

EDEN: A modular platform for neural simulator research

or, Experiencing Computational Neuroscience as a software researcher

- Sotirios Panagiotou, Mario Negrello, Christos Strydis
- Erasmus University Medical Center, Rotterdam, The Netherlands
- Contact: s.panagiotou@erasmusmc.nl

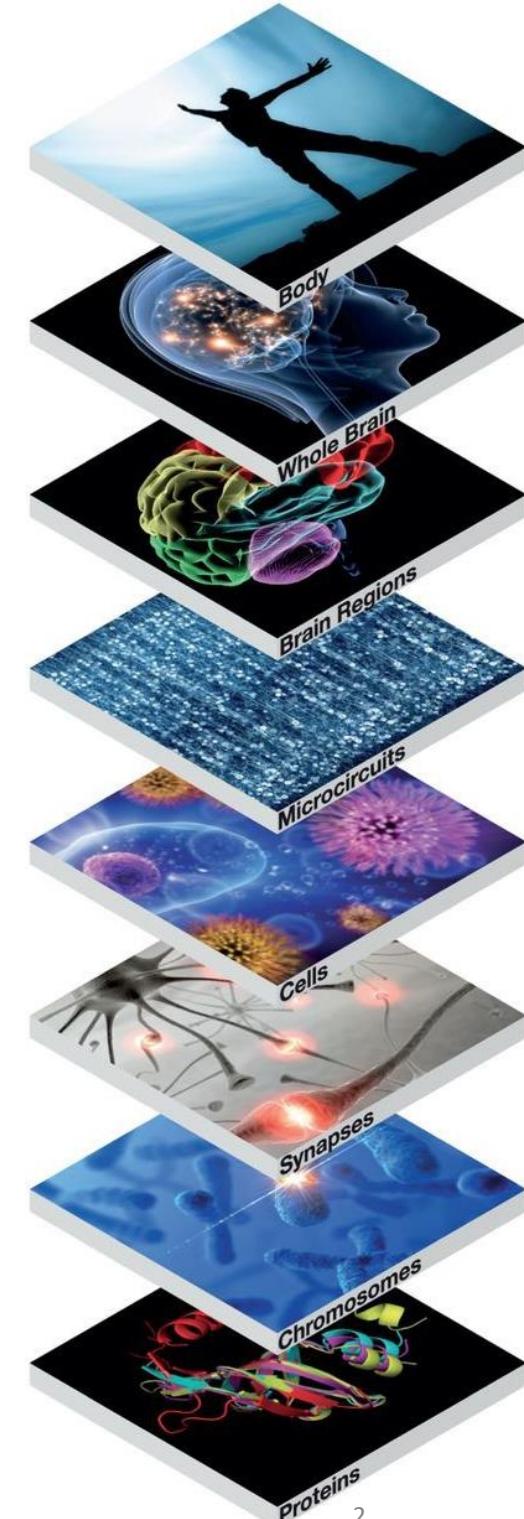


<https://neurocomputinglab.com>



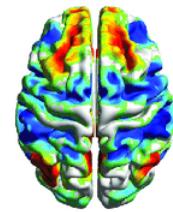
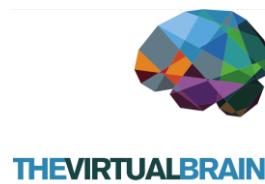
In-silico neuroscience

- We want to do experiments on neurons.
- But neurons are kind of difficult to understand...
- Hence, we think of *models* and propose *hypotheses*. After thinking these through, we set up and do experiments.
- But models become too big to think through!
- Enter... *computer simulation*:
Models are simulated through many scales of:
 - Space (molecules → neurons → brain regions)
 - Time (receptors → action potentials → learning)

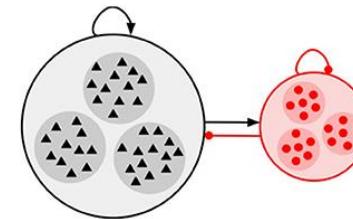


Issues with current neuro-simulators

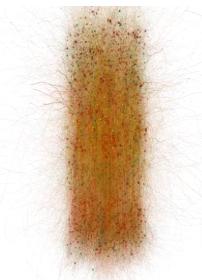
- Traditional simulators are made specifically for *one type of model* each; coverage for other types is patchy at best.



Neural mass modelling



Spiking neural networks

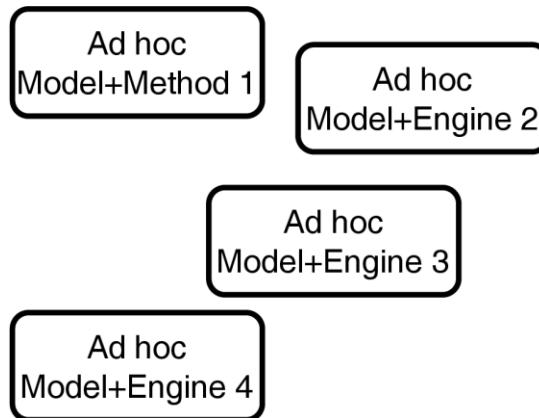


Detailed electrophysiology

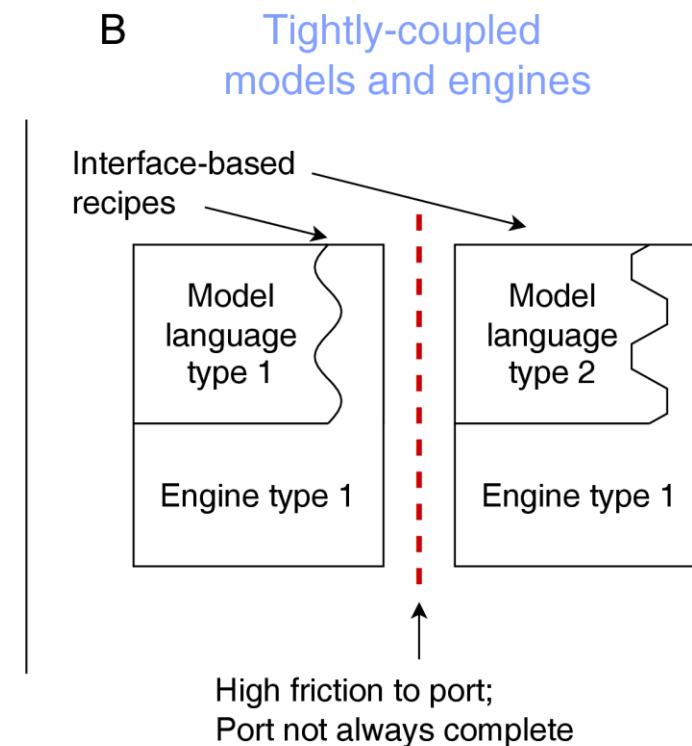
- This leads to several problems:
 - Each simulator has its own *modelling language* ... difficult to translate models.
 - Language* and underlying *engine* are tightly bound ... difficult to improve.
 - Acceleration options are limited by the assumed engine designs!
 - Models may increasingly involve multiple LODs at once ... *very difficult* to interface³ sims!

Neural simulator design approaches

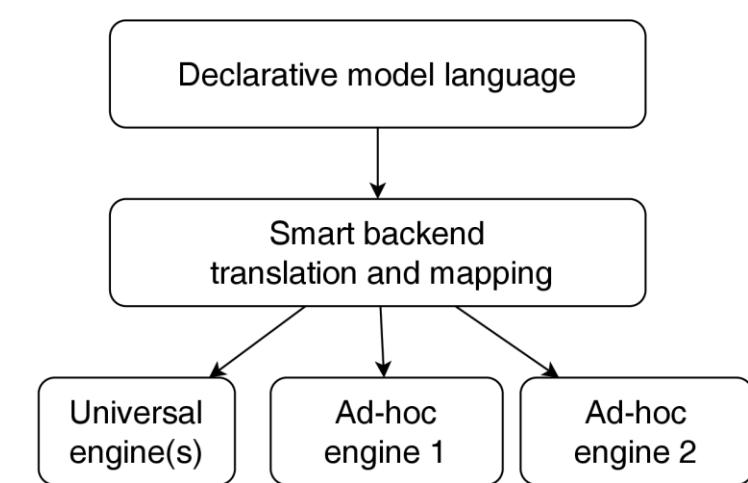
A Ad hoc
model and engine



B Tightly-coupled
models and engines



C Loosely-coupled
models and engines



Porting involves a rewrite

Engine backends access the model description

So I made another neuro-simulator

- EDEN aims for *ease of use, generality* and *simulation speed*.
- It adopts NeuroML, the emerging neuro-modelling standard as input language.
 - This sim-agnostic model format decouples how the model is *described*, from how it is *run*!
- EDEN can then analyse the given model, to decide on how to best run it.
 - Can *switch* between sim algorithms or HW devices, instead of being tied to just one!
 - A compiler-like design, with NeuroML as an intermediate language.
- **Used for Msc and Bsc student projects in our lab since 2020**
 - *User guide:* <https://docs-eden-simulator.org>
- **pip install eden-simulator**

EDEN tech: Overview

- **NeuroML front-end:** User-friendly common interface for model validation & loading.
- **Model-Analysis stage:** The description of the model is analysed to deduce which parts of the model suit which sim algorithms.
- **Simulation back-end:** Auto-specialized, high-performance machine code for each part of the model.

Wide variety of user-provided SNN models



Unified model specification language



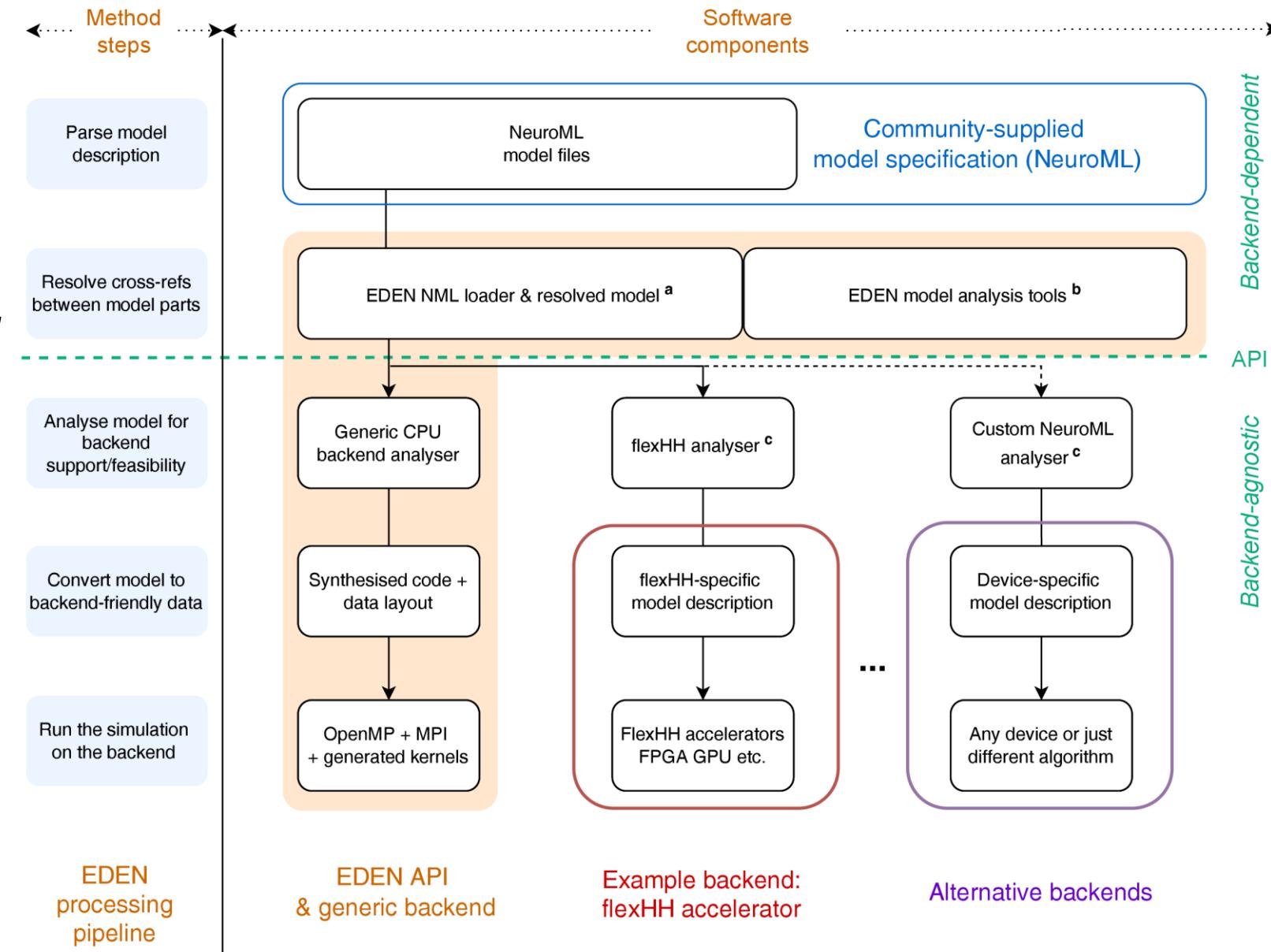
Simulation backend construction kit



Neuro-inspired SNN simulation hardware or code kernel

In further detail

Once a *computational scheme* is designed for a hardware *platform* and *class* of neuron models, it's easy to implement as an EDEN add-on!



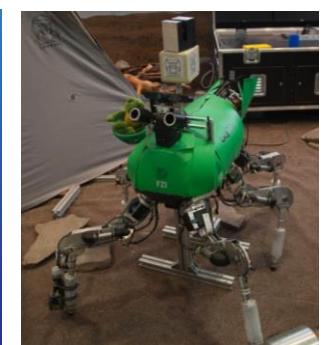
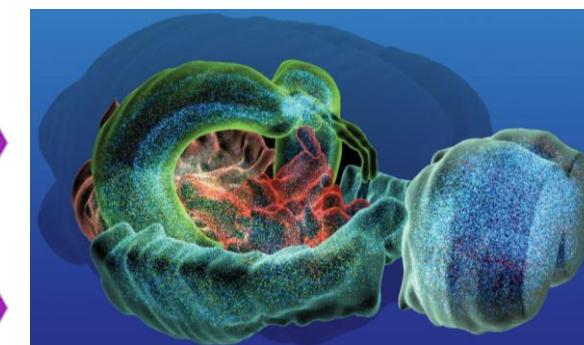
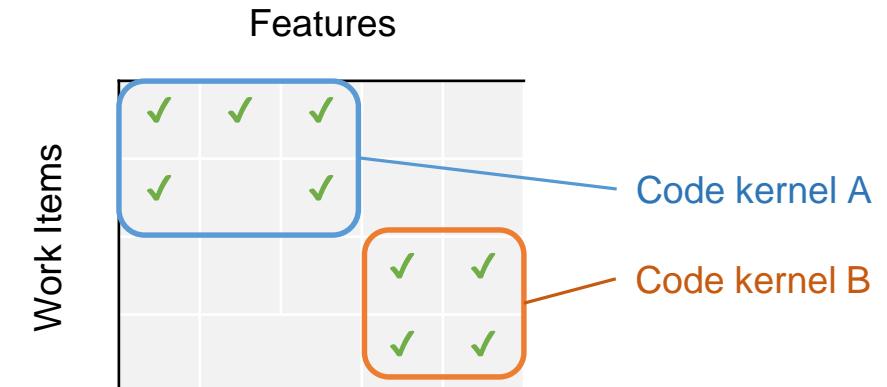
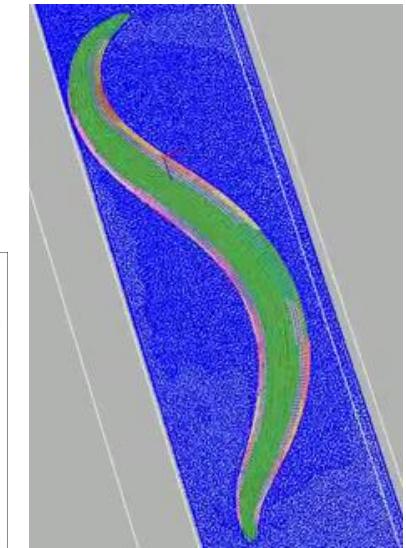
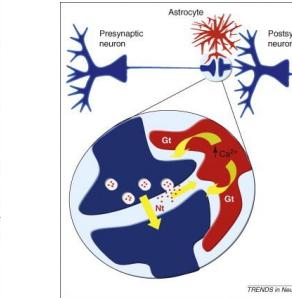
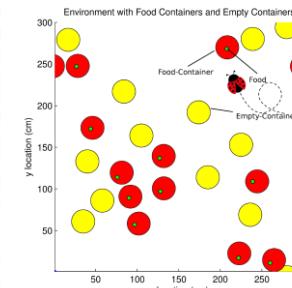
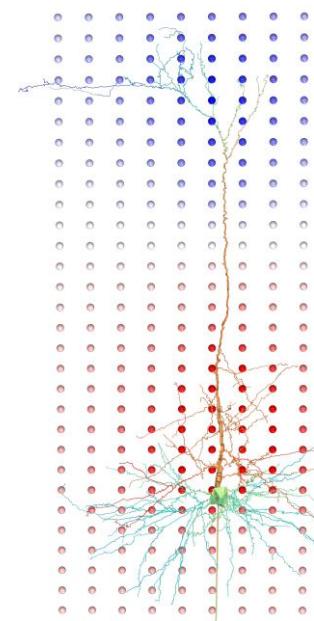
^a All references resolved, for direct processing

^b Facilitating backend-specific analysers

^c Additions for EDEN integration

Next steps

- Enhanced input language features
to cover more use cases
and neuroscience experiments
- Improved workload analysis
to detect more optimisations
and study HPC aspects
- JAX backend
For acceleration, interop
and parameter fitting



Conclusions

Neural sim code / accelerator adoption is limited by model porting

- Decouple model description from sys config through NeuroML
 - Analyse the model to run with the EDEN API
and deploy on specialised backends
-
- 😊 Sim *developers* can use the EDEN framework to
tap the NeuroML corpus and survey user requirements at scale
- 😊 Sim *users* can focus on their experiments, rather than writing numerical codes

Questions?



<http://neurocomputinglab.com>

c.strydis@erasmusmc.nl



<http://eden-simulator.org>

pip **install** eden-simulator

s.panagiotou@erasmusmc.nl

