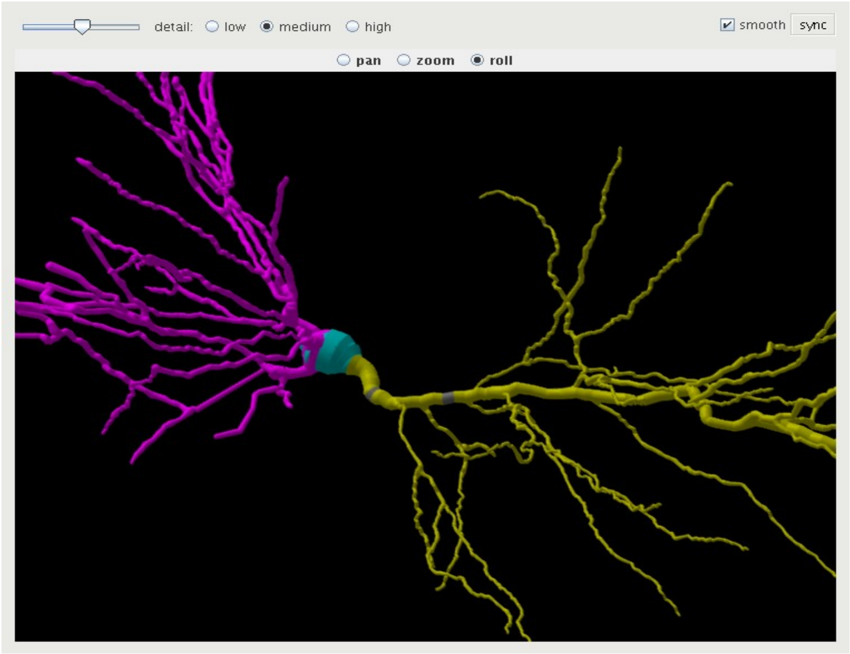
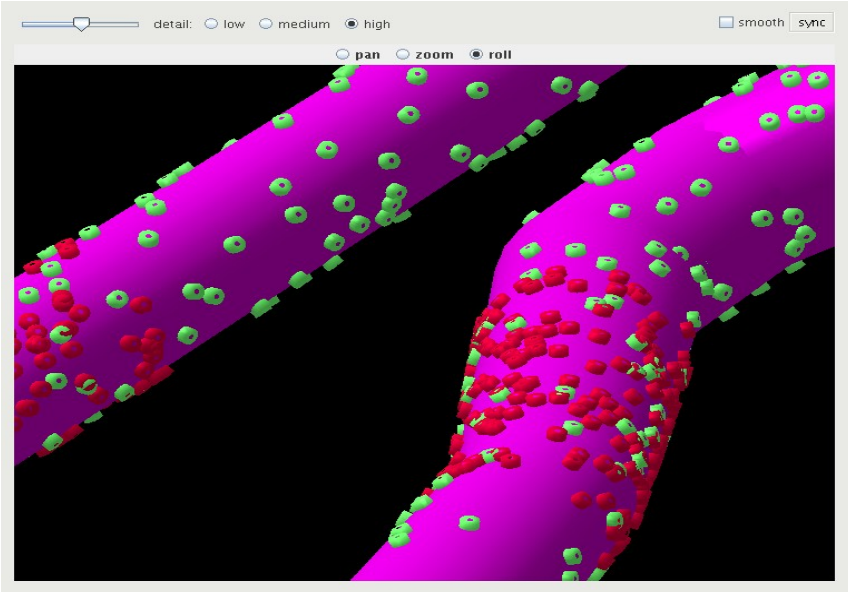
A perspective from Neuroscience



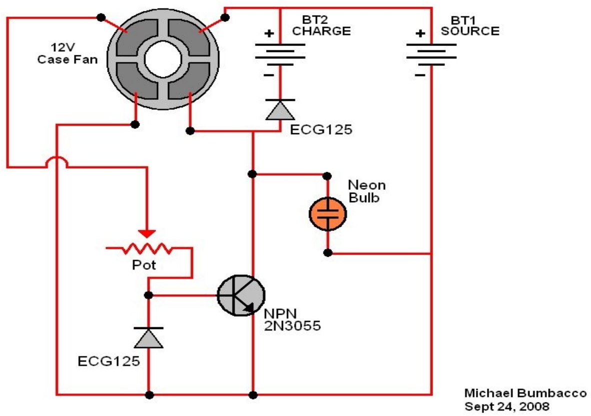


- NeuroML
  - www.neuroml.org (examples)
- NineML
  - www.nineml.org
- *...PSICS*
  - www.psics.org

Physically extended sytems
Tools implement problem-specific methods
Initial focus on what the tools can do, not what the model is

- Works OK(ish) for neurons, ion channels, possibly networks.
- Extension to synapse models risks an explosion of component types and unmanageable specifications.
- → shift to a catalog of user-defined types
- But want to keep the simple top-level specification and avoid duplicating the maths

Electrical Circuit



+

Shared understanding
Not restated every time

Kirchoff's Laws

$I_1 + i_2 + i_3 = 0$

$I_4 - i_3 + i_5 = 0$

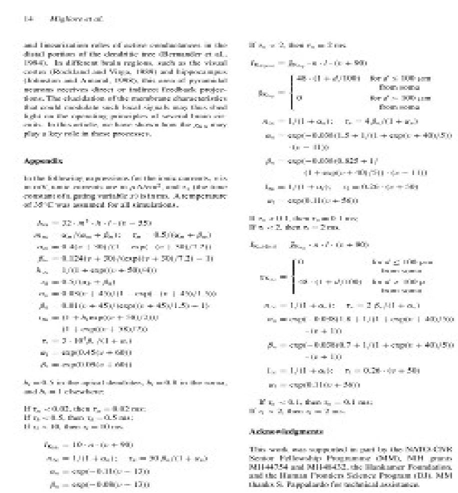
Etc

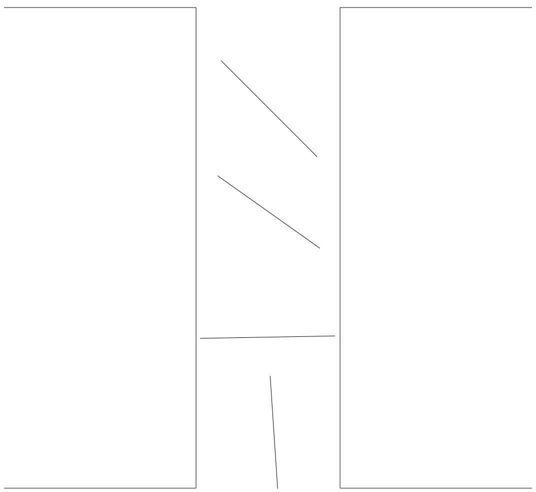
Abstracting Kirchoff's laws as shared knowledge allows a mode to be communicated at the diagram level rather than the equation level.

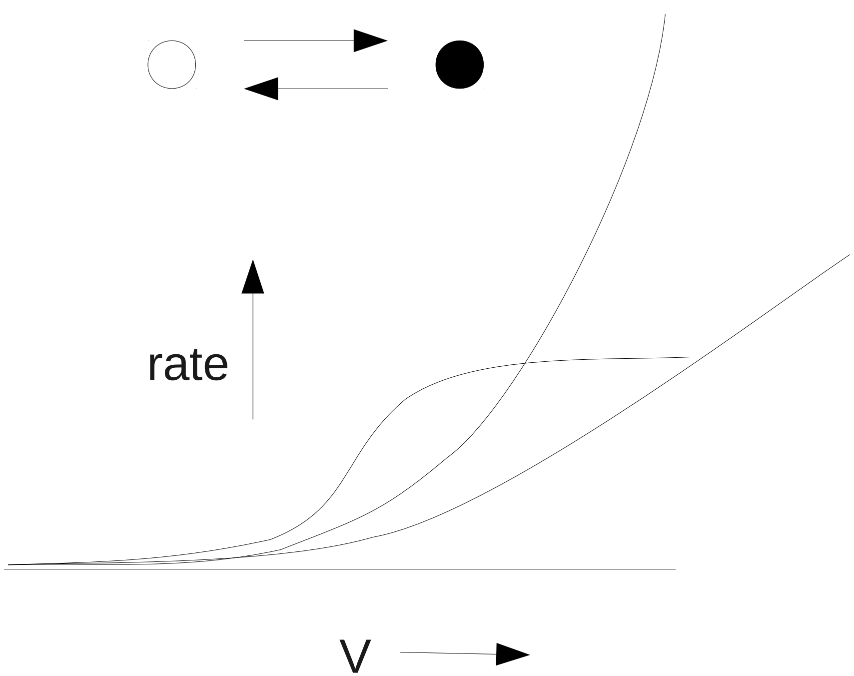
Hodgkin Huxley ion channel model

? + ?

$$I_{Na} = 32 \cdot m^3 \cdot h \cdot i \cdot (v - 55)$$
$$m_\infty = \alpha_m/(\alpha_m + \beta_m); \quad \tau_m = 0.5/(\alpha_m + \beta_m)$$
$$\alpha_m = 0.4(v + 30)/(1 - \exp(-(v + 30)/7.2))$$
$$\beta_m = 0.124(v + 30)/(\exp((v$$
$$h_\infty = 1/(1 + \exp((v + 50)/4$$
$$\tau_h = 0.5/(\alpha_h + \beta_h)$$
$$\alpha_h = 0.03(v + 45)/(1 - \exp$$
$$\beta_h = 0.01(v + 45)/(\exp((v -$$
$$i_\infty = (1 + b_i \exp((v + 58)/2)$$
$$(1 + \exp((v + 58)/2))$$
$$\tau_i = 3 \cdot 10^4 \beta_i / (1 + \alpha_i)$$
$$\alpha_i = \exp(0.45(v + 60))$$
$$\beta_i = \exp(0.09(v + 60))$$







Original HH model and almost all derivatives have:

- Serial independent gates
- Gate opening and closing governed by a rate expression with one of three forms:
  - exp(V)
  - exp(V) / (1+exp(v))
  - V / (1 – exp(-V))
- Each rate expression has three parameters – V scale, Rate scale, V offset

PSICS exploits this structure to allow concise expression of HH style models:

```
<KSChannel id="HH_Na" permeantIon="Na" gSingle="20pS">
        <KSComplex id="m" instances="3">
                <ClosedState id="c"/>
                <OpenState id="o"/>
                <ExpLinearTransition from="c" to="o" rate="1.per_ms" midpoint ="-40.mV" scale="10mV"/>
                <ExpTransition from="o" to = "c" rate="4.per_ms" midpoint="-65.mV" scale="-18mV"/>
        </KSComplex>
        <KSComplex id="h">
                <ClosedState id="c"/>
                <OpenState id="o"/>
                <ExpTransition from="c" to="o" rate="0.07per_ms" midpoint="-65.mV" scale="-20.mV"/>
                <SigmoidTransition from="o" to="c" rate="1per_ms" midpoint="-35mV" scale="10mV"/>
        </KSComplex>
</KSChannel>
```

```
<KSChannel id="HH_K" permeantIon="K" gSingle="20pS">
        <KSComplex id="n" instances="4">
                <ClosedState id="c"/>
                <OpenState id="o"/>
                <ExpLinearTransition from="c" to="o" rate="0.1per_ms" midpoint ="-55.mV" scale="10mV"/>
                <ExpTransition from="o" to = "c" rate="0.125per_ms" midpoint="-65.mV" scale="-80mV"/>
        </KSComplex>
</KSChannel>
```

But this depends on external definitions for the element types.
What if we want to express the whole lot from scratch?

Need a way to express -

- The structures shared by many models, once
- For a particular model, just the parts unique to that model, with a reference to the shared structure

XML shorthand for as yet undefined canonical form

Desired content of the top layer of the model specification

```
<Include file="hhchannel.xml" />
<Unit symbol="mV" dimension="voltage" powTen="-3" />
<Unit symbol="per_ms" dimension="per_time" powTen="3" />
<Unit symbol="pS" dimension="conductance" powTen="-12" />

<HHChannel id="na" conductance="20pS">
  <HHGate id="m" power="3">
    <Forward type="HHExpLinearRate" rate="1.per_ms" midpoint="-40mV" scale="10mV" />
    <Reverse type="HHExpRate" rate="4per_ms" midpoint="-65mV" scale="-18mV" />
  </HHGate>
  <HHGate id="h" power="1">
    <Forward type="HHExpRate" rate="0.07per_ms" midpoint="-65.mV" scale="-20.mV" />
    <Reverse type="HHSigmoidRate" rate="1per_ms" midpoint="-35mV" scale="10mV" />
  </HHGate>
</HHChannel>

<HHChannel id="k" conductance="20pS">
  <HHGate id="n" power="4">
    <Forward type="HHExpLinearRate" rate="0.1per_ms" midpoint="-55mV" scale="10mV" />
    <Reverse type="HHExpRate" rate="0.125per_ms" midpoint="-65mV" scale="-80mV" />
  </HHGate>
</HHChannel>
```

Syntactic fiddles:

“<XXX .../>”
is shorthand for
“<Component type='XXX'/>”

“a='value unit'”
is shorthand for
“<value parameter='a'
size='val'
unit='unit'/>'

```xml
<Dimension name="voltage" m="1" l="2" t="-3" i="-1" />
<Dimension name="time" t="1" />
<Dimension name="per_time" t="-1" />
<Dimension name="conductance" m="-1" l="-2" t="3" i="2" />
<Dimension name="capacitance" m="-1" l="-2" t="4" i="2" />
<Dimension name="current" i="1" />

<Type name="HHRate">
   <Parameter name="rate" dimension="per_time" />
   <Parameter name="midpoint" dimension="voltage" />
   <Parameter name="scale" dimension="voltage" />
   <Behavior>
      <IndependentVariable name="v" dimension="voltage" />
      <DerivedVariable name="r" dimension="per_time" />
   </Behavior>
</Type>

<Type name="HHExpRate" extends="HHRate">
   <Behavior inherit="variables">
      <DerivedVariable name="r" value="rate * exp((v - midpoint)/scale)" />
   </Behavior>
</Type>

<Type name="HHSigmoidRate" extends="HHRate">
   <Behavior inherit="variables">
      <DerivedVariable name="r" value="rate / (1 + exp(0 - (v - midpoint)/scale))" />

   </Behavior>
</Type>

<Type name="HHExpLinearRate" extends="HHRate">
   <Behavior inherit="variables">
      <DerivedVariable name="x" value="(v - midpoint) / scale" />
      <DerivedVariable name="r" value="rate * x / (1 - exp(0 - x))" />
   </Behavior>
</Type>
```

```
<Type name="HHGate">
   <Parameter name="power" dimension="none" />
   <Child name="Forward" type="HHRate" />
   <Child name="Reverse" type="HHRate" />
   <Behavior>
      <IndependentVariable name="v" dimension="voltage" />
      <StateVariable name="q" dimension="none" />
      <ExternalVariable name="rf" dimension="per_time" select="Forward/r" />
      <ExternalVariable name="rr" dimension="per_time" select="Reverse/r" />
      <TimeDerivative variable="q" value="rf * (1 - q) - rr * q" />
      <DerivedVariable name="fcond" dimension="none" value="q^power" />
   </Behavior>
</Type>


<Type name="HHChannel">
   <Parameter name="conductance" dimension="conductance" />
   <Children name="gates" type="HHGate" min="0" max="4" />
   <Behavior>
      <IndependentVariable name="v" dimension="voltage" />
      <ExternalVariable name="gatefeff" dimension="none"
                        select="product(gates[*]/fcond)" />
      <DerivedVariable name="g" value="conductance * gatefeff" />
   </Behavior>
</Type>
```

Status

- Part of NeuroML 2.0 development project. Docs at www.psics.org/lems
- Provides a set of elements and an interpreter for building and running models using user-defined types.
- Early days – mainly experimental at present
- Can retrofit many NeuroML elements so that some models can be run on the generic interpreter
- Some success retrofitting SBML and SED-ML
- The INCF NineML project is aiming to develop a standard for this type of model specification
    - Focus on simple cells and networks but should generalize