NEST Desktop

An educational GUI for neuroscience

Sebastian Spreizer
Primary Objectives

- A tool of classroom strength
- No learning programming language required
- Frontend app for the NEST simulator
Client-server concept

<table>
<thead>
<tr>
<th>NEST Desktop</th>
<th>NEST Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>client</td>
<td>local or remoted server</td>
</tr>
<tr>
<td>web interface</td>
<td>API, simulation script</td>
</tr>
<tr>
<td>HTML5</td>
<td>Python (Flask, pyNEST)</td>
</tr>
</tbody>
</table>

Data transfer via GET/POST in JSON format
Data structure

JSON format

Data to NEST Server
- Kernel
- Models
- Collections
- Simulation

Data from NEST Server
- Simulation outcome

```json
"description": "It contains a minimal configuration for simulating two neurons and connecting them with a synapse.

"kernel": {  
  "resolution": 0.1
},  
"models": {  
  "stimulator-0": {  
    "existing": "poisson_generator",  
    "params": {  
      "rate": 6500.0
    }
  },  
  "neuron-1": {  
    "existing": "iaf_psc_alpha",  
    "params": {  
    }
  },  
  "recorder-2": {  
    "existing": "spike_detector",  
    "params": {  
    }
  }
},  
"collections": [{  
  "idx": 0,  
  "element_type": "stimulator",  
  "model": "stimulator-0",  
  "sketch": {  
    "x": 50,  
    "y": 50
  }
},  
{  
  "idx": 1,  
  "element_type": "neuron",  
  "model": "neuron-1",  
  "n": 100,  
  "sketch": {  
    }
},  
{  
  "idx": 2,  
  "element_type": "recorder",  
  "model": "recorder-2",  
  "sketch": {  
    }
},  
"connectomes": [{  
    "idx": 0,  
    "pre": 0,  
    "post": 1,  
    "syn_spec": {  
      "weight": 10.0
    }
},  
{  
    "idx": 1,  
    "pre": 1,  
    "post": 2  
}]
},  
"simulation": {  
  "time": 1000.0
}
```
NEST Desktop

Under the hood

Angular modules

- Navigation (left)
- Controller (right)
- Graph (Sketch, chart)
- Model
- Network
- Results

Design concept
Goals

- Software development activities for NEST Desktop
  - Integration into the HBP infrastructure
  - Implementation of session management, e.g. Docker
  - Integration of VIOLA and the NEST Instrumentation App
  - Integration of in situ pipeline
  - Developer documentation of NEST Desktop
- Integration into the HBP storage infrastructure and collaboration tools
- Usability evaluation in a classroom setting
- Conceptual work on provenance tracking

And keep it in open-source and public for everyone.
This talk would motivate you ...

- to use NEST Desktop for teaching
- to tell others about NEST Desktop
- to build own frontend application