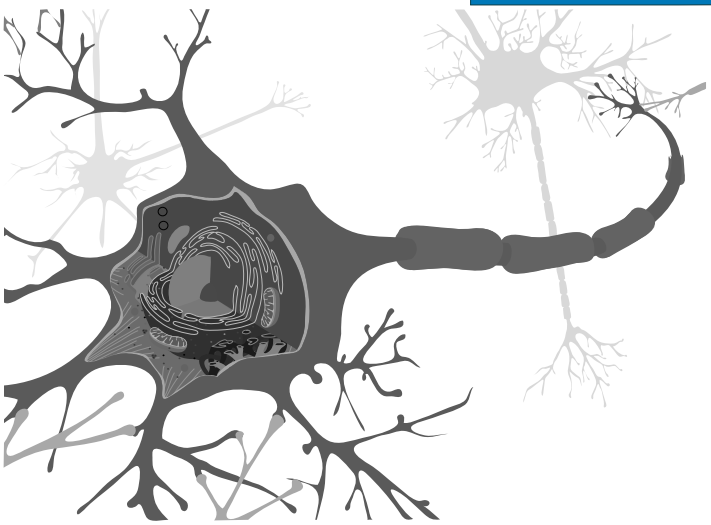
SE | RWTH AACHEN UNIVERSITY

Dimitri Plotnikov

NESTML - die domänenspezifische Sprache für den NEST-Simulator neuronaler Netzwerke im Human Brain Project

nest::ml



Aachener Informatik-Berichte,
Software Engineering Band 33

Hrsg: Prof. Dr. rer. nat. Bernhard Rumpe

NESTML - die domänenspezifische Sprache für den NEST-Simulator neuronaler Netzwerke im Human Brain Project

Von der Fakultät für Mathematik, Informatik und Naturwissenschaften der RWTH Aachen University zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Dimitri Plotnikov, M.Sc.RWTH
aus Simferopol, Ukraine

Berichter: Universitätsprofessor Dr. rer. nat. Bernhard Rumpe
Universitätsprofessor Dr. rer. nat. Abigail Morrison

Tag der mündlichen Prüfung: 07.12.2017

Aachener Informatik-Berichte, Software Engineering

herausgegeben von
Prof. Dr. rer. nat. Bernhard Rumpe
Software Engineering
RWTH Aachen University

Band 33

Dimitri Plotnikov
RWTH Aachen University

NESTML - die domänenspezifische Sprache für den NEST-Simulator neuronaler Netzwerke im Human Brain Project

Shaker Verlag
Aachen 2018

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

Zugl.: D 82 (Diss. RWTH Aachen University, 2017)

Copyright Shaker Verlag 2018
Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany.

ISBN 978-3-8440-5779-9
ISSN 1869-9170

Shaker Verlag GmbH • Postfach 101818 • 52018 Aachen
Telefon: 02407 / 95 96 - 0 • Telefax: 02407 / 95 96 - 9
Internet: www.shaker.de • E-Mail: info@shaker.de

Kurzfassung

Biologische Nervensysteme weisen erstaunliche Komplexität auf. Neurowissenschaftler versuchen, diese Komplexität durch die Modellierung und Simulation der zugrunde liegenden biologischen Prozesse zu erfassen und zu verstehen. Modellkomplexe, die notwendig sind, um die neuromorphen Prozesse abzubilden. Dementsprechend schwierig ist es passende Modelle zu erstellen. Deswegen sind leistungsstarke Werkzeuge notwendig, die es Neurowissenschaftlern ermöglichen, diese Modelle zunächst kompakt, aber auch umfassend auszudrücken und aus diesen Modellen effizienten Code für digitale Simulationen zu generieren.

Domänenspezifische Sprachen erlauben gegenüber General Purpose Programmiersprachen begrenzten und problemorientierten Funktionsumfang an. Die erhöhte Produktivität der Entwickler und verbesserte Qualität der resultierenden Softwaresysteme [VDK98, SEHV12, FHR08] gelten als Vorteile einer DSL. Die erhöhte Produktivität resultiert aus der Tatsache, dass Modelle in einer passend gewählten Notation im Vergleich zu einer äquivalenten Darstellung in einer GPL kompakter sind. Verschiedene Modellierungssprachen für die Computational Neuroscience wurden bereits vorgeschlagen [GCC+10, RCC+11]. Da diese Sprachen jedoch typischerweise Simulatorunabhängigkeit anstreben, unterstützen sie oft nur eine Untermenge der vom Modellierer gewünschten Eigenschaften.

Diese Arbeit präsentiert den Entwurf und die Implementierung der modularen und erweiterbaren domänenspezifischen Sprache NESTML, die Konzepte aus den Neurowissenschaften als vollwertige Sprachkonstrukte zur Verfügung stellt und Neurowissenschaftler so bei der Erstellung von Neuronemodellen für das neuronale Simulationswerkzeug NEST unterstützt.

NESTML wurde mithilfe von MontiCore [GKR+08, Kra10] entwickelt. MontiCore ist eine Language Workbench zur Erstellung von domänenspezifischen Sprachen. MontiCore verwendet und erweitert das Grammatikformat von *ANTLR4* [Par13], das auf dem EBNF-Formalismus [ASU86] basiert, um zusätzliche Konzepte für die Grammatikwiederverwendung. MontiCore stellt eine modulare Infrastruktur für das Parsen von Modellen, den Aufbau der Symboltabllen und zum Prüfen der Kontextbedingungen bereit. Damit können die Entwicklungskosten von NESTML signifikant gesenkt werden.

Abstract

Biological nervous systems exhibit astonishing complexity. Neuroscientists aim to capture this complexity by modeling and simulation of the underlying biological processes. Therefore it is hard to create suitable models which are necessary to depict the neuromorphic processes, which makes it difficult to create these models. Powerful tools are thus needed, which enable neuroscientists to express models in a comprehensive and concise way and generate efficient code for digital simulations.

Domain-specific languages allow limited and problem-oriented functional scope compared to general purpose programming languages. The increased productivity of the developers and improved quality of the resulting software systems [VDK98, SEHV12, FHR08] are regarded as advantages of a DSL. The increased productivity results from the fact that models in a suitably selected notation are more compact in comparison to an equivalent representation in a general purpose language. Several languages for computational neuroscience have been proposed [GCC+10, RCC+11]. However, as these languages often seek simulator independence they typically only support a subset of the features desired by the modeler.

This thesis presents the design and implementation of the modular and extensible domain specific language NESTML, which provides neuroscience domain concepts as first-class language constructs and supports domain experts in creating neuron models for the neural simulation tool NEST.

NESTML is developed using MontiCore [GKR+08, Kra10]. MontiCore is a language workbench for the creation of domain-specific languages. MontiCore expands the grammar format of *ANTLR4* [Par13], which is based on the EBNF formalism [ASU86] with additional concepts for the grammar reuse. MontiCore provides a modular infrastructure for parsing models, building symbol tables and verifying the context-conditions. This allows the development costs of NESTML to be significantly reduced.

Danksagung

An dieser Stelle möchte ich mich bei den lieben Menschen bedanken, die mich während meiner Promotion unterstützt und so zum Erfolg dieser Dissertation beigetragen haben. Mein ganz besonderer Dank gilt meinem Doktorvater Prof. Dr. Bernhard Rumpe für die spannende und lehrreiche Zeit am Lehrstuhl, für die ideenreichen und konstruktiven Diskussionen.

Prof. Dr. Abigail Morrison danke ich herzlich für die Betreuung meiner Arbeit aus der neurowissenschaftlichen Perspektive und die Übernahme des Zweitgutachtens. Ich möchte mich ebenfalls bei Prof. Dr. Klaus Wehrle für die Leitung meines Promotionskomitees und bei Prof. Dr. Thomas Noll für die Durchführung der Theorieprüfung bedanken.

Dr. Jochen Martin Eppler danke ich sehr für das Mentoring sowie für seine fachliche und freundschaftliche Unterstützung während meiner Promotion. Dr. Boris Orth danke ich für die Möglichkeit im Forschungszentrum Jülich arbeiten zu dürfen.

Vielen Dank an Jochen, Guido, Inga, Achim, Katrin, Alexander, Markus und Robert für das Korrekturlesen früher Fassungen der Dissertation.

Aber auch außerhalb der Büroräume mussten Familie und Freunde unter meiner knappen Zeit und meinen wechselhaften Launen leiden. Ich möchte mich daher zunächst entschuldigen und im gleichen Atemzug für die tolle und absolut nicht selbstverständliche Unterstützung bedanken. Durch meine Familie wurde mir nicht nur mein Ausbildungsweg ermöglicht, sondern eben auch die Chance, jederzeit spannende Dinge nebenbei zu erleben.

Ganz besonders möchte ich mich bei meiner Ehefrau Olga bedanken, die mich zu jeder Zeit liebevoll unterstützt und immer wieder motiviert hat, auch wenn das hieß, dass ich viele Samstage, Sonntage, Urlaubs- und Feiertage an der Dissertation gearbeitet und nicht meine Freizeit mit ihr verbracht habe. Dadurch hat sie erheblich zur Fertigstellung dieser Arbeit beigetragen. Meiner Tochter Marie danke ich für ihre freundliche Art, die mich mit einem einzigen Lächeln auch aus dem tiefsten Tal befreien kann.

Mönchengladbach, Dezember 2017
Dimitri Plotnikov

Inhaltsverzeichnis

1 Einleitung **1**
1.1 Motivation und Kontext . . . 1
1.2 Modellierungssprachen . . . 4
1.3 Der NESTML-Ansatz . . . 6
1.4 Wichtigste Resultate . . . 8
1.5 Aufbau der Arbeit . . . 9

2 Mathematische Modellierung von biologischen Neuronen **11**
2.1 Neuronen als Teil des Nervensystems . . . 11
2.2 Mathematische Spezifikation der Neuronendynamik . . . 14
2.3 Charakteristische Neuronendynamiken . . . 16
2.4 Zusammenfassung . . . 21

3 Nutzungsszenarien und Anforderungen für eine Neuronen-Modellierungssprache **23**
3.1 Nutzungsszenarien mit ihren Rollen . . . 23
3.2 Anforderungen an die Neuronenbeschreibungssprache . . . 27
3.3 Verwandte Arbeiten . . . 32
3.3.1 Network Interchange for Neuroscience Modeling Language (NineML) 32
3.3.2 Low Entropy Model Specification . . . 36
3.3.3 NeuroML . . . 39
3.3.4 XML als konkrete Syntax . . . 44
3.3.5 Simulator-spezifische Modellierungssprachen . . . 45
3.4 Evaluierung der existieren Ansätze . . . 49

4 Die MontiCore Language Workbench **53**
4.1 Domänenspezifische Sprachen . . . 53
4.2 MontiCore . . . 55
4.3 Integriertes MontiCore-Grammatikformat . . . 57
4.4 Wiederverwendung der Grammatikdefinition . . . 61
4.5 Symboltabelle . . . 62
4.6 Visitoren zum Traversieren der ASTs . . . 67
4.7 Kontextbedingungen . . . 68
4.8 Codegenerierung . . . 70
4.9 Zusammenfassung . . . 73

5 NESTML: eine domänenspezifische Sprache für die Spezifikation von Punktneuronen 75
5.1 Exemplarisches NESTML-Neuron . . . 75
5.2 NEST Modeling Language . . . 78
5.2.1 Neuronen- und Komponentendefinition . . . 78
5.2.2 Variablenblöcke . . . 80
5.2.3 Ein- und ausgehende Ports . . . 82
5.2.4 Spezifikation von Differenzialgleichungen . . . 84
5.2.5 Spezifikation des Verhaltens . . . 85
5.3 ProceduralDSL: Aktionssprache für die imperative Spezifikation . . . 86
5.4 EquationsDSL für die Beschreibung der Differenzialgleichungen . . . 93
5.5 Ausdruckssprache ExpressionsDSL . . . 94
5.6 UnitsDSL: Sprache für der physikalischen Einheiten . . . 95
5.7 Zusammenfassung . . . 98

6 Umsetzung von NESTML mit der MontiCore Workbench 99
6.1 Grammatiküberblick . . . 99
6.1.1 NESTML-Grammatik . . . 101
6.1.2 Procedural-Grammatik . . . 104
6.1.3 Equations-Grammatik . . . 106
6.1.4 Expressions-Grammatik . . . 106
6.1.5 Units-Grammatik . . . 108
6.2 Symboltabelle von NESTML . . . 108

7 Methodik für die Entwicklung neuer Neuronen mit NESTML 117
7.1 Developing biological neuron models with NESTML . . . 117
7.1.1 Derivation of a mathematical model for biological neurons . . . 117
7.1.2 NEST Modeling Language . . . 120
7.2 Installation and usage of the NESTML environment . . . 124
7.3 Semantic checks of NESTML neurons and components . . . 128
7.3.1 Parse errors . . . 128
7.3.2 Semantic errors in the *ProceduralDSL* . . . 129
7.3.3 Semantic errors in the neuron specification . . . 131
7.3.4 Semantic errors in the equation specification . . . 136

8 Ein Codegenerator für den NEST-Simulator 137
8.1 Generierung des Simulationscodes für den NEST-Simulator . . . 137
8.1.1 Abbildung der Neuronendeklaration . . . 141
8.1.2 Abbildung des Datenzustandes . . . 142
8.1.3 Abbildung von Ports . . . 146
8.1.4 Abbildung von `update`- und `function`-Blöcken . . . 149

8.1.5 Abbildung des `equations`-Blockes 150
8.1.6 Das Analyseframework für Differenzialgleichungen 152
8.1.7 Integration von handgeschriebenem Code 156
8.2 Generierung des Modulintegrationscodes 157
8.3 Konsolen-basierte Schnittstelle für das NESTML-Frontend 159
8.4 Zusammenfassung 160

9 Evaluierung von NESTML 161
9.1 NESTML-Workshops 161
9.2 Validität von der Evaluierung 165

10 Diskussion und Zusammenfassung 169
10.1 Evaluierung der Sprachanforderungen 169
10.2 Zusammenfassung 171

Literaturverzeichnis 175

A Abkürzungsverzeichnis 193

B Tags 195

C Tutorialevaluierung 197
C.1 Fragebogen 197
C.2 Praktische Aufgabe 198

D Grammatiken von NESTML 199
D.1 Units-Grammatik 199
D.2 Commons-Grammatik 200
D.3 Equations-Grammatik 202
D.4 Procedural-Grammatik 203
D.5 NESTML-Grammatik 205

E Modelle für Tutorial 209
E.1 rc_neuron 209
E.2 alpha-shaped postsynaptic responce 210
E.3 shapes 211
E.4 Izhikevich neuron 212

F Curriculum Vitae 213

Tabellenverzeichnis 215

Abbildungsverzeichnis 217