# Markup Languages for *In Silico* Oncology

David Johnson, Jonathan Cooper, and Steve McKeever

*Abstract*—**In silico models of cancer progression are numerous and diverse. Integration of different cancer models into virtual research environments and computational frameworks require the models themselves to be interoperable across the research community. In this paper we describe a vision to create a common computational framework in which these models can be formulated and stored in online repositories, such as the Center for the Development of a Virtual Tumor's (CViT) Digital Model Repository (DMR).**

## I. Introduction

THE European Commission (EC) funded Transatlantic TUmour MOdel Repositories (TUMOR) project aims to develop a platform for creating a collaborative research community and enabling the clinical application of cancer models. A key component of the platform will be a cancer model repository for storing and accessing models developed by two other major EC projects, ACGT (Advancing Clinico Genomic Trials on Cancer) [1] and Contra Cancrum [2]. The 'transatlantic' component of the project is to design the European repository to interoperate with the US-based CViT DMR [3] ultimately providing an international research environment that will connect experts in the United States and Europe. Facilitating exchange and reuse of cancer models from both perspectives is key to advancing the state-of-the-art in oncology.

## II. Motivation

Currently, the CViT DMR allows data files and executables of cancer models to be uploaded. However, there is no interoperability between these entries as there are no agreed standards on how such codes should be written. Cancer models are developed and implemented by hand, and require domain expertise in order to manipulate and evaluate simulation runs. In practice this means that there is no reuse of code or provision for coupling models that severely reduces the scope for collaborative developments. The heart modelling and systems biology communities have begun to address both issues through the creation of domain specific markup languages such as CellML [4] and SBML [5]. Until now however, these languages have not had the mathematical sophistication required by the cancer modelling community. With the development of FieldML [6] and the latest version of CellML, the required expressivity will soon be available.

We have begun to integrate these heart models into a database that will allow for interoperability and model reuse [7]. We aim to provide a similar level of functionality to the cancer modelling community, and for online model repositories, such as the DMR, by encoding tumour models in SMBL/CellML/FieldML. Existing models, however, will also be supported using annotations (namely our construction metadata discussed below) and wrapped so that they fit into overarching workflows. Re-coding will not be necessary and a degree of interoperability will be supported for existing codes.

The motivation for modelling markup is two-fold:

- Firstly we want to describe the implementation of these cancer models in an abstract manner that is not tied to any particular programming notation.
- Secondly we want to be able to couple our models. This kind of development has been undertaken in the context of heart modelling [8] but, to our knowledge, is not available as yet for cancer modelling. This has led to a bewildering proliferation of cancer models, most of which cannot easily be used by different research groups.

## III. Markup for Cancer Models

The Oxford University Computing Laboratory has substantial experience in developing such frameworks in the context of multi-scale modelling [9]. We aim to transfer this knowledge across to cancer modelling to create the first integrated, modular computational modelling framework for cancer biology. This will mean that it will no longer be necessary to "re-invent the wheel" as the next generation of researchers rewrite code that was written by previous researchers. It will provide the flexibility that is absolutely essential in allowing the modeller to focus on key issues for a particular application. Moreover, tool support in the form of type and unit checking will enable structural integrity of all models [10].

### A. Code Construction

A construction phase will automatically translate the models into optimized code for simulation on computational

Manuscript received May 14, 2010.

David Johnson is with the Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, OX1 3QD, United Kingdom. (phone: +44 (0) 1865 273838; fax: +44 (0) 1865 273839; e-mail: david.johnson@comlab.ox.ac.uk).

Jonathan Cooper is with the Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, OX1 3QD, United Kingdom. (e-mail: jonathan.cooper@comlab.ox.ac.uk).

Steve McKeever is with the Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, OX1 3QD, United Kingdom. (e-mail: steve.mckeever@comlab.ox.ac.uk).

frameworks that support the required numerical solvers. Tailoring the compilation will be achieved through appropriate construction metadata. This is data that is distinct from the models but describes how we would like them to be executed (SED-ML [11] could be incorporated here, although we will need more features than it currently provides). The construction need not be one to one. In fact a single description could be used to create multiple simulation codes, where each is capable of addressing a specific scientific issue based not only on a set of initial parameters but also on the selection of appropriate components. We also aim to provide a run-time framework similar to what Chaste (see below) provides for the heart modelling community but developed in such a way as to ensure low coupling. Other platforms will rapidly be supported to ensure portability of models.

A second family of metadata will target compile-time analysis techniques by describing the parameters of the overall system and those of each individual component. A generic platform based on SBML/CellML/FieldML allows for generic optimizations. Both continuous and discrete analysis methods will be supported. Various sensitivity analysis techniques will be studied to check for robustness or for allowing more coarse-grained components to be switched in and out during run-time calculations. This would occur based on the accuracy required to ensure computational tractability. A separate class of analysis techniques would rely on traditional compilation methods and examine the types of parameters to ensure more efficient evaluation using provably correct optimizations [12], [13]. Developing such generic optimization techniques means that they could be applied to all models in our repository. It is also important to ensure that these optimizations are correct, in other words that they do not introduce errors into our calculations.

### B. *The Chaste Framework*

We plan to apply the proven functionality of the Chaste [14] computational framework, developed for cardiac electrophysiology, to the area of multi-scale cancer modelling. Both application areas make use of a common core for describing meshes, linear algebra (based on PETSc [15]) and solving ordinary differential equations and partial differential equations (using the finite element method). On the cardiac side, CellML is used to describe the single cell models, which are systems of ordinary differential equations. The PyCml tool [16] translates these CellML descriptions into Chaste-compatible C++ code. Currently, Chaste can load (tetrahedral) meshes in several formats, including VTK and the 'triangle' data format. It is envisaged that once FieldML and a suitable API exist, support for this standard will be added to Chaste in a similar fashion, albeit that FieldML provides more information than just a mesh. Support for SBML is being considered, and also for CellML 1.2 in the future. As a starting point, we will begin with the encoding of a continuous model in a joint CellML and FieldML specification. This encoding will be used to drive our construction phase development that will create a prototype compiler for such applications. A second phase would focus on discrete model formulations (cellular automata and agent-based modelling) that will be described using the state features of the emerging CellML 1.2 format. Once this second compilation phase is complete we would aim to integrate all three cell based models, each addressing a different problem, on to one platform.

For coupling multi-scale models a third family of metadata will be developed. An ontology is needed to describe the environment for each component and not just the parameters. Thus we plan to include contextual interactions, such as the adhesive properties of the membrane or the extra-cellular environment in which the tumours reside, into our models. The long-term goal is to allow modellers to retrieve models from different repositories and couple them together in order to run simulations that address a specific scientific query in a seamless fashion.

Our approach will be incremental and allow the bio-chemistry to be developed alongside the coupling of other phenomena. We aim to provide robust tool support in order to facilitate mathematical modelling and ensure a greater degree of modularity and integrity of resultant simulation codes.

## IV. Conclusions

Portability of computational cancer models will be essential for facilitating future research in an international research environment. In this paper we described our aim to extend on emerging model markup languages that have developed out of cardiac modelling, to apply them to models used in cancer modelling. Enabling portability of cancer models, along with the development of authoring and code construction tools, will serve to facilitate cooperation, sharing, and advancement in *in silico* oncology.

## References


[1] M. Tsiknakis et al, "Semantic Grid Infrastructure Enabling Integrated Access and Analysis of Multilevel Biomedical Data in Support of Postgenomic Clinical Trials on Cancer," *IEEE Trans. Inf. Technol. Biomed.,* vol. 12, no. 2, pp. 205-217, March 2008.

[2] K. Marias et al, "Contra Cancrum Website." Internet: www.contracancrum.eu [May 14, 2010].

[3] T.S. Deisboeck, L. Zhang, and S. Martin, "Advancing Cancer Systems Biology: Introducing the Center for the Development of a Virtual Tumor, CViT," *Cancer Informatics,* vol. 5, pp. 1-8, March 2007.

[4] C.M.Lloyd, M.D.B.Halstead, and P.F.Nielsen, "CellML: its future, present and past," *Progress in Biophysics and Molecular Biology,* vol. 85, pp. 433-450, 2004.

[5] M. Hucka et al, "Evolving a Lingua Franca and Associated Software Infrastructure for Computational Systems Biology: The Systems Biology Markup Language (SBML) Project," *Systems Biology,* vol. 1, no. 1, pp. 41-53, June 2004.

[6] G. Richard Christie and P.M.F. Nielsen, S.A. Blackett, C.P. Bradley and P.J. Hunter, "FieldML: Concepts and Implementation," *Philosophical Transactions of the Royal Society A.,* vol. 367, no. 1895, pp. 1869-1884, May 2009.

[7] D. Gianni, S. McKeever, and N. Smith, "euHeartDB: A Web-enabled Database for Geometrical Models of the Heart," in *Proc. 5th Int. Conf. Functional Imaging and Modeling of the Heart,* Nice, France, LNCS 5528, pp. 407-416, 2009.

[8] A. Garny et al, "CellML and Associated Tools and Techniques," *Philosophical Transactions of the Royal Society A.,* vol. 366, no. 1878, pp. 3017-3043, September 2008.

[9] Y. Sun, S. McKeever, K. Balali-Mood, and M. Sansom, "A Multiscale Model for Efficient Simulation of a Membrane Bound Viral Fusion Peptide," in *Proc. IEEE Int. Conf. Bioinformatics and Biomedicine,* Fremont, CA, 2007, pp. 294-301.

[10] J. Cooper, and S. McKeever, "A model-driven approach to automatic conversion of physical units," *Software - Practice and Experience,* vol. 38, no. 4, pp. 337-359, April 2008.

[11] D. Köhn and N. Le Novère, "SED-ML An XML Format for the Implementation of the MIASE Guidelines," in *Proc. 6th Int. Conf. Computational Methods in Systems Biology,* Rostock, Germany, LNBI 5307, pp. 176-190, 2008.

[12] J. Cooper, and S. McKeever, "Experience report: a Haskell interpreter for CellML," in *Proc. 12th ACM SIGPLAN Int. Conf. Functional Programming,* Freiburg, Germany, 2007, pp. 247-250.

[13] J. Cooper, S. McKeever, and A. Garny, "On the Application of Partial Evaluation to the Optimization of Cardiac Electrophysiological Simulations," in *Proc. ACM SIGPLAN Symposium on Partial Evaluation and Semantics-based Program Manipulation,* Charleston, SC, 2006, pp. 12-20.

[14] J. Pitt-Francis et al, "Chaste: a Test-driven Approach to Software Development for Biological Modelling," *Computer Physics Communications,* vol. 180, no. 12, pp. 2452-2471, December 2009.

[15] S. Balay et al, "PETSc Web page." Internet: www.mcs.anl.gov/petsc [May 14, 2010].

[16] J. Cooper, "Automatic Validation and Optimisation of Biological Models," D.Phil. thesis, University of Oxford, Oxford, UK, 2009.