Drug Disease Model Resources

ddmore

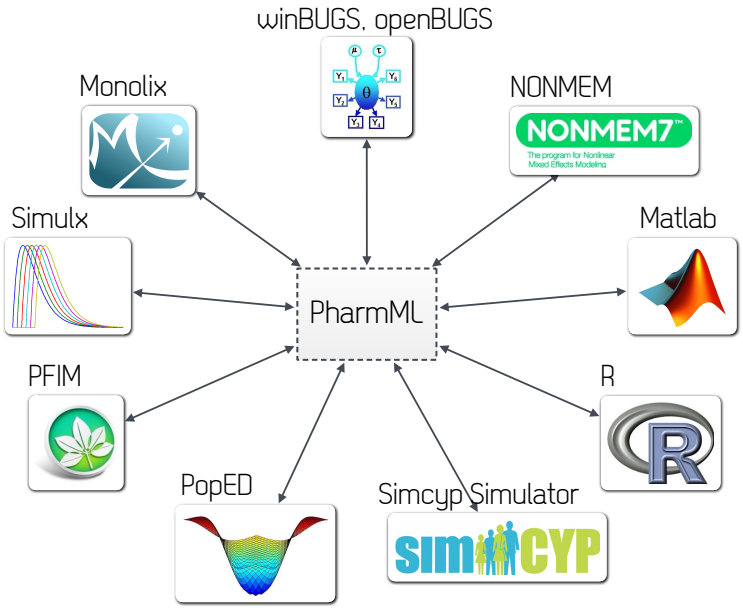
2^{nd} Public Release

Pharmacometrics Markup Language (PharmML)

Language Specification for Version 0.6

Maciej J Swat[1], Sarala Wimalaratne[1], Niels Rode Kristensen[2], Florent Yvon[1], Stuart Moodie[1,3], Nicolas Le Novère[1,4]

[1] *EMBL - European Bioinformatics Institute, Cambridge, UK*,
[2] *Novo Nordisk A/S, Bagsværd, Denmark*,
[3] *Eight Pillars Ltd, Edinburgh, UK*,
[4] *Babraham Institute, Cambridge, UK*



January 23, 2015
1^{st} public version, 0.2.1, was released on November 21, 2013

Contents

Chapter 1

What is PharmML?

1.1 The Problem

The principal problem that PharmML addresses is the reliable exchange of pharmacometric models between software tools. This is illustrated in Figure 1.1, where PharmML is the exchange medium for pharmacometric models for the main modelling and simulation tools in the field. This is not an unreasonable goal and has been successfully realised in the field of Systems Biology.

In Systems Biology such problem does not exist. Software tools exchange models using the Systems Biology Markup Language (SBML; `www.sbml.org`) [13] and many published models can be found in the BioModels Database (`http://www.ebi.ac.uk/biomodels-main/`) [22]. Modellers don't worry about the content of an SBML file; they rely on the fact that when they exchange it between the modelling tools, it just works. Crucial to its success has been an active community of tool developers and modellers who have supported and used it during that time. Equally important has been the provision of sophisticated software libraries (libSBML and JSBML) that take away much of the pain a software tool developer would otherwise experience supporting what is now quite a complex standard. It is a virtuous circle. Users demand their modelling tools support SBML. Developers provide reliable SBML support using libSBML, which enables them to give their users what they want. The more tools that support SBML, the more useful it becomes. The cost of supporting SBML is not negligible but quality libraries like libSBML make the cost acceptable.

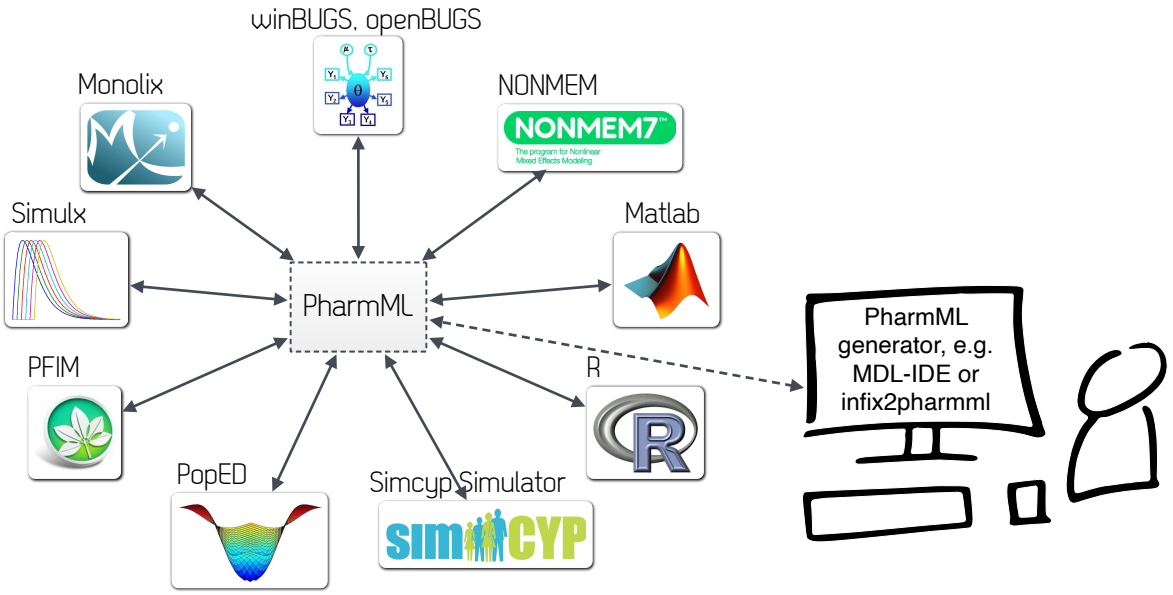


Figure 1.1: Interoperability platform to exchange models via PharmML.

This lesson has not been ignored by the pharmacometrics community and in fact a number of years ago the NLME consortium (a consortium of pharmaceutical companies now all part of DDMoRe) started to work

on a very similar standard to PharmML. This resulted in early drafts of an XML based exchange language, called PharML, but work on it was unfortunately discontinued and the standard has never been used and validated. There are a number of other exchange standards in related modelling fields, which we have drawn on in the development of our work to varying degrees, including:

CellML Supports the exchange and storage of computer based mathematical models of biological systems [27].

NeuroML Supports the exchange and description of models "to describe the biophysics, anatomy and network architecture of neuronal systems at multiple scales"[1] [11].

NineML Describes neuronal networks in a "simulator independent language"[2] that is design to interact with NeuroML [12].

SED-ML Encodes simulation experiments of SBML and CellML models "to ensure exchangeability and reproducibility of simulation experiments"[3] [38].

PharmML is supposed to be the solution in pharmacometrics to the problem. An XML based language that will be able to encode models from NONMEM, MONOLIX, BUGS and related tools. We intend this to be a community standard nucleated around the members of the DDMoRe consortium. In addition we are developing a software library (libPharmML[4]) to help tool providers incorporate support of PharmML and to facilitate its general adoption in the field.

1.2 The Solution

Having described the problem, we here articulate the *kind* of solution we wanted PharmML to be. Developing a language as complex as PharmML is a difficult undertaking and we wanted to make sure that we had some firm principles in place to help when designing the language. We've set these aims and objectives below. PharmML should:

describe the mathematics of a model The language should not include information about the authorship of a model, its update history, or the nature of the disease process or drug that is being modelled. These aspects will be captured by the annotation of the PharmML document and are out of scope of this specification.

describe the task(s) associated with a model The task(s), such as simulation or estimation, to be performed with a model should be encoded in the language.

be declarative The language should describe *what* information is present in a model and *what* the associated task(s) are. It should not describe *how* the information is organised, or *how* the task(s) should be performed.

be platform independent Language elements specific to a particular modelling tool should not be included. For example it should not describe a structural model using a name specific to PREDPP in NONMEM.

serve as an exchange format for the DDMoRe infrastructure The language should either support features required by the infrastructure or provide extension mechanisms so that additional information can be associated with the PharmML document.

provide support for ontological annotation The language should provide a mechanism for it to be annotated with information that is useful to describe the model, but which is beyond the scope of the PharmML document itself.

enable custom extension Provide an extensibility mechanism so that software tools can associate additional, possibly tool specific information, with a PharmML document.

reuse existing standards where appropriate Where an established information standard exists that can be used to represent information within the PharmML document, we should adopt it.

[1] Quoted from `http://www.neuroml.org` on 15 Mar 2013.
[2] Quoted from `http://software.incf.org/software/nineml` on 15 Mar 2013.
[3] Quoted from `http://sed-ml.org` on 15 Mar 2013.
[4] `https://sourceforge.net/projects/libpharmml.ddmore.p`

1.3 PharmML & SO – workflow support and more

PharmML covers the input for a pharmacometric task, i.e. the model, trail design, according task description and experimental data. Another requirements posted for our work-package is to provide additionally a structure to store any type of the numerical results coming from a target tool. This is now an ongoing effort and first draft specification of the so called Standardised Output (SO) is already in use within the framework.

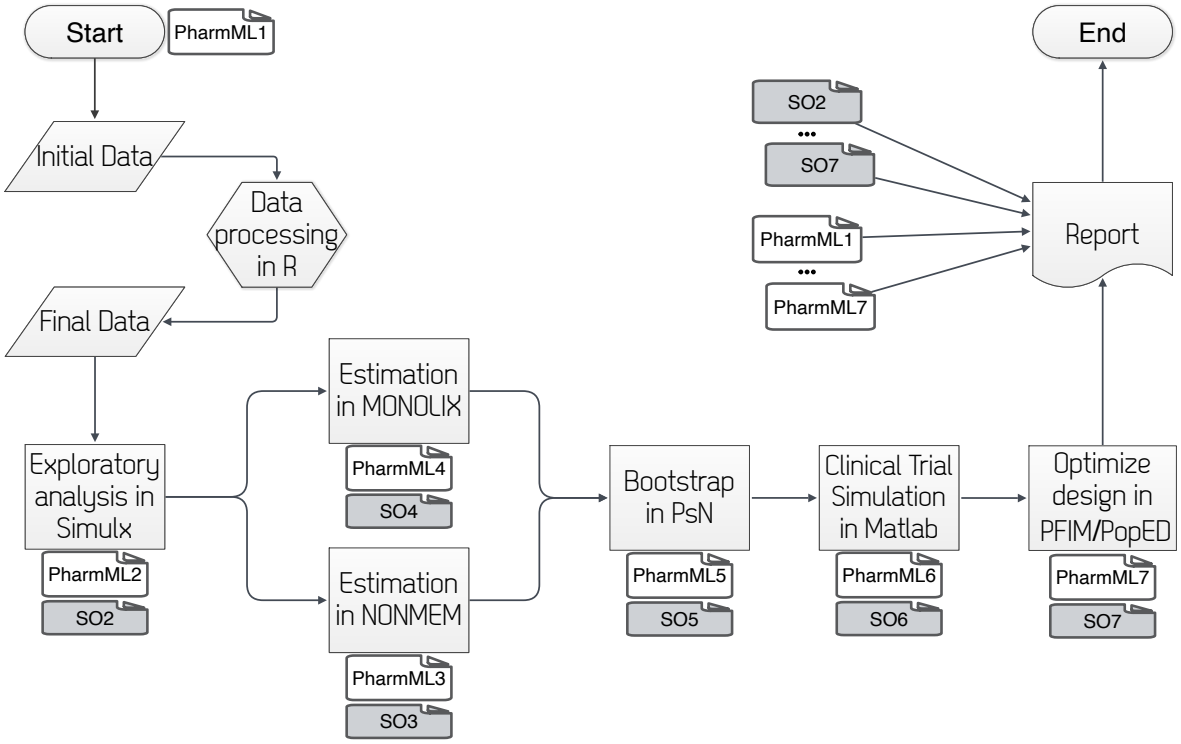


Figure 1.2: PharmML and Standardised Output (SO) supporting a typical workflow in Pharmacometrics featuring major target tools of the DDMoRe platform. Here, it starts with data processing in R, which can consist of data formatting, merging and/or missing-data imputation. After that an explanatory analysis is carried out in Simulx, followed by estimation using either Monolix or NONMEM. Subsequent steps are bootstrapping using PsN, clinical trial simulation in Matlab and finally Optimal Design in either PFIM or PopED. At every step of the workflow, the PharmML model can be stored and the results following each step can be recorded in the corresponding SO file. Documenting workflows in such a detailed way can potentially simplify reporting and ensures reproducibility.

A first public release of SO is planned in few months. Together, PharmML and SO are expected to facilitate:

- Smooth and error-free transmission of models between tools
- Use of complex workflows via standardised model and output definitions, see for an example Figure 1.2
- Easier reporting and bug tracking
- Improved interaction with regulatory agencies regarding modelling and simulation
- Reuse of existing model resources, e.g. BioModels database
- Development of new tools and methods
- Expanding the community developing/applying pharmacometric models.

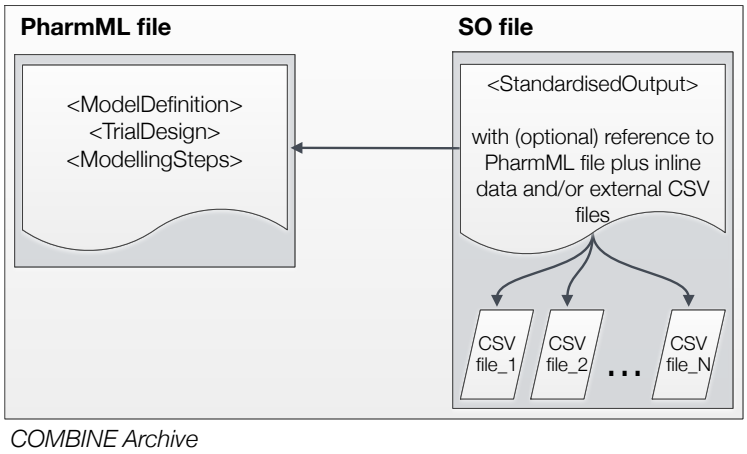


Figure 1.3: PharmML and Standardised Output (SO) relationship.

1.4 Creating PharmML coded models

As indicated in Figure 1.1 once a model is encoded in PharmML it can be shared with any compatible target tool to perform simulation, estimation or optimal design. Modellers will be able in near future to write models using a human readable language also developed within DDMoRe, the Modelling Description Language, MDL (`http://ddmore.eu/mdl`). To facilitate its use, an Integrated Development Environment tool, MDL-IDE, is available, within which the model is automatically translated to PharmML and can be passed to PharmML compatible tools. Development of the MDL and the MDL-IDE is still on-going, but initial results are very promising.

Alternatively, in cases when only the Structural Model is required, modellers can already use the web-editor infix2pharmml (`http://infix2pharmml.sourceforge.net`) to edit complete PharmML models.

1.5 The DDMoRe Consortium

The Drug Disease Model Resources (DDMoRe) consortium aims to promote collaborative drug and disease modelling and simulation research. Its aim is to develop tools and standards that will help the consortium members and later the wider scientific community achieve this goal. Providing PharmML is a key goal of the consortium as it underpins a number of related deliverables of the consortium. In particular:

- The DDMoRe infrastructure in which PharmML is used to exchange models between the different modelling tools.
- The DDMoRe model repository in which PharmML will be used to upload and export models to and from the repository. It will also serve as the storage medium for the repository.
- The DDMoRe library of reference models and data-sets, which will provide models in several therapeutic areas. These models will be encoded using PharmML.

The contribution of the DDMoRe consortium members in guiding and reviewing the standard has been enormous. As the standard evolves their role in using and then promoting the standard to the wider community will be invaluable.

Peer review is important in the development of PharmML and to date we have hosted number of face-to-face meetings since the development of PharmML commenced in August 2011. These meetings were:

- A number of DDMoRe consortium meetings in Leiden and Hoofddrop, in the years 2012–2014.
- The DDMoRe technical workshop hosted by Novo Nordisk in Copenhagen, 28–30 Jan 2013.
- Two PharmML technical workshops hosted by University of Pavia, November 2013 & 2014.

1.6 How PharmML was developed

PharmML was designed and implemented by a relative small group of individuals, but its development has very much been a collaborative process. At the beginning of this project we had a number of development guidelines that we adhered to. We aimed to:

- start with a limited scope and expand the functionality we encode over time.
- drive development using use cases which reflect the current scope.
- test the implemented use cases by generating executable models.
- have frequent review meetings with experts to make sure we are on the right track.
- use existing technology standards if it is possible and reasonable to do so.
- use existing information standards if applicable to avoid re-inventing the wheel.
- make sure the standard is in a form and uses names and terms that make sense to the expert community.

In the first half of 2014 an Interoperability Group within the DDMoRe consortium was appointed to facilitate the testing of PharmML and the interoperability platform driven by it. This typically includes

- Creating a MDL coded model in the MDL-IDE (see Figures 1.1 and 1.2 for an schematic representation of this and the following steps).
- Translation from MDL to PharmML.
- Translation from PharmML to a target tool language, e.g. MLXTRAN coded model for use in Monolix/Simulx or NMTRAN for NONMEM.
- Performing a task in a target tool and the export of results into SO.

Initial results are very promising, with a number of models being successfully processed though this pipeline providing a proof-of-concept for the interoperability concept. Figure 1.4 gives an overview of the development timeline, more details can be find in the detailed changelog in chapter **??**.

1.7 Imperative or Declarative?

In developing PharmML we have designed it to be a declarative language (see section 1.2). While we feel this is the best approach to take for an information exchange langage, it does present us with number of challenges when dealing with NONMEM, the leading tool in the field. Despite having a specifically defined language (NM-TRAN [3]), NONMEM offers a lot of flexibility to the user, and experienced users can make NONMEM do things that it was never designed to do, to a large extent because the imperative approach used in NONMEM facilitates this.

The challenge is to convert from the imperative to the declarative language, because there are many ways to do the same thing in the imperative language. Therefore, generating a PharmML document from a NONMEM control stream is challenging.

1.8 The Evolution of PharmML

This document represents the second public release of PharmML. The first one was released on 21^{st} November, 2013, [26].

Any piece of software is upgraded as users request new features and developers find better ways to do the same thing. A successful standard is no different and with success in mind we expect PharmML to evolve and change further as it is subject to the same influences. To manage this process we have adopted the following strategy to record versions of the PharmML specification.

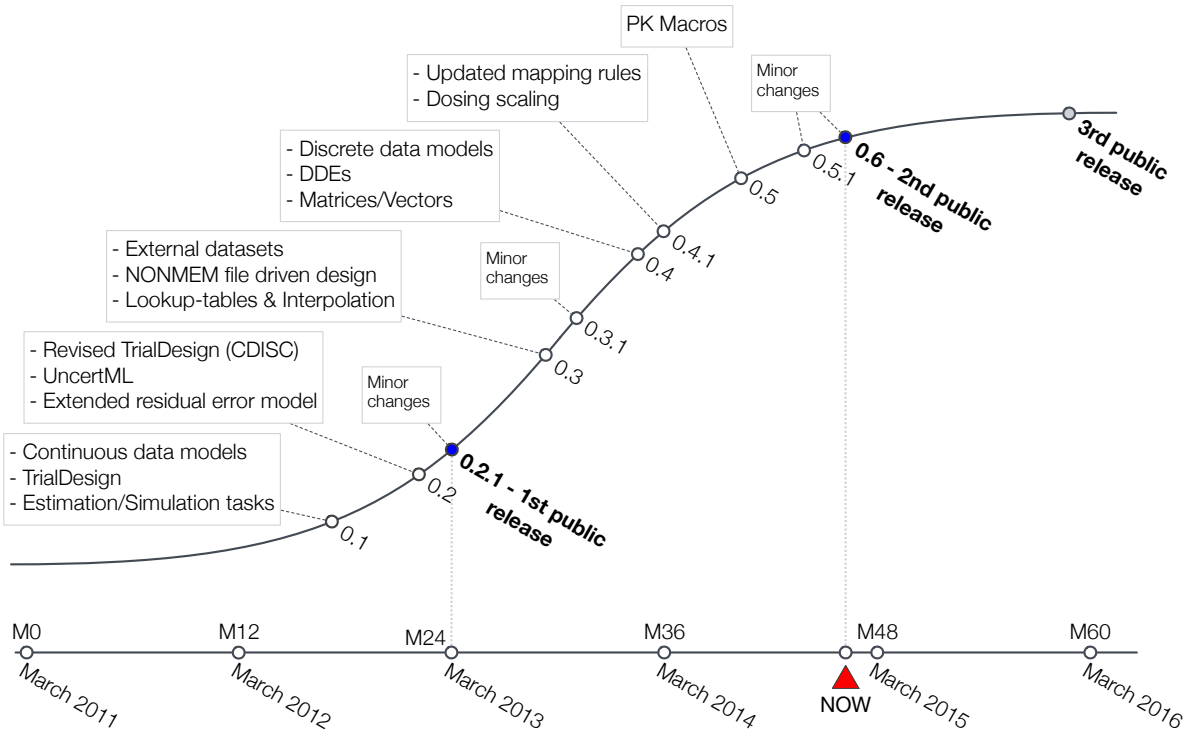


Figure 1.4: Timeline of the PharmML development process, deliverable schedule and version features.

Version number To record changes in the specification we will use the following three level numbering system, of the form $x.y.z$. The levels correspond to the following types of revision:

x **major** Significant new features or radical change of design.

y **minor** New features or evolutionary design changes.

z **patch** Error corrections.

1.9 Feedback

User and developer feedback is important. Typically, this feedback will be in the form of a specific issue: either to report defects identified in the specification or to request new features. Either way the specific issues can be submitted to the tracker at: `https://github.com/pharmml/pharmml-spec/issues`. In some cases it is practical to raise an issue that is broader than a specific issue or requires some discussion within the community. Here the contact is the PharmML forum at `http://pharmml.org`.

Bibliography

[1] Alfred V Aho, Ravi Sethi, and Jeffrey D. Ullman. *Compilers, principles, techniques, and tools*. Addison-Wesley Pub. Co., Reading, Mass., 1986.

[2] S L Beal, Sheiner L B, and Boeckmann A J (Eds.). *NONMEM Users Guides, (1989-2006)*. Icon Development Solutions, Ellicott City, Maryland, USA, 2006.

[3] Stuart L. Beal, Lewis B. Sheiner, Alison J. Boeckmann, and Robert J. Bauer. NONMEM User's Guides. (1989-2009). Technical report, Icon Development Solutions, Ellicott City, MD, USA, June 2009.

[4] Julie Bertrand and France Mentré. Mathematical expressions of the pharmacokinetic and pharmacodynamic models implemented in the Monolix software. Technical report, INSERM U738, Paris Diderot University, September 2008.

[5] Peter L Bonate. *Pharmacokinetic-pharmacodynamic modeling and simulation*. Springer, New York, 2011.

[6] CDISC. The Clinical Data Interchange Standards Consortium; link: www.cdisc.org, September 2013.

[7] CDISC SDM-XML Technical Committee. CDISC Study Design Model in XML (SDM-XML), Version 1.0. Technical report, Clinical Data Interchange Standards Consortium, Inc, 2011.

[8] DDMoRe/Copenhagen meeting. MML/Technical meeting. 2013.

[9] Annette J. Dobson. *An introduction to generalized linear models*. Chapman & Hall/CRC texts in statistical science series. Chapman & Hall/CRC, Boca Raton, 2nd ed edition, 2002.

[10] P Girard, T F Blaschke, H Kastrissios, and L B Sheiner. A markov mixed effect regression model for drug compliance. *Stat Med*, 17(20):2313–33, Oct 1998.

[11] Padraig Gleeson, Sharon Crook, Robert C. Cannon, Michael L. Hines, Guy O. Billings, Matteo Farinella, Thomas M. Morse, Andrew P. Davison, Subhasis Ray, Upinder S. Bhalla, Simon R. Barnes, Yoana D. Dimitrova, and Angus Silver. Neuroml: A language for describing data driven models of neurons and networks with a high degree of biological detail. *PLoS Computational Biology*, 6(6), 2010.

[12] Anatoli Gorchetchnikov, Ivan Raikov, Mike Hull, and Yann Le Franc. Network Interchange for Neuroscience Modeling Language (NineML) Specification Version 0.1. Available via the World Wide Web at `http://software.incf.org/software/nineml/wiki/nineml-specification`, 2010.

[13] M. Hucka, A. Finney, H. M. Sauro, H. Bolouri, J. C. Doyle, H. Kitano, , Sbml Forum the rest of the, A. P. Arkin, B. J. Bornstein, D. Bray, A. Cornish-Bowden, A. A. Cuellar, S. Dronov, E. D. Gilles, M. Ginkel, V. Gor, I. I. Goryanin, W. J. Hedley, T. C. Hodgman, J. H. Hofmeyr, P. J. Hunter, N. S. Juty, J. L. Kasberger, A. Kremling, U. Kummer, N. Le Novère, L. M. Loew, D. Lucio, P. Mendes, E. Minch, E. D. Mjolsness, Y. Nakayama, M. R. Nelson, P. F. Nielsen, T. Sakurada, J. C. Schaff, B. E. Shapiro, T. S. Shimizu, H. D. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner, and J. Wang. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19(4):524–531, 2003.

[14] Michael Hucka, Frank T. Bergmann, Stefan Hoops, Sarah M. Keating, Sven Sahle, James C. Schaff, Lucian P. Smith, and Darren J. Wilkinson. The Systems Biology Markup Language (SBML): Language Specification for Level 3 Version 1 Core. Available via the World Wide Web at `http://sbml.org/Documents/Specifications`, 2010.

[15] Inria POPIX. Mixed Effects Models for the Population Approach - Wiki Popix, 2013.

[16] Ron Keizer and Mats Karlsson. Stochastic models. Technical report, Uppsala Pharmacometrics Research Group, Uppsala, 2011.

[17] Brian W. Kernighan and Dennis M. Ritchie. *The C Programming Language*. Prentice Hall Professional Technical Reference, 2nd edition, 1988.

[18] Marc Lavielle. *Mixed Effects Models for the Population Approach: Models, Tasks, Methods and Tools*. Chapman & Hall/CRC Biostatistics Series, 2014.

[19] Marc Lavielle and Joachim Grevel. WP6.1 Clinical Trial Simulator: Prototype 01. Description of capabilities. Technical report, INRIA Saclay, September 1, 2011.

[20] Marc Lavielle and Lixoft Team. MLXTRAN, A Brief Overview. Technical report, INRIA Saclay & Lixoft, 2012.

[21] Marc Lavielle and Lixoft Team. MLXTRAN, The model coding language for Monolix. Technical report, INRIA Saclay & Lixoft, May 2014.

[22] Chen Li, Marco Donizelli, Nicolas Rodriguez, Harish Dharuri, Lukas Endler, Vijayalakshmi Chelliah, Lu Li, Enuo He, Arnaud Henry, Melanie I. Stefan, Jacky L. Snoep, Michael Hucka, Nicolas Le Novère, and Camille Laibe. BioModels Database: An enhanced, curated and annotated resource for published quantitative kinetic models. *BMC Systems Biology*, 4:92, Jun 2010.

[23] Lixoft. Monolix 4.1.4 - User Guide. Technical report, Lixoft, 2012.

[24] Lixoft. Monolix 4.3 - Model MLXTRAN Tutorial. Technical report, Lixoft, 2014.

[25] Lixoft. Monolix 4.3.2 - User Guide. Technical report, Lixoft, May 2014.

[26] Stuart Moodie, Maciej J Swat, Niels R Kristensen, and N. Le Novère. PharmML: The Pharmacometrics Markup Language, Language Specification Version 0.2.1. November 2013.

[27] P. F. Nielsen and M. D. Halstead. The evolution of CellML. In *Engineering in Medicine and Biology Society, 2004. IEMBS '04. 26th Annual International Conference of the IEEE*, volume 2, pages 5411–5414, 2004.

[28] NLME Consortium. Pharmacometrics Markup Language (PharML), Level 1, Version 1. Technical report, Novartis Pharma AG, Novo Nordisk A/S, Servier and Uppsala University, June 2008.

[29] Terence Parr. *Language implementation patterns: create your own domain-specific and general programming languages*. The pragmatic programmers. Pragmatic Bookshelf, Raleigh, N.C., 2010.

[30] Ines Paule, Pascal Girard, Gilles Freyer, and Michel Tod. Pharmacodynamic models for discrete data. *Clin Pharmacokinet*, 51(12):767–86, Dec 2012.

[31] Elodie L Plan, Alan Maloney, Iñaki F Trocóniz, and Mats O Karlsson. Performance in population models for count data, part i: maximum likelihood approximations. *J Pharmacokinet Pharmacodyn*, 36(4):353–66, Aug 2009.

[32] Benjamin Ribba, Gentian Kaloshi, Mathieu Peyre, Damien Ricard, Vincent Calvez, Michel Tod, Branka Cajavec-Bernard, Ahmed Idbaih, Dimitri Psimaras, Linda Dainese, Johan Pallud, Stéphanie Cartalat-Carel, Jean-Yves Delattre, Jérôme Honnorat, Emmanuel Grenier, and François Ducray. A tumor growth inhibition model for low-grade glioma treated with chemotherapy or radiotherapy. *Clin Cancer Res*, 18(18):5071–80, Sep 2012.

[33] Mike Smith and Nick Holford. Warfarin PK Rosetta Stone. Technical report, Pfizer & Uppsala University, December 10, 2012.

[34] Maciej J Swat, Sarala M. Wimalaratne, Niels R Kristensen, Florent Yvon, Roberto Bizzotto, and Marc Lavielle. PK Macros in PharmML 0.6. Technical report, EMBL-EBI, Hinxton, UK; Novo Nordisk A/S, Bagsværd, DK; Babraham Institute, Babraham, UK, January 2015.

[35] The Data Mining Group (DMG). Predictive model markup language, v4.2.1.

[36] UncertML Team. Uncertainty Markup Language: UncertML Version 3.0, 2014.

[37] W3C. Mathematical Markup Language (MathML) Version 3.0, 2010.

[38] Dagmar Waltemath, Frank T. Bergmann, Richard Adams, and Nicolas Le Novère. Simulation Experiment Description Markup Language (SED-ML): Level 1 Version 1. Available via the World Wide Web at `http://http://sedml.org/specifications.html`, 2011.