in Nerve, A. L. Hodgkin and A. F. Huxley, 1952, *J. Physiol*, 117, 500-544

2 Status of the Model and Documentation

The CellML description of the Hodgkin-Huxley model (which can be downloaded in various formats as described in Section 6) and documentation were revised and updated in July 2001 to conform to the 18 May 2001 versions of the CellML 1.0 specification[1] and CellML Metadata 1.0 specification[2].

The model description itself is intended to demonstrate the best practices for the definition of electrophysiological models using CellML, as intended by the authors of the CellML specification.

The metadata included in the CellML description should not be regarded as a demonstration of best practice for metadata. It is neither complete nor final, as the version of the CellML Metadata specification on which it is based has a status of *Research Summary*.

3 Hodgkin Huxley Model Overview

The current flow across the cell membrane depends on the capacitance of the membrane and the resistance of the ion channels. The total ionic current is represented by the sum of the sodium current, potassium current and a small leakage current. The leakage current represents the collective contribution of ions such as chloride and bicarbonate. Hodgkin and Huxley developed an electrical circuit diagram to represent the ion flows in their model. A modified version of this diagram is shown in Figure 1, where the currents have been reversed to be consistent with the convention proposed by Noble, commonly adopted for cardiac muscle.

The current is represented mathematically as the sum of the resistive and capacitive components. The capacitive component I_{cap} is derived from Ohm's law, where C_m and V_m denote the membrane capacitance and trans-membrane voltage, respectively.

[1] http://www.cellml.org/public/specification/20010518/index.html
[2] http://www.cellml.org/public/metadata/20010518/index.html

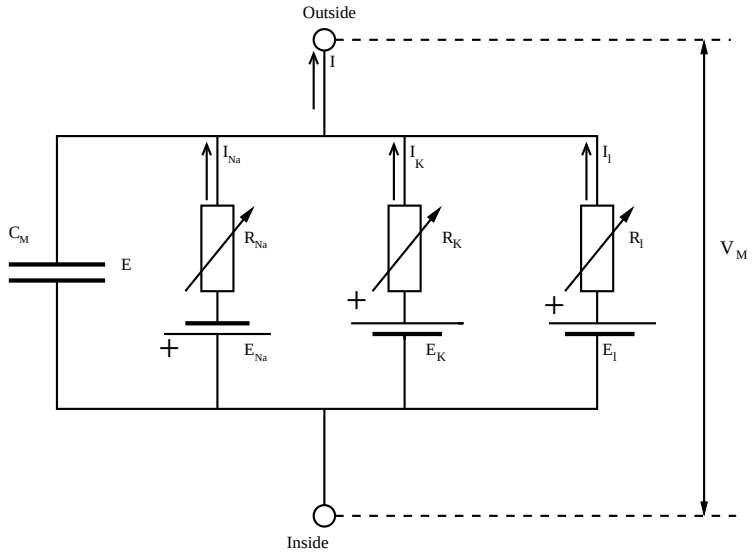

FIGURE 1: An electrical circuit diagram describing the current flows across the cell membrane that are captured in the Hodgkin Huxley model.

$$I_{cap} = C_m \cdot \frac{dV_m}{dt} \tag{1}$$

The resistive components are characterised in the original model as conductances (g), the reciprocals of the resistances. The currents are dependent on the transmembrane voltage (V_m) and the equilibrium potentials (E) of the individual ions. There are two forces driving the ion flux: the chemical gradient of the ions and the electrical gradient resulting from the different concentrations of charged particles on each side of the membrane. The equilibrium potential or reversal potential arises from equating the electrical and chemical potentials. The difference between the equilibrium potential for an ion and the transmembrane potential is the driving force for the ionic flow. The resistive currents can therefore be written:

$$I_{ion} = g_{ion} \cdot (V_m - E_{ion}) \tag{2}$$

The experiments suggested that g_{Na} and g_K were time dependent but the conductances of the other ions could be considered to be constant. Depolarisation of the membrane causes a fast transient increase in sodium conductance and a gradual and sustained increase in potassium conductance. The time dependence is represented by a gate variable or activation coefficient x which indicates the probability of a gate in the channel being open. The conductance for a time dependent channel can thus be written in terms of its gate variable x ($0 <= x <= 1$) and a maximum conductance $g_{ion,max}$ as follows:

$$g_{ion} = g_{ion,max} \cdot x \tag{3}$$

where x satisfies the differential equation:

$$\frac{dx}{dt} = \alpha_x \cdot (1 - x) - \beta_x \cdot x \tag{4}$$

α_x and β_x are rate coefficients which are a non-linear functions of voltage and have the units of 1/time.

The activation coefficient can be raised to the power of k if there are k gates in series in one ion channel. Channels can also be given more then one gate variable to describe activation-inactivation behaviour. This gives current equations of the form:

$$I_{ion} = g_{ion,max} \cdot x \cdot y \cdot (V_m - E_{ion}) \tag{5}$$

where x is the activation variable and y is the inactivation variable. The rate coefficients for the inactivation variable α_y and β_y are in general much slower than the rate coefficients for the activation variable. An example of this type of channel is the fast sodium channel involved in the depolarisation of the membrane to generate the upstroke of the action potential.

3.1 Sodium conductance

The sodium conductance control theory that Hodgkin and Huxley chose to adopt involved two control particles. The first moves out of an inhibitory site when the transmembrane potential reaches a threshold allowing sodium ions to flow though the channel. A second inhibitory particle moves slowly to occupy the inhibitory site and stop the ion flow. These are represented by two variables each governed by a first order differential equation. The conductance is then expressed:

$$g_{Na} = g_{Na,max} \cdot m^3 \cdot h \tag{6}$$

where m is an activation coefficient, h is an inactivation coefficient and $g_{Na,max}$ is the maximum sodium conductance. Both activation and inactivation variables are governed by the differential equations

$$\frac{dm}{dt} = \alpha_m \cdot (1 - m) - \beta_m \cdot m \tag{7}$$

$$\frac{dh}{dt} = \alpha_h \cdot (1 - h) - \beta_h \cdot h \tag{8}$$

where the rate constants α and β are functions of voltage but not of time. Expressions for rate constants were found by fitting curves to experimental data from nerve fibres to give:

$$\alpha_m = \frac{0.1 \cdot (V_m + 25.0)}{exp\,(0.1 \cdot (V_m + 25.0)) - 1.0} \tag{9}$$

$$\beta_m = 4 \cdot exp\left(\frac{V_m}{18.0}\right) \tag{10}$$

$$\alpha_h = 0.07 \cdot exp\left(\frac{V_m}{20.0}\right) \tag{11}$$

$$\beta_h = \frac{1.0}{exp\,(0.1 \cdot (V_m + 30.0)) + 1.0} \tag{12}$$

The total sodium current is then given by

$$\begin{aligned} i_{Na} &= g_{Na} \cdot (V_m - E_{Na}) \\ &= g_{Na,max} \cdot m^3 \cdot h \cdot (V_m - E_{Na}) \end{aligned} \tag{13}$$

where E_{Na} is the reversal potential for the sodium channel.

3.2 Potassium conductance

The potassium conductance can be modelled similarly, but there are only activation type channels for potassium. The equations are:

$$g_K = g_{K,max} \cdot n^4 \tag{14}$$

$$\frac{dn}{dt} = \alpha_n \cdot (1 - n) - \beta_n \cdot n \tag{15}$$

$$\alpha_n = \frac{0.01 \cdot (V_m + 10.0)}{exp(0.1 \cdot (V_m + 10.0)) - 1.0} \tag{16}$$

$$\beta_n = 0.125 \cdot exp\left(\frac{V_m}{80.0}\right) \tag{17}$$

where n is the activation coefficient for the potassium channel, $g_{K,max}$ is the maximum conductance of potassium and α_n and β_n are the voltage dependent rate constants.

The total potassium current is then given by

$$\begin{aligned} i_K &= g_K \cdot (V_m - E_K) \\ &= g_{K,max} \cdot n^4 (V_m - E_K) \end{aligned} \tag{18}$$

where E_K is the reversal potential for the potassium channel.

3.3 Leakage current

The full Hodgkin Huxley model also included the leakage current to maintain the constant resting membrane potential in the absence of any depolarisation. The leakage current is given by

$$i_L = g_L \cdot (V_m - E_L) \tag{19}$$

where E_L is the reversal potential for the leakage current.

3.4 Putting it all together

The full Hodgkin Huxley membrane current model is assembled by now combining the capacitive and resistive currents across the membrane into a single equation.

$$I = C_m \cdot \frac{dV_m}{dt} + i_{Na} + i_K + i_L \tag{20}$$

All the equilibrium potentials are related to the resting membrane potential E_R. The set of values used in the model are given in the table Table 1.

4 Representing the Model in CellML

In CellML any model is represented by a network of components, typically chosen to reflect the natural physical entities in the system. Information is passed from one component to another through connections. A diagram showing the network defined in the CellML description of the Hodgkin-Huxley model is shown in Figure 2.

Constant	Units	Value
C_m	μF cm^{-2}	1.0
E_R	mV	-75.0
E_{Na}	mV	E_R + 115.0
E_K	mV	E_R - 12.0
E_L	mV	E_R + 10.613
$g_{Na,max}$	mS cm^{-2}	120.0
$g_{K,max}$	mS cm^{-2}	36.0
g_L	mS cm^{-2}	0.3

TABLE 1: Values for constants in the Hodgkin-Huxley equations.

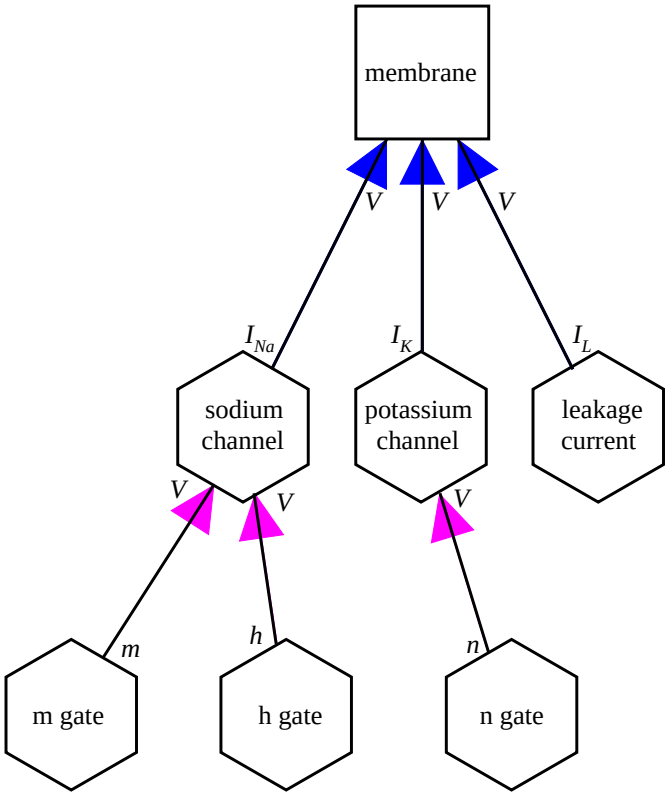

FIGURE 2: The network defined in the CellML description of the Hodgkin-Huxley model. Where a variable is passed from one component to another along a connection, the name of the variable is shown next to the component in which it is declared with a public or private interface value of `"out"`. The **`environment`** component discussed in Section 5.3.1 is not included in this diagram. A key describing the significance of the shapes of the components and the colours of the connections between them is in the notation guide.

At the highest level, the model consists of four components with identifiers `membrane`, `sodium_channel`, `potassium_channel` and `leakage_current`. The membrane physically contains the two channels and the leakage current, as indicated by the blue arrows in Figure 2. The channels and current act independently and are not connected to each other. Both the sodium and potassium channels encapsulate *and* contain further components: the sodium channel contains two components representing the m and h gates, and the potassium channel contains a component representing the n gate. The addition of an encapsulation relationship between the sodium channel and the m and h gates informs modellers and processing software that the m and h gates are important parts of the sodium channel model. It also prevents any other components that aren't also encapsulated by the sodium channel component from connecting to the m and h gates, effectively hiding them from the rest of the model.

CellML processing software might choose to use the containment and encapsulation information in the model definition for the rendering of the model. An initial rendering might have the sodium channel, potassium channel, and leakage currents drawn inside the membrane component. Some label might indicate that the sodium and potassium channel components contain hidden detail. When the user double clicks on the sodium channel, they are shown the hidden m and h gate components.

The breakdown of the model into components and the definition of encapsulation and containment relationships between them is somewhat arbitrary. When considering how a model should be broken into components, modellers are encouraged to consider which parts of a model might be re-used, and how the physiological elements of the system being modelled are naturally bounded. Containment relationships should be used to provide simple rendering information for processing software (ideally, this will correspond to the layout of the physical system), and encapsulation should be used to group sets of components into sub-models.

5 Analysis of the CellML Code

5.1 The `<model>` element

The root element of the XML document containing the CellML definition of the Hodgkin-Huxley model is the `<model>` element shown in Figure 3. (At this point, those not familiar with XML may want to consult the quick introduction to XML[3] for help with understanding XML terms.) The `<model>` element defines a `name` attribute that allows this model to be unambiguously referenced by other models. For instance, this would be necessary if this model were to be combined with other models or partial models to create a larger model. It also defines an `id` attribute in the CellML Metadata namespace (which is mapped to the `cmeta` prefix). This `id` attribute is of XML type ID and may be used to attach metadata to this element, as discussed in Section 5.6.

The `<model>` element also defines two XML namespaces. The first is the CellML namepsace, which is set to the default namespace and also mapped explicitly to the `cellml` prefix. The default namespace declaration (which looks like an `xmlns` attribute) means that the `<model>` element is itself in the CellML namespace. As such, it can be identified by any namespace-aware XML parser as a CellML element. In addition, any child elements that are not placed in another namespace, or that do not declare a new default namespace, will also be in the CellML namespace. Mapping the CellML namespace to the `cellml` prefix simplifies the declaration of elements and attributes in the CellML namespace later in the file. For instance, the `cellml` prefix can be used to place the `units` attribute on the `<cn>` element (which is in the MathML namespace) in the CellML namespace. The declaration of the CellML namespace as both the default namespace and as a namespace mapped to the `cellml` prefix is recommended practice for any `<model>` element.

[3] http://www.cellml.org/examples/introduction/xml_guide.html

```
<model
    name="hodgkin_huxley_squid_axon_1952"
    cmeta:id="hodgkin_huxley_squid_axon_1952"
    xmlns="http://www.cellml.org/cellml/1.0#"
    xmlns:cellml="http://www.cellml.org/cellml/1.0#"
    xmlns:cmeta="http://www.cellml.org/metadata/1.0#">

  ...

</model>
```

FIGURE 3: The root element of the XML document containing the CellML description of the Hodgkin-Huxley model is the **<model>** element shown above.

The model element also defines the CellML Metadata namespace and maps it to the **cmeta** prefix. This allows the **id** attribute to be placed in the CellML Metadata namespace.

5.2 Declaration of units

The CellML specification defines a standard dictionary of units[4] that can be used in CellML models without further definition. This dictionary consists of the base SI units (as defined by the Bureau International des Poids et Mesures[5]) and some additional units commonly found in the types of biological models likely to be defined using CellML.

A modeller who wishes to use units not declared in the standard dictionary can define additional units in the CellML model document. The units definitions from the Hodgkin Huxley model are shown in Figure 4. These units definitions occur immediately within the root **<model>** element, giving them document-wide scope.

The units definitions are declared with a set of **<units>** elements. Each **<units>** element must have a **name** attribute, which declares an identifier that can then be referenced in the **units** attribute on **<variable>** and **<cn>** elements (**<cn>** elements enclose bare numbers in MathML).

Each **<units>** element may contain one or more **<unit>** elements. The units defined by the **<units>** element are a combination of the contents of these **<unit>** elements. Section 5.2.2[6] of the CellML specification explains how the attributes for each **<unit>** element contribute to a new units definition. CellML processing software may choose to use these units definitions to verify the consistency of units across connections (inserting scale-factors, if appropriate) and check the dimensions of equations, as described in Section 5.2.6[7] and Section 5.2.7[8] of the CellML specification, respectively.

5.3 Model components

The CellML definition of the Hodgkin-Huxley model consists of a total of eight components. Seven of these correspond to physiological entities and were shown in the diagram in Figure 2. Within each component variables are declared for use within the mathematics that determine the model's function. The interface

[4] http://www.cellml.org/public/specification/20010518/units.html #sec_units_cellml_units_dictionary
[5] http://www.bipm.fr/
[6] http://www.cellml.org/public/specification/20010518/units.html#sec_units_user_defined_units
[7] http://www.cellml.org/public/specification/20010518/units.html#sec_units_conversion_between_units_definitions
[8] http://www.cellml.org/public/specification/20010518/units.html#sec_units_equation_dimension_checking

```
<units name="millisecond">
  <unit prefix="milli" units="second" />
</units>

<units name="per_millisecond">
  <unit prefix="milli" units="second" exponent="-1" />
</units>

<units name="millivolt">
  <unit prefix="milli" units="volt" />
</units>

<units name="milliS_per_cm2">
  <unit prefix="milli" units="siemens" />
  <unit prefix="centi" units="metre" exponent="-2" />
</units>

<units name="microF_per_cm2">
  <unit prefix="micro" units="farad" />
  <unit prefix="centi" units="metre" exponent="-2" />
</units>

<units name="microA_per_cm2">
  <unit prefix="micro" units="ampere" />
  <unit prefix="centi" units="metre" exponent="-2" />
</units>
```

FIGURE 4: The units definitions from the CellML description of the Hodgkin-Huxley model.

on each variable declaration determines whether the value of the variable is calculated in the current component and exported, or calculated elsewhere and imported. The last component represents the model's environment. This is a convenient container for variables and constants that are global to the model, and for variables that are not associated with any other component.

In the subsequent sections, the components corresponding to the environment, membrane, sodium channel, and sodium channel activation gate are all examined in detail. The remainder of the components closely resemble one of the components discussed and are not examined here.

5.3.1 The `environment` component

The first **<component>** element defined in the CellML description of the Hodgkin-Huxley model has an identifier of **environment**. The complete definition is given in Figure 5. As the name suggests, this component does not correspond to a physical compartment in a cell, but rather is an abstract container used to define variables whose reach or scope could be thought of as *global*.

```
<component name="environment">
  <variable name="time" public_interface="out" units="millisecond" />
</component>
```

FIGURE 5: The **environment** component represents a convenient abstract container, which is used to define the independent variable **time** separately from the rest of the model.

In the Hodgkin-Huxley model, the only global variable is the independent variable **time**. In CellML models no variables are given any kind of precedence or implicitly assumed to exist. This makes the model definition more robust and model components more re-usable. It is therefore necessary to declare variables to represent time and space if they are needed in a model, but it is not necessary to indicate that a variable like time is an independent variable, as this can be determined from analysis of the differential equations. One could theoretically call the variable that represents time **A1** and use the names **t** and **time** to represent other concepts in your model. It would be unwise to use the names **time** and **t** to represent anything other than time, or the names **x**, **y**, or **z** to represent anything other than space. Doing so would make it more difficult for other model authors to re-use your work.

Note that the name **environment** has no special significance, but is simply a human-readable identifier enabling the modeller to determine the purpose of this component. We expect that software would make use of ontology or metadata information associated with the component to work out that this component would not be rendered in the same way as the **membrane** component, for example.

5.3.2 The `membrane` component

The component at the root of the geometric hierarchy defined within the model is a component with an identifier of **membrane**, the full definition of which is shown in Figure 6.

The **membrane** component declares eight variables. The first three of these, **V**, **E_R** and **I_stim** correspond to the quantities membrane voltage, membrane equilibrium potential, and applied stimulus current, respectively. Their declarations have **public_interface** attributes with a value of `"out"`, indicating that they are defined in this component, that their values may be set and manipulated in this component and that their values are available to other components in the model. **initial_value** attributes are used on each of the **<variable>** elements to set the value of the variable when the independent variables (in this case just **time**) have values of 0.0.

```
<component name="membrane">
  <!-- these variables are defined here and used in other components -->
  <variable
      name="V" public_interface="out"
      initial_value="-75.0" units="millivolt" />
  <variable
      name="E_R" public_interface="out"
      initial_value="-75.0" units="millivolt" />
  <variable
      name="I_stim" public_interface="out"
      initial_value="0.0" units="microA_per_cm2" />

  <!-- these variables are defined here and only used internally -->
  <variable name="C" initial_value="1.0" units="microF_per_cm2" />

  <!-- these variables are imported from other components -->
  <variable name="time" public_interface="in" units="millisecond" />
  <variable name="i_Na" public_interface="in" units="microA_per_cm2" />
  <variable name="i_K" public_interface="in" units="microA_per_cm2" />
  <variable name="i_L" public_interface="in" units="microA_per_cm2" />

  <!--
    The membrane voltage (V) is calculated as an ordinary
    differential equation in terms of the currents.
  -->
  <math xmlns="http://www.w3.org/1998/Math/MathML">
    <apply id="membrane_voltage_diff_eq"><eq />
      <apply><diff />
        <bvar><ci> time </ci></bvar>
        <ci> V </ci>
      </apply>
      <apply><divide />
        <apply><minus />
          <ci> I_stim </ci>
          <apply><plus />
            <ci> i_Na </ci>
            <ci> i_K </ci>
            <ci> i_L </ci>
          </apply>
        </apply>
        <ci> C </ci>
      </apply>
    </apply>
  </math>
</component>
```

FIGURE 6: The definition of the **membrane** component from the CellML description of the Hodgkin-Huxley model.

The variable **C**, which represents membrane capacitance, is declared with no interface, indicating that its value may only be manipulated within the **membrane** component and is not visible to other components.

The final four variables, **time**, **i_Na**, **i_K**, and **i_L** represent time, and the sodium, potassium, and leakage currents through the membrane, respectively. They have **public_interface** attributes with a value of "in", indicating that their value is calculated in another component and imported into this one.

After the variable declarations, a block of MathML[9] defines the single differential equation for **V** shown in Equation (21). This is the inverse of Equation (20) in Section 3.4. The **id** attribute on the first **<apply>** element allows metadata such as comments or annotations to be associated with the equation itself. This is included for demonstration purposes only, as no metadata is associated with any of the equations in this model.

$$\frac{d\,V}{d\,time} = \frac{I\_stim - (i\_Na + i\_K + i\_L)}{C} \tag{21}$$

As **E_R** and **I_stim** can only be changed in this component, the lack of any equations modifying the values of these variables indicates that they are constant with respect to the independent variable **time**.

5.3.3 The sodium_channel component

Following the definition of the **membrane** component is the definition of a component with an identifier of **sodium_channel**. This component is physically inside the **membrane** component, and in turn, contains and encapsulates two components representing the m and h gates within the sodium channel. The full definition of the **sodium_channel** component is given in Figure 7.

Three variables are defined in this component: **i_Na**, **g_Na**, and **E_Na**, which correspond to the sodium current, maximum sodium channel conductance, and reversal potential of the sodium channel, respectively. **i_Na** is available for use by other components (it is imported into the **membrane** component for instance), whereas **g_Na** and **E_Na** are only used internally.

Much of the mathematical functionality of the sodium channel is defined in the encapsulated m and h gate components, so some of the **<variable>** elements in this component define a **private_interface** attribute. The three variables **time**, **V**, and **E_R** correspond to the quantities time, membrane voltage, and membrane reversal potential, respectively. They are each declared with a **public_interface** attribute value of "in", indicating that their value is imported from either a parent or sibling component in the encapsulation hierarchy. (For the proper definition of these terms, see Section 6.2.2 of the CellML specification[10].) The **time** and **V** **<variable>** elements also define a **private_interface** attribute with a value of "out", indicating that their value is available to any encapsulated components. In this sense, the **sodium_channel** component is acting as an interface between the encapsulated components and the rest of the model, which is not concerned with the inner workings of the sodium channel.

Finally, the two variables **m** and **h** correspond to the activation and inactivation coefficients on the sodium channel, respectively. These are declared with a **private_interface** attribute value of "in", indicating that the value of the variable is to be imported from an encapsulated component. The lack of a **public_interface** attribute ensures that these values are not accessible from outside the **sodium_channel** component.

The **sodium_channel** component then defines the two equations depicted in Equation (22) and Equation (23). The first sets the value of **E_Na** (the reversal potential of the channel) in terms of the membrane reversal potential. The second calculates the value of **i_Na** (the sodium current) in terms of the channel conductance, activation and inactivation coefficients, membrane voltage and reversal potential.

$$E\_Na = E\_R + 115.0 \tag{22}$$

[9] http://www.w3.org/Math/

[10] http://www.cellml.org/public/specification/20010518/grouping.html#sec_grouping_bs_encapsulation

```
<component name="sodium_channel">
  <!-- this variable is defined here and used in other components -->
  <variable name="i_Na" public_interface="out" units="microA_per_cm2" />

  <!-- these variables are defined here and only used internally -->
  <variable name="g_Na" initial_value="120.0" units="milliS_per_cm2" />
  <variable name="E_Na" units="millivolt" />

  <!-- these variables are imported from parent and sibling components -->
  <variable
      name="time" public_interface="in"
      private_interface="out" units="millisecond" />
  <variable
      name="V" public_interface="in"
      private_interface="out" units="millivolt" />
  <variable name="E_R" public_interface="in" units="millivolt" />

  <!-- these variables are imported from encapsulated components -->
  <variable name="m" private_interface="in" units="dimensionless" />
  <variable name="h" private_interface="in" units="dimensionless" />

  <math xmlns="http://www.w3.org/1998/Math/MathML">
    <!--
      The following equation determines the reversal potential of the
      sodium channel in terms of the membrane resting potential.
    -->
    <apply id="E_Na_calculation"><eq />
      <ci> E_Na </ci>
      <apply><plus />
        <ci> E_R </ci>
        <cn cellml:units="millivolt"> 115.0 </cn>
      </apply>
    </apply>

    <!--
      The following equation calculates the sodium current in terms
      of the conductance, the membrane voltage, and the gate variables.
    -->
    <apply id="i_Na_calculation"><eq />
      <ci> i_Na </ci>
      <apply><times />
        <ci> g_Na </ci>
        <apply><power />
          <ci> m </ci>
          <cn cellml:units="dimensionless"> 3.0 </cn>
        </apply>
        <ci> h </ci>
        <apply><minus />
          <ci> V </ci>
          <ci> E_Na </ci>
        </apply>
      </apply>
    </apply>
  </math>
</component>
```

FIGURE 7: The definition of the **sodium_channel** component from the CellML description of the Hodgkin-Huxley model.

$$i\_Na = g\_Na \cdot m^3 \cdot h \cdot (V - E\_Na) \tag{23}$$

5.3.4 The `sodium_channel_m_gate` component

The Hodgkin-Huxley model of the sodium channel presented in Section 3.1 consists of two types of gates labelled m and h, which are arranged in series with three m gates and one h gate as indicated in the sodium current calculation in Equation (23). The gates are represented mathematically as coefficients in the 0-1 range, where m is an activation coefficient, meaning that the gate opens as the membrane voltage increases, and h is an inactivation coefficient. The value of each coefficient is determined by its own differential equation, depending on its own value as well as two voltage dependent rate constants. As the sodium current calculation only depends on the values of the activation and inactivation coefficients, the calculation of these coefficients can be conveniently moved into their own components. The **`sodium_channel_m_gate`** component defines the variables and equations pertaining to the m gate activation coefficient. The full component definition is given in Figure 8.

The **`sodium_channel_m_gate`** component defines five variables. The first, **`m`** (the activation coefficient), is declared with a **`public_interface`** attribute value of `"out"`, making it available to any components in the parent and sibling sets in the encapsulation hierarchy. In this case it is the parent **`sodium_channel`** component which makes use of this variable. It also declares an **`initial_value`** attribute value of `"0.05"`, which sets the value of **`m`** to 0.05 at the point where all independent variables (in this case just time) have a value of 0.0.

The **`alpha_m`** and **`beta_m`** variables correspond to the rate coefficients for the m gate and are declared with no interface. Finally the **`V`** and **`time`** variables are declared with a **`public_interface`** attribute value of `"in"` indicating that their value is imported from a component in the parent or sibling set. In this case the parent **`sodium_channel`** component makes those variables available with **`private_interface`** attribute values of `"out"`.

The equations defined in the subsequent MathML block correspond to Equation (7), Equation (9), and Equation (10) from Section 3.1.

5.4 Containment and encapsulation relationships

As discussed in Section 4, the CellML description of the Hodgkin-Huxley model defines both physical and encapsulation relationships between the components in the model. The geometric relationships might be used by CellML processing software as a rendering guideline, and the encapsulation relationships allow processing software to present different levels of model complexity to the user.

Both containment (i.e., physical) and encapsulation relationships between components are defined in CellML using the **`<group>`** element. Each group element may contain one or more **`<relationship_ref>`** elements, each of which references a type of relationship that will apply to all parent-child component pairs defined within the **`<group>`** element. Parent-child pairs are defined by simply referencing the child component in a **`<component_ref>`** element that is defined inside a **`<component_ref>`** element referencing the parent component. The group elements from the CellML description of the Hodgkin-Huxley model are shown in Figure 9.

The first **`<group>`** element specifies the entirety of the containment hierarchy. The **`membrane`** component is the root of the hierarchy, containing the **`sodium_channel`**, **`potassium_channel`**, and **`leakage_current`** components. In turn the **`sodium_channel`** component contains the **`sodium_channel_m_gate`** and **`sodium_channel_h_gate`** components, and the **`potassium_channel`** component contains the **`potassium_channel_n_gate`** component. Only the **`environment`** component is not referenced in the containment hierarchy.

```
<component name="sodium_channel_m_gate">
  <!-- this variable is defined here and used in other components -->
  <variable
      name="m" public_interface="out"
      initial_value="0.05" units="dimensionless" />

  <!-- these variables are defined here and only used internally -->
  <variable name="alpha_m" units="per_millisecond" />
  <variable name="beta_m" units="per_millisecond" />

  <!-- these variables are imported from parent and sibling components -->
  <variable name="V" public_interface="in" units="millivolt" />
  <variable name="time" public_interface="in" units="millisecond" />

  <math xmlns="http://www.w3.org/1998/Math/MathML">
    <!--
      The rate constants on the m and h gates are functions
      of membrane voltage
    -->
    <apply id="alpha_m_calculation"><eq />
      <ci> alpha_m </ci>
      <apply><divide />
        <apply><times />
          <cn cellml:units="per_millisecond"> 0.1 </cn>
          <apply><plus />
            <ci> V </ci>
            <cn cellml:units="millivolt"> 25.0 </cn>
          </apply>
        </apply>
        <apply><minus />
          <apply><exp />
            <apply><times />
              <cn cellml:units="dimensionless"> 0.1 </cn>
              <apply><plus />
                <ci> V </ci>
                <cn cellml:units="millivolt"> 25.0 </cn>
              </apply>
            </apply>
          </apply>
          <cn cellml:units="dimensionless"> 1.0 </cn>
        </apply>
      </apply>
    </apply>

    <apply id="beta_m_calculation"><eq />
      <ci> beta_m </ci>
      <apply><times />
        <cn cellml:units="per_millisecond"> 4.0 </cn>
        <apply><exp />
          <apply><divide />
            <ci> V </ci>
            <cn cellml:units="millivolt"> 18.0 </cn>
          </apply>
        </apply>
      </apply>
    </apply>

    <apply id="dm_dt"><eq />
      <apply><diff />
        <bvar><ci> time </ci></bvar>
        <ci> m </ci>
      </apply>
      <apply><minus />
        <apply><times />
          <ci> alpha_m </ci>
          <apply><minus />
            <cn cellml:units="dimensionless"> 1.0 </cn>
            <ci> m </ci>
          </apply>
        </apply>
        <apply><times />
          <ci> beta_m </ci>
          <ci> m </ci>
        </apply>
      </apply>
    </apply>
  </math>
</component>
```

FIGURE 8: The definition of the **sodium_channel_m_gate** component from the CellML description of the Hodgkin-Huxley model.

```
<group>
  <relationship_ref relationship="containment" />
  <component_ref component="membrane">
    <component_ref component="sodium_channel">
      <component_ref component="sodium_channel_m_gate" />
      <component_ref component="sodium_channel_h_gate" />
    </component_ref>
    <component_ref component="potassium_channel">
      <component_ref component="potassium_channel_n_gate" />
    </component_ref>
    <component_ref component="leakage_current" />
  </component_ref>
</group>

<group>
  <relationship_ref relationship="encapsulation" />
  <component_ref component="sodium_channel">
    <component_ref component="sodium_channel_m_gate" />
    <component_ref component="sodium_channel_h_gate" />
  </component_ref>
  <component_ref component="potassium_channel">
    <component_ref component="potassium_channel_n_gate" />
  </component_ref>
</group>
```

FIGURE 9: The complete set of **<group>** elements used to define containment and encapsulation relationships in the CellML description of the Hodgkin-Huxley model.

The second **<group>** element specifies all of the encapsulation relationships between components in the model. The definition indicates that the **sodium_channel** component encapsulates the **sodium_channel_m_gate** and **sodium_channel_h_gate** components, and the **potassium_channel** component encapsulates the **potassium_channel_n_gate** component. The components that are not referenced in the encapsulation hierarchy are siblings of the **sodium_channel** and **potassium_channel** components, both of which have no parent component.

As for many model design issues in CellML, there's more than one way to do it. It would be possible to define each group of parent-child pairs in a separate **<group>** element. One could also associate both the containment and encapsulation relationships with, for instance, the grouping of the **sodium_channel** component and its children. The form shown in Figure 9 is the recommended best practice because it allows someone looking at the XML code directly to see the entirety of each hierarchy in a single glance, rather than having to hunt through the model definition and piece the hierarchy together. This also makes it easier to validate that a given hierarchy does not contain duplicate component references.

5.5 Passing variable values with connections

The CellML model description concludes with ten **<connection>** elements, which are used to pass the values of the model's variables from one component to another. Each **<connection>** element contains a single **<map_components>** element, which references the components to be connected, and then one or more **<map_variables>** elements, which reference the variables from each component to be mapped together. The **<connection>** element used to pass the values of the **V**, **E_R**, and **i_Na** variables between the **membrane** and **sodium_channel** components is shown in Figure 10.

```
<connection>
  <map_components
        component_1="membrane" component_2="sodium_channel" />
  <map_variables variable_1="V" variable_2="V" />
  <map_variables variable_1="E_R" variable_2="E_R" />
  <map_variables variable_1="i_Na" variable_2="i_Na" />
</connection>
```

FIGURE 10: The **<connection>** element used to pass the values of the **V**, **E_R** and **i_Na** variables between the **membrane** and **sodium_channel** components. This is one of ten **<connection>** elements in the CellML description of the Hodgkin-Huxley model.

The **variable_1** attribute on each **<map_variables>** element references a variable in the component referenced by the **component_1** attribute on the **<map_components>** element. Similarly, the **variable_2** attribute on each **<map_variables>** element references a variable in the component referenced by the **component_2** attribute on the **<map_components>** element. Each variable's value is passed from the component where it is declared with an interface value of `"out"` to the component where it is declared with an interface value of `"in"`, independent of whether a variable's name appears in the **variable_1** or **variable_2** attributes. In the example shown in Figure 10, all of the variables referenced have the same name in both the **membrane** and **sodium_channel** components, but this need not necessarily be the case.

5.6 Model metadata

The CellML description of the Hodgkin-Huxley model also contains some limited metadata, defined within the system described in the metadata framework section of the CellML specification[11], and based on the syntax described in the 18 May 2001 draft of the CellML Metadata specification[12]. This metadata demonstrates a small fraction of the full CellML Metadata data model and has been included for demonstration purposes only. The metadata from the CellML description of the Hodgkin Huxley model is shown in Figure 11, with the comments removed for brevity.

CellML metadata is classified and associated with CellML documents and CellML elements using the Resource Description Framework (RDF)[13]. The RDF specification defines an XML-based syntax for defining metadata. This syntax is suitable for embedding within CellML documents. Within the RDF framework, the CellML Metadata specification recommends using elements from the Dublin Core for classifying various simple types of metadata as described in the XML Encoding of Simple Dublin Core Metadata[14] specification. Personal information is encoded using the syntax proposed in a W3C note entitled Representing vCard Objects in RDF/XML[15]. Citation information is stored using an XML serialization of the Object Management Group's Bibliographic Query Service[16] specification developed for the CellML Metadata specification. Finally, the CellML Metadata specification proposes new elements for several types of metadata for which no XML-based syntax has already been standardised.

In the example in Figure 11, an **<rdf:RDF>** element (an **<RDF>** element in the RDF namespace) encloses the metadata block, which defines metadata for two objects. The **<rdf:RDF>** element declares all of the namespaces that will be used within the element, redefining the CellML Metadata namespace even though it is declared on the parent **<model>** element. This is the recommended best practice as it ensures that the **<rdf:RDF>** element and its children form a self-contained and hence portable block of XML. Metadata is associated with a particular object by placing it inside an **<rdf:Description>** element that defines an **about** attribute. The first **<rdf:Description>** element has an empty **about** attribute, indicating that the enclosed metadata refers to the current document. The second **<rdf:Description>** element has an **about** attribute value of `"#hodgkin_huxley_squid_axon_1952"`. The hash (#) signifies that the subsequent text is the value of an attribute of type ID within the current document, which in this case is the **cmeta:id** attribute on the **<model>** element, which has this value.

The first **<rdf:Description>** element defines some fairly basic information about the CellML document. Firstly, various vCard elements are used to describe a person who is identified as the document author with the Dublin Core **<dc:creator>** element. The subsequent blocks of metadata define the creation date and last modified date for the document. Finally the document's publisher is identified as `The University of Auckland`.

In the second **<rdf:Description>** element, a human readable name for the model is provided using the Dublin Core **<dc:title>** element. A comment on the model is provided using CellML Metadata's **<cmeta:annotation>** element, and the author of the annotation is identified. Finally, a **<bqs:reference>** element is used to associate a citation with the model. The citation is identified by its Pubmed numeric identifier.

[11] http://www.cellml.org/public/specification/20010518/metadata.html
[12] http://www.cellml.org/public/metadata/20010518/index.html
[13] http://www.w3.org/RDF
[14] http://www.dublincore.org/documents/2001/04/11/dcmes-xml/
[15] http://www.w3.org/TR/2001/NOTE-vcard-rdf-20010222
[16] http://www.omg.org/cgi-bin/doc?dtc/01-03-02

```
<rdf:RDF
      xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
      xmlns:cmeta="http://www.cellml.org/metadata/1.0#"
      xmlns:bqs="http://www.cellml.org/bqs/1.0#"
      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="">
      <dc:creator rdf:parseType="Resource">
        <vCard:N rdf:parseType="Resource">
          <vCard:Family>Hedley</vCard:Family>
          <vCard:Given>Warren</vCard:Given>
        </vCard:N>
        <vCard:EMAIL rdf:parseType="Resource">
          <rdf:value>w.hedley@auckland.ac.nz</rdf:value>
          <rdf:type rdf:resource="http://imc.org/vCard/3.0#internet" />
        </vCard:EMAIL>
        <vCard:ORG rdf:parseType="Resource">
          <vCard:Orgname>The University of Auckland</vCard:Orgname>
          <vCard:Orgunit>The Bioengineering Research Group</vCard:Orgunit>
        </vCard:ORG>
      </dc:creator>

      <dc:date rdf:parseType="Resource">
        <dcq:dateScheme>W3C-DTF</dcq:dateScheme>
        <dcq:dateType>created</dcq:dateType>
        <rdf:value>2000-06-01</rdf:value>
      </dc:date>

      <dc:date rdf:parseType="Resource">
        <dcq:dateScheme>W3C-DTF</dcq:dateScheme>
        <dcq:dateType>modified</dcq:dateType>
        <rdf:value>2000-07-26</rdf:value>
      </dc:date>

      <dc:publisher>The University of Auckland</dc:publisher>
    </rdf:Description>

    <rdf:Description about="#hodgkin_huxley_squid_axon_1952">
      <dc:title>The Hodgkin-Huxley Squid Axon Model, 1952</dc:title>

      <cmeta:annotation rdf:parseType="Resource">
        <cmeta:annotation_type>comment</cmeta:annotation_type>
        <rdf:value>
          This is the CellML description of Hodgkin and Huxley's inspirational
          work on a mathematical description of currents through the membrane
          of a nerve fibre (axon) in a giant squid, and their application
          to the modelling of excitation in the nerve. It is generally regarded
          as the first example of a mathematical model of biology.
        </rdf:value>
        <dc:creator rdf:parseType="Resource">
          <vCard:FN>Warren Hedley</vCard:FN>
        </dc:creator>
      </cmeta:annotation>

      <bqs:reference rdf:parseType="Resource">
        <dc:identifier rdf:parseType="Resource">
          <bqs:identifier_scheme>Pubmed</bqs:identifier_scheme>
          <rdf:value>2185861</rdf:value>
        </dc:identifier>
      </bqs:reference>
    </rdf:Description>
  </rdf:RDF>
```

FIGURE 11: The metadata from the CellML description of the Hodgkin Huxley model, with the comments removed.

6 Download This Model

The CellML description of this model is available in a number of formats. If you have your browser set up to view text files served with the "`text/xml`" MIME type, then you can have a look at the XML file here. If not, you can save that file to disk by shift-clicking on the preceding link. A "pretty-printed" browsable HTML version of the XML file is available here — note that you cannot download and save this version for later viewing since it makes use of stylesheets for formatting. If you wish to save or print out the "pretty-printed" version of the XML, a PDF version is also available here. A gzipped tarball (the Unix equivalent of a winzip file) including this documentation, the raw XML and the pretty-printed PDF version of the XML is available here.

Here are those links again:

- hodgkin_huxley_squid_axon_1952.xml[17] — the raw XML.
- hodgkin_huxley_squid_axon_1952.html[18] — an HTML version for browsing online.
- hodgkin_huxley_squid_axon_1952.pdf[19] — a PDF version suitable for printing.
- cellml_hodgkin_huxley_squid_axon_1952.tar.gz[20] — a gzipped tarball with the XML and this documentation.
- hodgkin_huxley_squid_axon_1952_maths.pdf[21] — a PDF of the equations described in the model generated directly from the CellML description using the MathML Renderer[22].

[17] http://www.cellml.org/examples/models/hodgkin_huxley_squid_axon_1952.xml
[18] http://www.cellml.org/examples/models/hodgkin_huxley_squid_axon_1952.html
[19] http://www.cellml.org/examples/models/hodgkin_huxley_squid_axon_1952.pdf
[20] http://www.cellml.org/examples/downloads/cellml_hodgkin_huxley_squid_axon_1952.tar.gz
[21] http://www.cellml.org/examples/maths_pdf/hodgkin_huxley_squid_axon_1952_maths.pdf
[22] http://www.cellml.org/public/tools/index.html