35th Annual International Conference of the IEEE EMBS
Osaka, Japan, 3 - 7 July, 2013

Standards and tools supporting collaborative development of the virtual physiological human*

David P. Nickerson, Alan Garny, Poul M. F. Nielsen, and Peter J. Hunter

***Abstract*— The development of a virtual physiological human has an ambitious goal that requires the participation of a large and diverse community of scientists. To be successful in achieving this goal, members of this community must be able to share their work and easily collaborate on new developments and novel applications of existing work. To aid in this, various standardization projects have evolved as part of the Physiome community, as well as supporting computational tools and infrastructure. We present here an overview of the current state of these standardization efforts and key tools that support the collaborative development, integration, and exchange of computational physiology models under the Physiome umbrella.**

I. INTRODUCTION

The Virtual Physiological Human (VPH) initiative is a worldwide effort to develop next-generation computer technologies to integrate all information available for each patient, and generate computer models capable of predicting how the health of that patient will evolve under certain conditions [1]. To achieve this goal requires the participation of a very large and diverse community of scientists from around the globe. In order for such a large and geographically dispersed community to effectively contribute, the initiative requires the use of standardized technologies for the exchange of information so that scientists are able to unambiguously interpret, interrogate, and apply each other's work. Furthermore, accurate tracking of provenance data is required to ensure that scientists are able to judge the suitability of any piece of information, and that appropriate recognition can be distributed accurately. The latter provides an attractive factor for encouraging scientists to contribute in an open and collaborative manner.

The IUPS Physiome Project, from which the VPH initiative grew, has been leading the way in developing standards for encoding models of computational physiology [2]. With growing connections between the computational physiology and computational systems biology communities, the COmputational Modeling in BIology NEtwork (COMBINE) initiative has been established to coordinate the development of the various community standards and formats for computational models (http://co.mbine.org). The primary goal of COMBINE is to develop a set of interoperable and non-overlapping standards covering all aspects of modeling in biology. Through the COMBINE initiative, joint standards are being developed to address requirements of both the computational systems biology and computational physiology communities.

II. STANDARDS

There are a number of standardization projects underway that are covered by the COMBINE initiative (http://co.mbine.org/standards). A comprehensive directory is also available under the BioSharing catalogue of standards (http://biosharing.org/standards_view) [3]. Of the COMBINE standards, CellML, FieldML, and SED-ML are the most relevant to this work, but as the COMBINE initiative develops, we are seeing greater application of existing standards in areas of modeling previously not represented in the user community of those standards. Thus, we are seeing the development of modeling tools and technologies that are becoming agnostic to the specific format in which a model is encoded and, instead, the annotation of the encoded model is becoming the focus of users and tool developers. In this way, users are able to perform their research tasks without needing to understand (or even care) which format their model is encoded in. They use tools that are able to execute *in silico* experiments regardless of the model being encoded in CellML or SBML, for example.

A. CellML

The CellML language is an open standard based on the XML markup language (http://www.cellml.org). The purpose of CellML is to store and exchange computer-based mathematical models, primarily of lumped parameter systems. CellML allows scientists to share models even if they are using different modeling tools. It also enables them

*Research supported by: Virtual Physiological Rat Project, NIH grant [P50-GM094503]; VPH Network of Excellence – Project ID (FP7-ICT-223920); EuHeart; The Maurice Wilkins Centre for Molecular Biodiscovery; RICORDO – Project ID (FP7-ICT-248502).

D. P. Nickerson, P. M. F. Nielsen, & P. J. Hunter are with the Auckland Bioengineering Institute, The University of Auckland, New Zealand (e-mail: [d.nickerson, p.nielsen, p.hunter]@auckland.ac.nz).

A. Garny is with the University of Oxford, England, though he is currently based at Inria Sophia Antipolis, France (e-mail: alan.garny@inria.fr).

978-1-4577-0216-7/13/$26.00 ©2013 IEEE

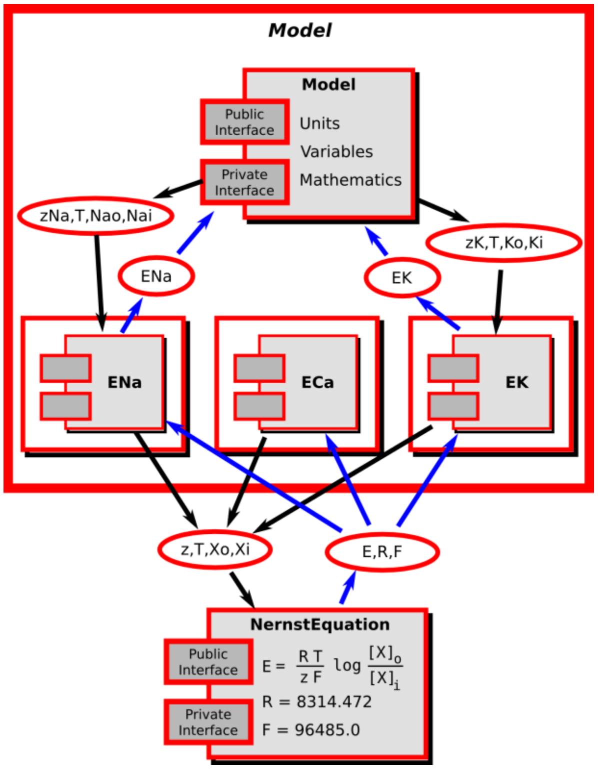

Figure 1. An illustrative example demonstrating modularity and reuse in CellML. This electrophysiology model includes sodium (Na), calcium (Ca), and potassium (K) ions. The NernstEquation component is *imported* three times as ENa, ECa, and EK for calculating the reversal potential of the three ion species. The ovals between the components illustrate the variables being passed between them (the connections between the model and ECa are not shown to aid clarity). Each of ENa, ECa, and EK can be thought of as independent duplicates of the NernstEquation component, requiring the input variables ion valence (**z**), temperature (**T**), and intra- and extracellular ion concentrations (**Xi** and **Xo**, respectively). The reversal potential (**E**), the gas constant (**R**) and Faraday's constant (**F**) are made available for use by other components. As demonstrated by the connections from the model component, the appropriate variables for each ion are used.

to reuse components from one model in another, thus accelerating model development [4].

CellML uses encapsulation and importing mechanisms to enable mathematical models to be encoded in an unambiguous modular form that can be reused. Fig. 1 illustrates this for one of the most common constituents of cellular electrophysiology models, namely the calculation of the Nernst equation for calculating ionic reversal potentials.

The CellML API is an open-source software library that tool developers are able to use to aid in the use of models encoded in CellML in their software projects [5]. Several tools have been developed which use CellML and a listing of some of them is available (http://www.cellml.org/tools).

B. FieldML

FieldML is a declarative language for building hierarchical models represented by generalized mathematical fields (http://www.fieldml.org) [6]. Whereas CellML is designed primarily for encoding lumped parameter models, FieldML can be used to represent the dynamic three-dimensional (3D) geometry and solution fields from computational models of cells, tissues and organs. It will enable model interchange for the bioengineering and general engineering analysis communities.

Similar to CellML, FieldML aims to deliver an exchange format that enables the modular description of models in an unambiguous and reusable manner. While still undergoing significant and fundamental developments, there are several prototype software tools allowing interested members of the community to investigate the concepts underlying FieldML (http://physiomeproject.org/software/fieldml/api). There are also prototype FieldML model descriptions available in the FieldML model repository (http://models.fieldml.org).

C. SED-ML

CellML and other model encoding standards allow modelers to encode their mathematical models in an unambiguous and tool-independent manner. Being able to exchange models is a significant advantage for the VPH community, but often a similar description of how to "apply" the model in a particular situation is required. The Simulation Experiment Description Markup Language (SED-ML) is an XML-based format for encoding simulation descriptions, to ensure exchangeability and reproducibility of simulation experiments [7]. It follows the guidelines established in the Minimum Information About a Simulation Experiment (MIASE) [8]. This allows modelers to not only exchange their models but also complete descriptions of how they use their models to analyze, investigate, or represent the underlying physiology.

The initial release of the SED-ML specification, Level 1 Version 1 (L1V1), was made in March 2011. SED-ML L1V1 was aimed at addressing the most immediate need of the COMBINE community, namely that of being able to exchange simulation experiments involving the evolution of a model over a defined temporal interval. Models are typically encoded in CellML or SBML, but L1V1 is sufficient to enable the use of any XML-based model format. SED-ML L1V2 is currently being developed with the goal of extending this capability to a more modular description of a simulation experiment. This will enable complicated experiment designs (e.g., multi-dimension parameter scans and repeated stochastic simulations) to be encoded as a hierarchical collection of relatively simple simulation steps, thus greatly expanding the scope of simulation experiments able to be encoded in SED-ML.

III. The Physiome Model Repository

The Physiome Model Repository (PMR) is the main online repository for the IUPS Physiome Project (http://models.physiomeproject.org) and is also the parent repository of the CellML (http://models.cellml.org) and FieldML (http://models.fieldml.org) model repositories [9]. It is a publicly accessible and free-to-use web-based repository with a comprehensive user interface and powerful access control features. Users are able to register for repository accounts that enable them to create and share information in the repository. Data are stored in version-controlled *workspaces*. *Exposures* can be made which create permanent Internet addresses for specific revisions of the contents of a workspace. As well as the web-interface for the repository, a comprehensive suite of *web-services* is being developed to enable software tools to directly interact with the repository.

Unlike other online model repositories (*e.g.,* BioModels Database (http://biomodels.net/biomodels/) and ModelDB (http://senselab.med.yale.edu/ModelDB/)), which provide access to static databases of mathematical models and associated data, PMR is designed to support the collaborative development and application of mathematical models in computational studies. The software framework upon which PMR is hosted makes use of a Distributed Version Control System (DVCS) to ensure that provenance is accurately tracked across collaborative developments of any work using PMR.

A. *Workspaces*

The basic unit of information in PMR is the workspace. Currently, each workspace in PMR is a Mercurial repository but the software framework has been developed so that any other VCS could be used. Each workspace is versioned independently, and users are able to control who can access their workspaces with a range of permissions. Workspaces may contain any kind of data, with the only caveat being that a repository curator must approve the workspace prior to it becoming publicly accessible. Hierarchies of workspaces can be assembled via the embedding of one workspace into another, in a manner very similar to the modularity and reusability of CellML itself. As with any data in the workspace, the description of the embedding is versioned. The embedding description specifies exactly which version of a workspace is embedded so the workspace history will include any changes in the versions of any embedded workspaces. Such information is important when curating the workspace contents and is required to ensure the accuracy of provenance tracking.

Due to the origin of the initial contents of PMR, workspaces have historically represented published papers – typically with one CellML model, a diagram of the model structure, and a textural description of the model. Best practice, however, suggests that workspaces are divided into more modular and focused units of work. For example, a user may create a workspace in which to develop a new mathematical model of some physiological mechanism. This workspace can then be embedded into multiple workspaces, each of which instantiates the mathematical model into a specific simulation experiment described in a SED-ML document. The mathematical model could also be embedded into a further workspace that integrates that particular mechanism into a larger system.

B. *Exposures*

As described above, workspaces are each a DVCS repository. In PMR, every revision of every workspace can be accessed via an obscure and lengthy URL (if the user has permission to view that workspace). While powerful, such level of access is rarely required since each workspace may contain a large and verbose history of changes, which is not easy for unfamiliar users to digest [10]. In PMR, users are able to create exposures of a specific revision of a workspace. An exposure provides two important features: a human-friendly URL to the workspace at the specified revision; and the generation of a documentation view of the workspace at that revision for display in the web-interface to PMR. Depending on the type of content being exposed, the user is able to trigger the inclusion of specific plugins used in the generation of the documentation view. Current plugins available include: a CellML plugin for generating a range of static pages specific to CellML models; an interactive 3D viewer for FieldML models; and a static HTML plugin which integrates the versioned HTML from the workspace into the PMR web-interface.

C. *Web-services*

The web-interface has historically been the primary mode of interaction with the content of PMR. With the popularity of mobile computing platforms and the need for application developers to provide direct access to PMR via their software, there is a need for the web-interface functionality of PMR to be available via web-services. In order to provide the full range of functionality and provenance management as the web-interface, access control via web-services has been the primary initial focus of these developments. This has recently been achieved and the provision of web-services has begun. Initial services currently being tested include the creation of workspaces and exposures and the modification of workspace content. These services will address the immediate needs of most tool developers, while further services are being considered.

IV. OpenCOR

OpenCOR is an open-source cross-platform modeling environment for the organization, editing, simulation and analysis of, so far, CellML files (http://www.opencor.ws). It offers both a command line and a graphical user interface (GUI), and relies on a plugin-based approach, making it possible to add new features (*e.g.*, a new way to edit a CellML file, a new numerical solver, support for another standard such as FieldML or SBML).

Among recent developments is a plugin for the annotation of CellML files. A CellML file is represented using a tree-like structure where the leaves are CellML elements. Annotations consist of one or several RDF triples which are listed either raw (*i.e.*, as a subject-predicate-object triple), if they are not recognized, or as a list of qualifier-resource-id triples, if they rely on BioModels.net qualifiers (http://biomodels.net/qualifiers). In the latter case, the user can retrieve information about a given qualifier (*e.g.*, bio:isVersionOf), resource (*e.g.,* uniprot) and id (*e.g.,* Q4KLA0). Existing annotations can be removed and new ones added by specifying both a BioModels.net qualifier and a term. The term can be a resource-id ordered pair (allowing for the direct creation of a new RDF triple) or a string that is used to retrieve possible ontological terms (as a list of name-resource-id triples), as suggested by the semanticSBML web-service (http://semanticsbml.org/semanticSBML/plugin_restapidoc).

An important feature for the community is going to be the support for SED-ML in both the command line and GUI versions of OpenCOR. We anticipate doing this through our CellML plugin, which gives access to the CellML API, including its SProS and SRuS services, which provide the API with SED-ML support. For the GUI, we will likely build on our current 'single cell' simulation plugin, which not only uses our CellML plugin but also handles the current needs of SED-ML.

Other planned efforts include further interaction with PMR via web-services, as well as support for annotating models with information from the RICORDO knowledgebase (http://www.ricordo.eu) [11]. In particular, future versions of OpenCOR will provide guided annotation tools that make use of composite annotations from the RICORDO knowledgebase to enable users to annotate their work with specific biological and physical detail.

V. Discussion

We have shown in Section II that there are established standards in which scientists are able to encode unambiguous and software-agnostic descriptions of their work relevant to the VPH. In Section III, we have described a public repository that enables the collaborative development and application of data and information encoded in these standards. We believe the combination of standards with PMR provides an excellent basis on which to proceed toward achieving the goals of the VPH. Section IV presents a new software environment that exemplifies the usage of these technologies in providing a tool that will allow scientists to make use of them all without needing a detailed understanding of the underlying standards or technologies.

Moving forward, there are two key areas that need further work. Annotating the data contained in PMR with biological, mathematical, and computational information is an ongoing task. Tools like OpenCOR are beginning to make such annotation much easier, but much of the data in PMR is poorly or incorrectly annotated. With over 450 workspaces already in PMR, it will take repository curators a long while to bring the annotation up to a level of completeness on par with the mathematical equations. The goal now is on ensuring tools, documentation, and best practices exist to enable new contributions to PMR and the VPH are annotated to an acceptable level.

The second key area is the association of mathematical models encoded in a standard format with external data. Such external data could be: the experimental data upon which the model was developed; numerical results, simulation tools that curators may use to test the validity of a model; new data to use in parameterizing a mathematical model; *etc*. Association of data with a model happens at the stage when a model is instantiated into a specific simulation experiment, thus current efforts in the SED-ML community are underway to address this requirement.

References

[1] Hunter P, Coveney PV, de Bono B, Diaz V, Fenner J, Frangi AF, Harris P, Hose R, Kohl P, Lawford P, McCormack K, Mendes M, Omholt S, Quarteroni A, Skår J, Tegner J, Randall Thomas S, Tollis I, Tsamardinos I, van Beek JH, Viceconti M. "A vision and strategy for the virtual physiological human in 2010 and beyond." *Philos Transact A Math Phys Eng Sci.* 2010 Jun 13, **368**(1920), pp. 2595-614. doi: 10.1098/rsta.2010.0048.

[2] Hunter PJ. "The IUPS Physiome Project: a framework for computational physiology." *Prog Biophys Mol Biol.* 2004 Jun-Jul, **85**(2-3), pp. 551-69.

[3] Sansone SA and Rocca-Serra P. "On the evolving portfolio of community-standards and data sharing policies: turning challenges into new opportunities." *GigaScience* 2012, **1**:10. doi:10.1186/2047-217X-1-10.

[4] Cuellar, A.A., Lloyd, C.M., Nielsen, P.F., Bullivant, D.P., Nickerson, D.P. and Hunter, P.J. "An overview of CellML 1.1, a biological model description language." *SIMULATION: Transactions of The Society for Modeling and Simulation International*. 2003 Dec, **79**(12), pp. 740-747.

[5] Miller AK, Marsh J, Reeve A, Garny A, Britten R, Halstead M, Cooper J, Nickerson DP, Nielsen PF. "An overview of the CellML API and its implementation." *BMC Bioinformatics*. 2010 Apr 8, **11**:178. doi: 10.1186/1471-2105-11-178.

[6] Christie GR, Nielsen PM, Blackett SA, Bradley CP, Hunter PJ. "FieldML: concepts and implementation." *Philos Transact A Math Phys Eng Sci.* 2009 May 28, **367**(1895), pp. 1869-84. doi: 10.1098/rsta.2009.0025.

[7] Waltemath D, Adams R, Bergmann FT, Hucka M, Kolpakov F, Miller AK, Moraru II, Nickerson D, Sahle S, Snoep JL, Le Novère N. "Reproducible computational biology experiments with SED-ML--the Simulation Experiment Description Markup Language." *BMC Syst Biol.* 2011 Dec 15, **5**:198. doi: 10.1186/1752-0509-5-198.

[8] Waltemath D, Adams R, Beard DA, Bergmann FT, Bhalla US, Britten R, Chelliah V, Cooling MT, Cooper J, Crampin EJ, Garny A, Hoops S, Hucka M, Hunter P, Klipp E, Laibe C, Miller AK, Moraru I, Nickerson D, Nielsen P, Nikolski M, Sahle S, Sauro HM, Schmidt H, Snoep JL, Tolle D, Wolkenhauer O, Le Novère N. "Minimum Information About a Simulation Experiment (MIASE)." *PLoS Comput Biol.* 2011 Apr, **7**(4), e1001122. doi: 10.1371/journal.pcbi.1001122.

[9] Yu T, Lloyd CM, Nickerson DP, Cooling MT, Miller AK, Garny A, Terkildsen JR, Lawson J, Britten RD, Hunter PJ, Nielsen PM. "The Physiome Model Repository 2." *Bioinformatics.* 2011 Mar 1, **27**(5), pp. 743-4. doi: 10.1093/bioinformatics/btq723.

[10] Waltemath D, Henkel R, Hälke R, Scharm M, Wolkenhauer O. "Improving the reuse of computational models through version control." *Bioinformatics.* 2013 Jan 18, **29**(6), pp. 742-748. Doi: 10.1093/bioinformatics/btt018.

[11] de Bono B, Hunter P. "Integrating knowledge representation and quantitative modelling in physiology." *Biotechnol J.* 2012 Aug, **7**(8), pp. 958-72. doi: 10.1002/biot.201100304.