

The Physiome Model Repository 2

Tommy Yu^{1,*}, Catherine M. Lloyd¹, David P. Nickerson¹, Michael T. Cooling¹, Andrew K. Miller¹, Alan Garny², Jonna R. Terkildsen¹, James Lawson¹, Randall D. Britten¹, Peter J. Hunter¹ and Poul M. F. Nielsen¹

¹Auckland Bioengineering Institute, The University of Auckland, Auckland 1010, New Zealand and ²Department of Physiology, Anatomy & Genetics, South Parks Road, Oxford OX1 3PT, UK

Associate Editor: Alfonso Valencia

ABSTRACT

Motivation: The Physiome Model Repository 2 (PMR2) software was created as part of the IUPS Physiome Project (Hunter and Borg, 2003), and today it serves as the foundation for the CellML model repository. Key advantages brought to the end user by PMR2 include: facilities for model exchange, enhanced collaboration and a detailed change history for each model.

Availability: PMR2 is available under an open source license at <http://www.cellml.org/tools/pmr/>; a fully functional instance of this software can be accessed at <http://models.physiomeproject.org/>.

Contact: tommy.yu@auckland.ac.nz

Received on September 21, 2010; revised on December 1, 2010; accepted on December 23, 2010

1 INTRODUCTION

Increases in the computational resources available to modellers have progressively allowed more detailed models to be created and simulated. The traditional approach of publishing a human-readable description of a model in an academic journal, and expecting reviewers and readers to implement the model from the manuscript alone, is becoming less feasible. It is therefore important for modellers to share machine-readable descriptions of models. This has led to the development of model description languages, such as CellML (Cuellar *et al.*, 2003) and SBML (Hucka *et al.*, 2003), standardized model processing libraries such as the CellML API (Miller *et al.*, 2010) and libSBML (Bornstein *et al.*, 2008), and model repositories such as the CellML model repository (Lloyd *et al.*, 2008), BioModels Database (Li *et al.*, 2010) and JWS online (Olivier *et al.*, 2004).

From its creation in 2000 until June 2009, the CellML model repository represented a relatively simple storage facility for CellML model files. Although all the models in the repository were freely accessible to the public, there was no mechanism in place to promote the sharing of models with their revision history intact. Furthermore, collaboration between several researchers is often required during the development of a model. Collaboration on a model can be greatly simplified by a tool which records the change history of a model, and makes that history available to other collaborators. Finally, once a model is ready to be published, it must be indexed and presented so that it can be easily searched for and viewed, thereby avoiding unnecessary duplication of modelling efforts. Since it was not

possible to extend the original repository software to support these more advanced usage requirements, a complete rewrite was required, and the Physiome Model Repository 2 (PMR2) was created. Here, we introduce PMR2 as a solution to the above concerns, and discuss the advantages it provides to the modelling community.

2 IMPLEMENTATION

Unlike the previous version of the CellML model repository, PMR2 is file format agnostic, meaning that it has no restrictions on what it can store. For example, SBML¹ and FieldML (Christie *et al.*, 2009) models can be stored alongside models in CellML and other formats. However, in order to properly handle (display, edit, etc.) these, there is a need to implement plugins for each different format. In addition to the model files, it is possible to upload additional supplementary data with a model, including images, software-specific information and even the original experimental data on which the model is based.

2.1 Workspaces, distributed version control systems and model revision history

All the files related to a particular model are stored together in PMR2 within a defined ‘workspace’, which can be regarded as a folder. The implementation of workspaces uses a distributed version control system (DVCS) called Mercurial (O’Sullivan, 2007). By providing version tracking, this system ensures that users within a group cannot accidentally overwrite or purge changes of other users. Furthermore, each change made to a model, or its associated files, is recorded as a single changeset: a time-stamped, informative comment from an identifiable user, which describes the changes they have made. As model files are progressively altered, the changesets preserve the history of model development. Finally, even if two users simultaneously change the same model file, distinct changesets allow their work to be later merged in a controlled manner.

Another advantage associated with using a DVCS is that it allows users to collaborate directly with each other, independent of a centralized online repository, as collaborators of any particular model will have a complete clone of the workspace with the model and related files they are working on. This allows each individual collaborator to work and commit changes to their local copy, creating new changesets which may then be shared between themselves, or pushed to a centralized repository when their work is ready to be reviewed or released.

¹For an example of an SBML model in the repository please see: <http://models.cellml.org/exposure/e24887f982>.

*To whom correspondence should be addressed.

2.2 CellML 1.1 and embedded workspaces

The previous version of the model repository software could only handle CellML 1.0 models. By introducing the concept of workspaces, PMR2 has enabled the storage of CellML 1.1 models: models that use the 'import' feature can reference components and units from other existing models. PMR2 version 0.2 has developed this concept further by including support for embedded workspaces, in which one or more workspaces can be included, by reference, into another workspace. Embedded workspaces are intended to manage the separation of the core model from its subcomponents, and thereby facilitate the sharing and reuse of model components independently from the source model. As this separation enables the development of the subcomponents to proceed independently of the main model, the version of the workspaces embedded is also tracked. Changes made to a workspace which has been embedded in another workspace will not affect the embedding workspace until the author explicitly chooses to update the version to use. This gives the author the opportunity to review the changes to the embedded workspace, and ensure that they won't adversely affect the models in the embedding workspace. Finally, embedded workspaces enable the import of components via relative URLs, thus promoting modular model development.

2.3 Model exposures

In addition to the concept of a workspace as a folder, the other main feature of PMR2 is the presentation view of the model and any associated data which may be within the workspace. As the contents of the workspace at the revision corresponding to any changeset are immutable, one can select a single changeset and create an 'exposure' from it. Presently, creating an exposure leads to both the generation of the presentation view of the contents of a workspace as at the selected revision, and to the exposure's URL being included in the main repository category listings. Currently, a curator may annotate their assessment of the quality of the coded version of the model using 'curation stars'.

PMR2 has been designed to be extensible to enable the support of the range of presentation styles required for the different file types. This requires a system where plug-ins can be installed with ease onto an instance of PMR2; we have implemented and built this system on the Zope Component Architecture (Baiju, 2007), which makes use of the adapter pattern (Gamma et al., 1995) to register and activate plugins based on their names. This system allows software developers to construct specific plugins, to generate the presentation styles required for any particular model type, which can be installed and enabled on an instance of PMR2. This enables modellers to use them to render pages to describe models, or activate specific browser plug-ins to create a richer web interface for viewing models.

The access control and presentational layer of PMR2 is managed by Plone (Aspeli, 2007), a Content Management System. The access control features of Plone allow authorized users to manage permissions for other users, such as allowing a user to view a private workspace, push changesets, create exposures and update workflow states such as expiring exposures.

3 CONCLUSION

PMR2 is accessible to the end user over a web interface. Exposure pages in a PMR2 based repository can be accessed simply by entering the URL of the repository into a web browser. PMR2 has been used to create the CellML Model Repository, which currently contains over 500 CellML models and can be viewed at <http://models.cellml.org/>. Modellers will need to use a Mercurial client to create local clones of existing workspaces.

In summary, the key features of PMR2 are:

- facilitated model exchange directly between modellers, without reliance on a central repository;
- a detailed revision history for each model;
- user access workflows to control privacy when required; and
- embedded workspaces to enable model reuse and promote modular model development.

The combination of these features, together with the CellML language and associated metadata specifications, provides a collaborative model development environment that is capable of enhancing communication throughout the modelling community.

ACKNOWLEDGMENTS

The authors would like to thank everyone in the CellML community who has contributed ideas and feedback on PMR2.

Funding: British Heart Foundation (to A.G.); Maurice Wilkins Centre for Molecular Biodiscovery; Virtual Physiological Human Network of Excellence; and Wellcome Trust.

Conflict of Interest: none declared.

REFERENCES

- Aspeli, M. (2007) *Professional Plone Development*. Packt Publishing, Birmingham, UK.
- Baiju, M. (2009) *A Comprehensive Guide to Zope Component Architecture*. Lulu Press Inc. Raleigh, NC.
- Bornstein, B.J. et al. (2008) LibSBML: an API Library for SBML. *Bioinformatics*, **24**, 880–881.
- Christie, G.R. et al. (2009) FieldML: concepts and implementation. *Philos. Transact. A Math. Phys. Eng. Sci.*, **367**, 1869–1884.
- Cuellar, A.A. et al. (2003) An overview of CellML 1.1, a biological model description language. *SIMULATION*, **79**, 740–747.
- Gamma, E. et al. (1995) *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Reading, MA.
- Hucka, M. et al. (2003) The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, **19**, 524–531.
- Hunter, P.J. and Borg, T.K. (2003) Integration from proteins to organs: the Physiome Project. *Nat. Rev. Mol. Cell Biol.*, **4**, 237–243.
- Li, C. et al. (2010) BioModels Database: an enhanced, curated and annotated resource for published quantitative kinetic models. *BMC Syst. Biol.*, **4**, 92.
- Lloyd, C.M. et al. (2008) The CellML Model Repository. *Bioinformatics*, **24**, 2122–2123.
- Miller, A.K. et al. (2010) An overview of the CellML API and its implementation. *BMC Bioinformatics*, **11**, 178.
- Olivier, B.G. and Snoep, J.L. (2004) Web-based kinetic modelling using JWS Online. *Bioinformatics*, **20**, 2143–2144.
- O'Sullivan, B. (2007) *Mercurial: The Definitive Guide*. O'Reilly Media, Inc. Sebastopol, CA.