

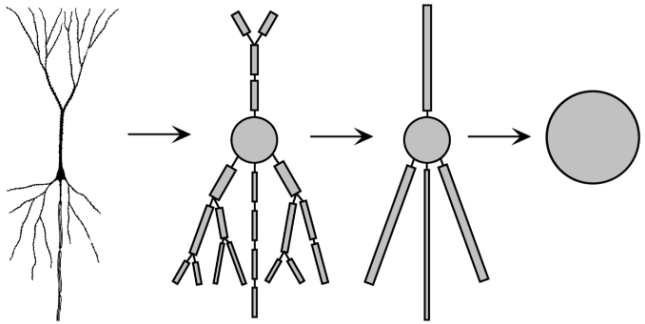


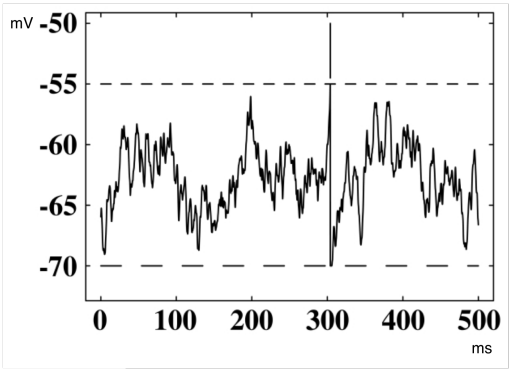

# NESTML

## A Domain Specific Language for Creating Neuron and Synapse Models for the NEST Simulator

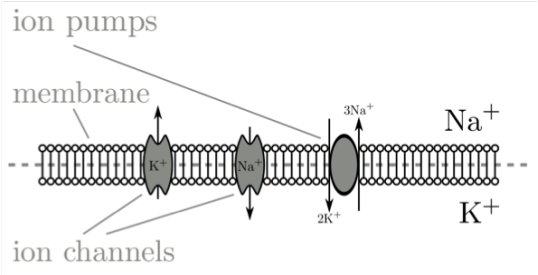
21.-23. September 2015 | Tammo Ippen

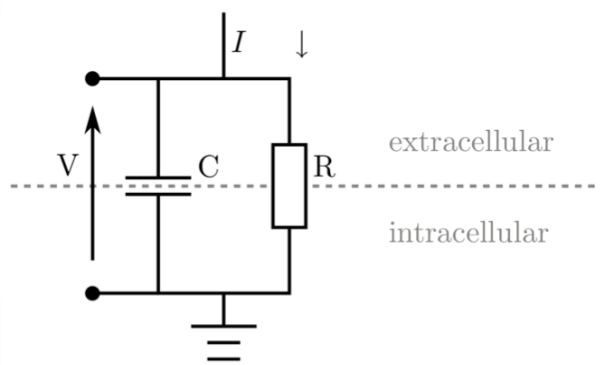
# Models of Neurons in NEST



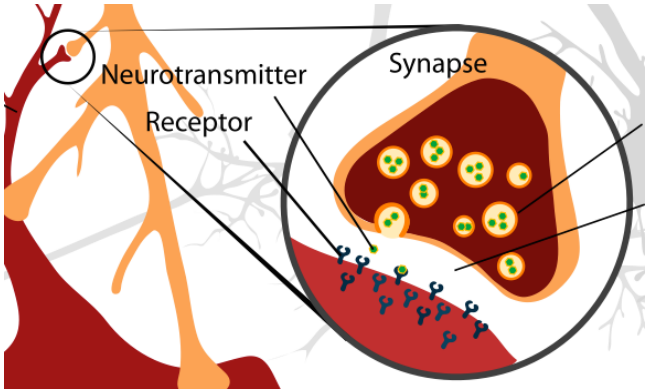


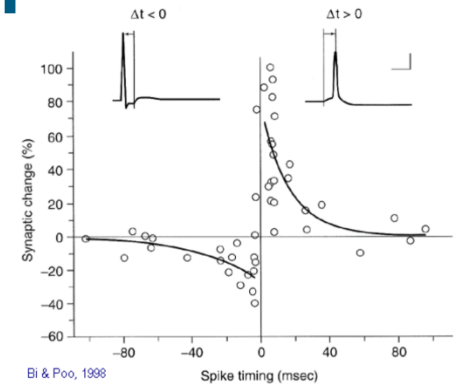

$$\frac{\delta V}{\delta t} = -\frac{V}{\tau_m} + \frac{I_{syn} + I_{ex}}{C_m}$$

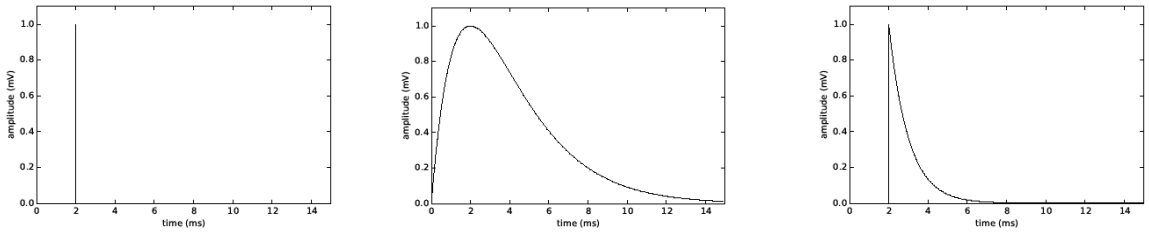


# Models of Synapses in NEST



# Implementing Models in NEST

- C++ classes, which are embedded in the NEST infrastructure
  - Requires deep knowledge in neuroscience, C++ and NEST
  - Error prone, repetitive, not concise
  - Maintenance intensive
- ➢ Developed the NEST Modelling Language (*NESTML*)
  - A domain specific language with clean and concise syntax
  - Describe neuron and synapse models with neuroscience concepts
  - Describe dynamics as a set of ODE or in an imperative way
  - Supports the modeler with syntactic and semantic checks for plausible models

# Example Neuron Model (excerpt)

```
neuron iaf_neuron:

  state:
    # Membrane potential.
    V_m mV
  end

  parameter:
    # Capacity of the membrane
    C_m pF = 250pF [C_m  > 0]
    # Membrane time constant.
    Tau ms = 10 [Tau > 0]
    # Threshold.
    V_th mV = -55
  end

  internal:
    h ms = resolution()
  end

  input:
    in_spikes   <- inhibitory spike
    ex_spikes   <- excitatory spike
    currentBuffer <- current
  end
  # ...
```

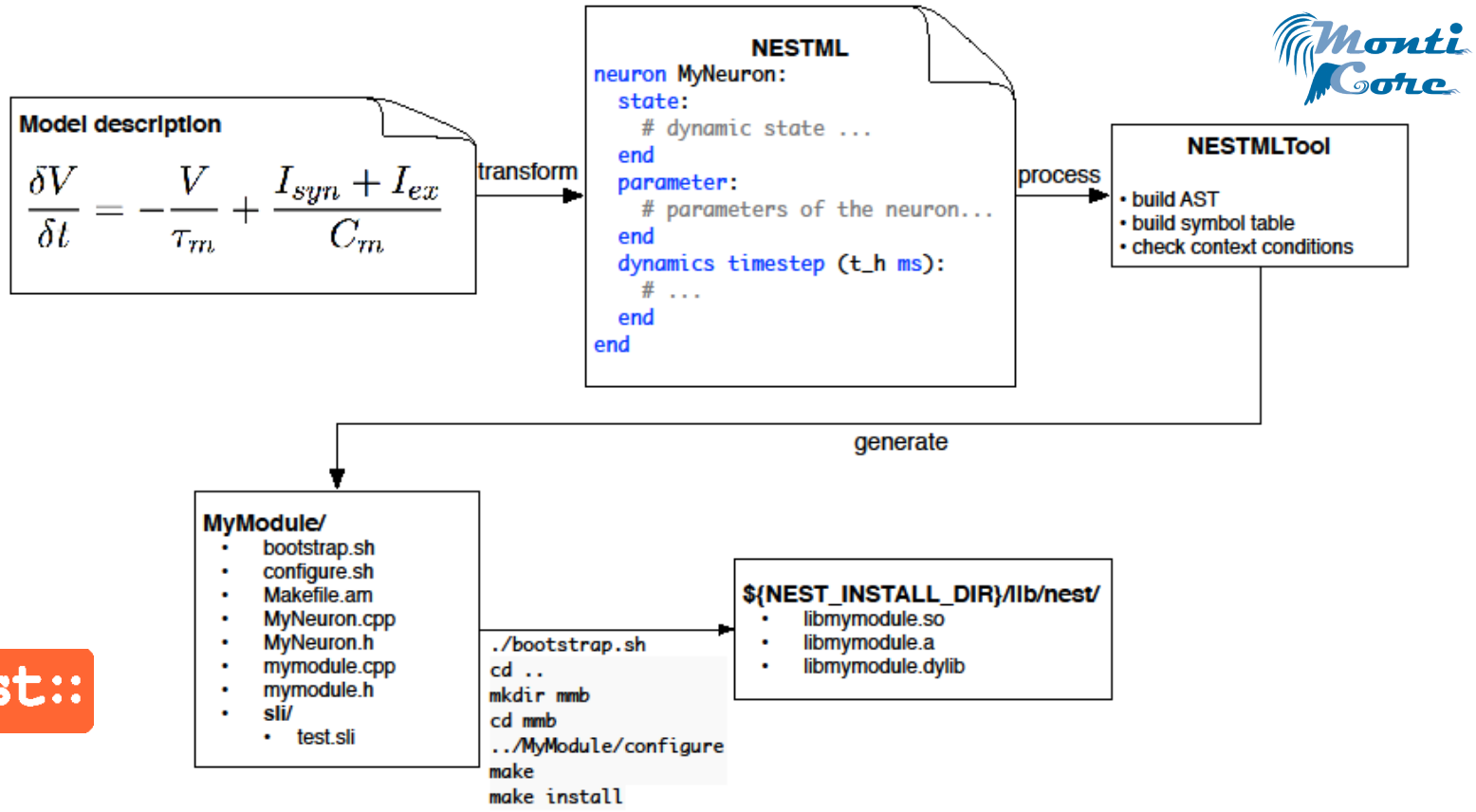
```
  # ...
  dynamics timestep(t ms):

    if r == 0: # not refractory
      ODE:
        I_shape === w * (E/tau_in) * t * exp(-1/tau_in*t)
        d/dt V === -1/Tau * V + 1/C_m * I_shape
      end
    else:
      r = r - 1
    end

    # threshold crossing
    if V_m >= V_th:
      r = RefractoryCounts
      V_m = V_reset

      emitSpike()
    end
  end
end
```

# Conclusion


Model description
$$\frac{\delta V}{\delta t} = -\frac{V}{\tau_m} + \frac{I_{syn} + I_{ex}}{C_m}$$
transform
NESTML
neuron MyNeuron:
state:
# dynamic state ...
end
parameter:
# parameters of the neuron...
end
dynamics timestep (t_h ms):
# ...
end
end
process
NESTMLTool
• build AST
• build symbol table
• check context conditions
generate
MyModule/
• bootstrap.sh
• configure.sh
• Makefile.am
• MyNeuron.cpp
• MyNeuron.h
• mymodule.cpp
• mymodule.h
• sli/
• test.sli
./bootstrap.sh
cd ..
mkdir mmb
cd mmb
../MyModule/configure
make
make install
${NEST_INSTALL_DIR}/lib/nest/
• libmymodule.so
• libmymodule.a
• libmymodule.dylib

# Collaborators


Prof. Dr. M. Diesmann


Prof. Dr. A. Morrison


Prof. Dr. B. Rumpe


Dr. J. M. Eppler


Prof. Dr. H.-E. Plesser


P. M. S. Nazari


T. Ippen


I. Blundell


D. Plotnikov

# Invitation: NESTML Community Workshop

On December 7-9, 2015 in Aachen, Germany.

Information and registration: *Will be announced!*

The workshop is supported by the JARA-HPC Seed *Fund NESTML - A modelling language for spiking neuron and synapse models for NEST and the Hemholtz Portfolio Theme Simulation and Modeling for the Human Brain*.