

POSTER PRESENTATION

Open Access

# Automated code generation from LEMS, the general purpose model specification language underpinning NeuroML2

Boris Marin<sup>1,2\*</sup>, Pdraig Gleeson<sup>1</sup>, Matteo Cantarelli<sup>1,3</sup>, Robert Charles Cannon<sup>4</sup>, Robin Angus Silver<sup>1</sup>

From The Twenty Third Annual Computational Neuroscience Meeting: CNS\*2014  
Québec City, Canada. 26-31 July 2014

Making models more transparent, reproducible and accessible is a major challenge in computational neuroscience. The wide range of modelling tools and strategies available, while convenient from a pragmatic point of view, often make model reuse harder. Efforts on standardization [1], model sharing [2,3] and practices learned from the open-source community [3-5], on the other hand, are proving increasingly invaluable in improving the transparency and reusability of computational models. NeuroML2, the latest version of a model description language for computational neuroscience, is built on LEMS (Low Entropy Model Specification, [6]) a general purpose model specification language. LEMS has a highly structured, hierarchical format, rooted in physical principles, which allows the definition of physical systems including arbitrary networks of nodes modeled as hybrid dynamical systems. There are currently two LEMS interpreters, jLEMS [5] and PyLEMS [6], implemented in Java and Python respectively. As reference implementations, numerical performance and accuracy are not their primary goals. In order to simulate NeuroML2/LEMS models in an accurate, robust and efficient way it is important to be able to perform the numerical integration using well tested scientific simulators, both neuroscience specific, e.g. *NEURON*, and general purpose, e.g. *XPPAUT*, *MATLAB*. Such simulators provide a robust numerical infrastructure important for modelling in general, such as high-order ODE integration and accurate event detection.

We have developed a set of template-based code generators for LEMS whose goal is to transform a LEMS

model into a number of representations compatible with different simulators, using back-end specific templates to produce native code. The translation of the models happens through an intermediate format, developed for this purpose, named dLEMS (distilled LEMS). The dLEMS abstraction is ODE centric for convenience of interoperability with the different target languages. It represents a flattened (e.g. hierarchical channel descriptions are reduced to the core ODEs) and dimensionless version of the LEMS model. The advantage of this approach, compared for instance to a compiler based strategy, is that users willing to provide support for a target simulator can more easily work with dLEMS, as it closely reflects the ODE representations used in general purpose simulators. Users will simply have to create new templates for their target simulator or modify existing ones. Our initial results show that the template-based code-generation approach is feasible, producing runnable simulations in a variety of platforms for single cell models – in particular, templates are available for the following platforms: C code generation (using the LLNL Sundials library for numerical integration), *XPPAUT*, *MATLAB* and *Modelica*. All tools we have developed for this are open source and freely available. We are presently working at extending the dLEMS specification of connections between elements, creating a simplified format for template based mapping to neuronal simulators.

#### Acknowledgements

BM is supported by the Brazilian agency CAPES (2257-13/0). RAS, PG, RCC, MC were supported by the Wellcome Trust (086699). RAS is in receipt of a Wellcome Trust Principal Research Fellowship (095667) and an ERC Advanced Grant (ERC-AG-L55).

\* Correspondence: borismarin@gmail.com

<sup>1</sup>Department of Neuroscience, Physiology and Physiology, University College London, London, UK

Full list of author information is available at the end of the article

#### Authors' details

<sup>1</sup>Department of Neuroscience, Physiology and Physiology, University College London, London, UK. <sup>2</sup>CAPES Foundation, Ministry of Education of Brazil, Brasilia, DF, Brazil. <sup>3</sup>Metacell LLC, San Diego, California, USA. <sup>4</sup>Textensor Limited, Edinburgh, UK.

Published: 21 July 2014

#### References

1. Gleeson P, Crook S, Cannon RC, Hines ML, Billings GO, Farinella M, Morse TM, Davison AP, Ray S, Bhalla US, Barnes SR, Dimitrova YD, Silver RA: **NeuroML: A language for describing data driven models of neurons and networks with a high degree of biological detail.** *PLoS Computational Biology* 2010, **10**(1):e1003376.
2. **ModelDB.** [<https://senselab.med.yale.edu/modeldb/>].
3. **The Open Source Brain project.** [<http://www.opensourcebrain.org>].
4. Gewaltig MO, Cannon R: **Current Practice in Software Development for Computational Neuroscience and How to Improve It.** *PLoS Comput Biol* 2014, **10**(1):e1003376, doi:10.1371/journal.pcbi.1003376.
5. Davison AP: **Collaborative modelling: the future of computational neuroscience?** *Network: Computation in Neural Systems* 2012, **23**:157-166.
6. **LEMS: Low Entropy Model Specification.** [<https://github.com/lems>].

doi:10.1186/1471-2202-15-S1-P45

**Cite this article as:** Marin *et al.*: Automated code generation from LEMS, the general purpose model specification language underpinning NeuroML2. *BMC Neuroscience* 2014 **15**(Suppl 1):P45.

**Submit your next manuscript to BioMed Central  
and take full advantage of:**

- Convenient online submission
- Thorough peer review
- No space constraints or color figure charges
- Immediate publication on acceptance
- Inclusion in PubMed, CAS, Scopus and Google Scholar
- Research which is freely available for redistribution

Submit your manuscript at  
[www.biomedcentral.com/submit](http://www.biomedcentral.com/submit)

