ELSEVIER

Progress in Biophysics & Molecular Biology 85 (2004) 433–450

Progress in Biophysics & Molecular Biology

www.elsevier.com/locate/pbiomolbio

CellML: its future, present and past

Catherine M. Lloyd*, Matt D.B. Halstead, Poul F. Nielsen

Bioengineering Institute, University of Auckland, Level 6, 70 Symonds Street, Auckland, New Zealand

Abstract

Advances in biotechnology and experimental techniques have lead to the elucidation of vast amounts of biological data. Mathematical models provide a method of analysing this data; however, there are two issues that need to be addressed: (1) the need for standards for defining cell models so they can, for example, be exchanged across the World Wide Web, and also read into simulation software in a consistent format and (2) eliminating the errors which arise with the current method of model publication. CellML has evolved to meet these needs of the modelling community. CellML is a free, open-source, eXtensible markup language based standard for defining mathematical models of cellular function. In this paper we summarise the structure of CellML, its current applications (including biological pathway and electrophysiological models), and its future development—in particular, the development of toolsets and the integration of ontologies.

Keywords: CellML; Markup language; XML; Ontologies; Mathematical modelling

1. Introduction

Over the past 20 years, the biological sciences have experienced an ''information explosion'' (Loew and Schaff, 2001; Hunter and Borg, 2003). New technologies have led to the elucidation of gene sequences, protein structures, mechanisms of gene expression, protein functions, metabolism and signalling pathways. Mathematical biology has an essential role to play in making sense of this vast amount of information. Mathematical modelling provides a method of integrating biophysical data which spans a wide range of temporal and spatial scales. In order to accurately represent the physiological processes which are occurring in the body, a hierarchy of models, from the subcellular through to the cellular, tissue, organ and organ system levels, will need to be integrated. Model integration requires a consistent mathematical framework, such as that being

*Corresponding author. Tel.: +64-9-3737599x85114; fax: +64-9-3677157.
E-mail address: c.lloyd@auckland.ac.nz (C.M. Lloyd).

0079-6107/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.pbiomolbio.2004.01.004

developed by the *Physiome Project*, an international effort to describe the physiology and the functional behaviour of the intact organism, in which the Bioengineering Institute at the University of Auckland is taking a leading role (Hunter et al., 2002; Hunter and Borg, 2003).

The considerable scale of the Physiome Project requires international and interdisciplinary cooperation. In order to facilitate the easy distribution and circulation of information and models among contributing scientists we are developing several eXtensible Markup Language (XML)-based markup languages (see W3C, World Wide Web Consortium, http://www.w3.org/). Specifically, at the cellular level, the development of the CellML language has been focused on providing a standard with which to represent models of cellular function (Cuellar et al., 2003; see CellML, http://www.cellml.org).

2. Background

2.1. The need for standards

There are many methods for representing biophysical models. Models may be represented at a conceptual level by spoken or written prose. This form of representation is generally not precise enough for rigorous creation, dissemination and checking. Models may be represented more rigorously at a mathematical level using mathematical symbols, capitalising on the well-tested foundations of mathematical knowledge. Mathematical equations provide a precise and high-level characterisation of the meaning of a model. This form is generally accepted as the preferred method for publicising biophysical models to the wider community. Models may also be represented at an algorithmic level as computer code. This form is a rigorous statement of how a model is to be computed, as opposed to the mathematical meaning of the model. It does not, in general, contain the semantic information of a mathematical representation. Because any mathematical model may, in general, be solved or approximated by a variety of algorithms (such as the Runga–Kutta, Method of Gear and Euler for integration), and such algorithms may be coded in a variety of computer languages (such as Java, Fortran, C and C++), for publication computer code is generally used only to supplement a mathematical description of a model.

There are, in turn, many methods for representing mathematical equations. Mathematical models have traditionally been published in journals as collections of symbols with generally well-accepted mathematical interpretations. Such representations rely on an implicit assumption that both the writer and reader of the equations have a common understanding of the terms used and their rules of interaction. Because of this implicit knowledge there is little that a computer can do to reliably interpret the meaning of such printed symbolic representations. In order to facilitate automatic interpretation of high-level mathematical equations, a greater level of semantic detail must be added. This has been the primary rationale for the creation of computer readable encodings of high-level mathematics such as OpenMath and Content MathML. Note that these forms of representing mathematical expressions have a well-defined semantic interpretation, as opposed to forms which are primarily concerned with the visual representation of the mathematical symbols, such as TeX and Presentation MathML.

Although mathematical modelling has been identified as a valuable method for analysing large amounts of experimental data, unfortunately, inaccuracies often arise with the current method of

mathematical model publication (Hedley et al., 2001; Hunter et al., 2002). Problems stem from the fact that models are developed and simulations are run in computer code, but they are then published in journals in the form of text and equations. Replicating published results, or further developing a published model, is frequently impeded due to errors introduced during the publishing process. Even when the source code in which the model has been developed is made freely available, the code is often specific to a particular computer platform, or it is incompatible with other modelling architectures, therefore delaying the integration of a model with others which are represented in another language. In addition, models of biological processes are best communicated when represented in terms of mathematical equations. Computer code is often an inappropriate level of representation for such models, and thus fails to capture the intentions of the model author.

Currently, one of the greatest challenges facing the field of computational cell biology is the lack of standards for defining models of cellular function (Hunter and Borg, 2003). Several alternative cellular function modelling languages have been independently developed (see Section 2.2); however, there is no single, common standard, and models need to be defined in a consistent format in order to facilitate their exchange between scientists.

A modelling language has to meet certain requirements. The basic structure of a modelling language needs to (a) embody the biological problem in a natural language context, (b) present a mathematical description of the biological problem and (c) represent both the mathematical and biological concepts in constructs that can be used in various computer applications to solve the model (Crampin et al., 2004).

2.2. The current solutions

In response to this need for a better understanding of complicated biological systems, research groups around the world are working on computational cell biology projects such as the *Virtual Cell*—a general computational tool which has been developed to build detailed models and run simulations of individual biochemical pathways (Leow and Schaff, 2001; see Virtual Cell, http://www.nrcam.uchc.edu/vcell_development/vcell_dev.html); *E-Cell*—a tool which is designed to simulate biochemical and genetic processes (Normille, 1999; see E-Cell Project, http://www.e-cell.org), and as such, acts as a complement to the Virtual Cell; COPASI (*COmplex PAthway Simulator*), a software package which has evolved from its precursor GEPASI (*GEneral PAthway Simulator*) to be able to simulate biochemical pathways in cells (see COPASI, http://www.vbi.vt.edu/research/projects/resproj_mends_copasi.htm); and *GENESIS* (GEneral NEural SImulation System), a neural electrophysiology simulator which is able to simulate of neural systems which range from complex models of single neurons to large networks made up of more abstract neuronal components (see GENESIS, http://www.gcnesis-sim.org/GENESIS).

Of all the languages currently available to describe models of cellular function, the *Systems Biology Markup Language* (*SBML*) developed by the ERATO Systems Biology Workbench Development Group at the California Institute of Technology, is the most prominent (Hucka et al., 2003; Finney and Hucka, 2003, http://www.sbml.org.specifications/sbml-level-2/version_1/sbml-level-2.pdf; see SBML, http://www.sbml.org). SBML is an XML-based language for describing models of biochemical networks such as signal transduction, metabolic pathways and gene regulation. SBML is an evolving language, and in the future it will be likely to include more

capabilities, but currently it has its limitations, such as its lack of ability to support large scale model building and hierarchical composition (Finney and Hucka, 2003). SBML is intended to provide a foundation for modelling biochemical networks; it does not focus on the integration of models of physiological processes over many spatial and temporal scales—a requirement that is central to integrate biology exemplified by the Physiome Project.

2.3. The emergence of CellML

CellML has evolved as a solution to the problems of inconsistencies between the computational and published models, and also to challenge the limitations of other computational cell biology modelling languages. CellML offers an unambiguous method of defining models of cellular function in a format that is both human and machine readable. It allows models to be exchanged across the Web, and shared among modellers, independent of modelling software. The range of the models that can be described using CellML overlaps with that of some other biological model specification languages. For example, SBML (Hucka et al. 2003), which "is oriented towards describing systems of biochemical reactions". CellML and SBML level 2 have adopted the same subset of MathML operators; therefore the two modelling languages are comparable in their ability to express mathematical relationships, and there is a degree of overlap between their applicability: although SBML is intended for use in describing reaction pathway models, it can also be used to express electrophysiological models. However, this does result in the loss of some of the model structure, as SBML does not have the relative freedom of CellML to define entities as components.

The structure of CellML was developed in order to meet the requirements of a modelling language previously described in Section 2.1. CellML is an XML-based language. XML was selected as the underlying language for describing CellML for three principal reasons: it is a widely adopted standard for describing domain-specific structured data; XML provides well-defined methods for integrating multiple domain-specific information into a single document; and many existing XML-based standards for describing domain-specific data are already available (e.g., MathML for describing mathematical expressions). From the start CellML has been designed to support large-scale modelling efforts: it is modular, enabling models and portions of models to be constructed independently and later integrated into a larger coherent model; and CellML provides facilities for information hiding so that lower level model details need not interfere with the current level of model development.

CellML uses Content MathML to provide both a human- and computer-readable representation, with unambiguous semantics, of the mathematical relationships of biophysical models. In addition, CellML provides further features that facilitate the process of model creation. In particular, the requirement that every variable has associated physical units enables CellML processors to automatically check equations for dimensional consistency. The modular structure of CellML through the use of components, encapsulation and import mechanisms facilitates the process of model building using previously tested models as bases for new models. Here again the requirement for physical units ensures that models interact correctly at the level of dimensional consistency.

We believe that many of the errors currently introduced during the traditional creation/dissemination/testing processes can be significantly reduced if CellML is used as the common

exchange format for the biophysical model. The ultimate goal is for model authors to publish a working CellML version of their model code when they publish their model study in a journal format. This would then ensure that future users can build on a correct, working version of the model. Rather than use their original source code, we would hope that the authors would use CellML as a common exchange format, such that it would be available to all potential users who may want to create, modify or solve a mathematical model.

This paper will go on to describe the current features of CellML, provide specific examples of how it can be applied, and also discuss its limitations and future directions. But before CellML can be fully introduced, first some technical terms need to be defined.

2.4. Technical terms

- An *ontology* can be defined concisely as a ''specification of a conceptualisation'' (Gruber, 1995). More explicitly, ontologies provide a controlled vocabulary of terms within a subject domain, their properties, or attributes, and also the relations between them. Ontologies provide a unified terminology for representing and communicating knowledge about a topic, and provide a formal structure for representation in machines.
- *Metadata* are ''information about information''. Ontologies are a form of metadata which, when applied to Web documents, can precisely and concisely describe the data they contain (Lassila, 1998). Metadata provide context for a document, such as the name of the document author, the date on which it was created and keywords related to its information content. Metadata are part of the *semantic* content of the Web. The term semantic stems from the Greek words for *sign*, *signify* and *significant*, and its present day understanding is *of, or relating to meaning*. In relation to the Web, semantic is referring to a structured description of a resource in a machine readable form. This is a concept which Tim Berners-Lee, the man responsible for inventing the Web, has been promoting (Berners-Lee and Hendler, 2001). Traditionally, Web searches have been based on finding matching words from within the document's main body of text, but increasingly, search engines are beginning to base their searches on the Web document's *semantic* content (Kamel Boulos et al., 2002).
- XML is a language which is both human and machine readable. XML is a set of rules for designing formats for structured documents.
- *Resource Description Framework* (RDF) is an XML-based framework for describing and exchanging *metadata*. RDF provides interoperability between applications that exchange machine-understandable information on the Web.
- Following on from RDF, *RDF Schema* (RDF-S) provides another layer to *metadata*. Most importantly, RDF-S presents the ability to support descriptions of classes of resources and their properties (Fensel et al., 2000).
- *Ontology Inference Layer* (OIL) and *DARPA Agent Markup Language* (DAML) are both ontology representation languages. DAML + OIL combines the two standards, overcoming the restrictions of each language by itself.
- *Web Ontology Language* (OWL) is the latest specification for a Web ontology language from the W3C Web Ontology Working Group. It is a direct successor of the DAML + OIL language, and it resolves some of the limitations of this earlier ontology language.

The specifications for XML, RDF, RDF-S, DAML+OIL and OWL languages are managed by the World Wide Web Consortium (W3C) (see W3C World Wide Web Consortium).

- *Transparent Access to Multiple Bioinformatics Information Sources* (TAMBIS) is a research project being run by the University of Manchester in the UK. The main aim of TAMBIS is to facilitate the acquisition of biological data by providing a single access point (an interface on the Web) for biological information sources round the world (see TAMBIS, http://imgproj.cs.man.ac.uk/tambis/).
- The National Library of Medicine's *Unified Medical Language System* (UMLS) defines a common set of terms for use in the development of computer systems that help health professionals and researchers to search for and integrate electronic biomedical information from a variety of different sources (see UMLS, http://www.nlm.nih.gov/research/umls/).
- *aMAZE* is a workbench for the representation, management, annotation and analysis of information on networks of cellular processes such as genetic regulation, biochemical pathways, and signal transductions (see aMAZE, http://www.amaze.ulb.ac.be/).
- *Biological pathways Exchange* (BioPAX) is a group developing a common exchange format for biological pathway data (see BioPAX, http://www.biopax.org/).
- *Open Knowledge Base Connectivity* (OKBC) is an application programming interface for accessing knowledge bases stored in knowledge representation systems (see OKBC, http://www.ai.sri.com/~okbc/).

3. Current features of CellML

This section will outline the basic structure of the CellML language. For a more complete description of CellML, please refer to the CellML 1.1 working specification (http://www.cellml.org/public/specification/index.html and Hedley et al. (2001).

3.1. Models and components

CellML models are represented as a network of interconnected *components* (Fig. 1). A component is the smallest functional unit of a model (a single component can represent a valid CellML model). Each component must contain at least one *variable*, and it may also contain mathematical equations to describe how that component behaves within the model. These equations are expressed in content MathML 2.0 (http://www.w3.org/TR/MathML2/), an XML-based language which is embedded within the CellML framework (Ausbrooks et al., 2001).

3.2. Variables

A CellML variable is a named entity that belongs to a single component. A variable may have attributes, such as an *initial_value*, *units*, *public_interface* and *private_interface*.

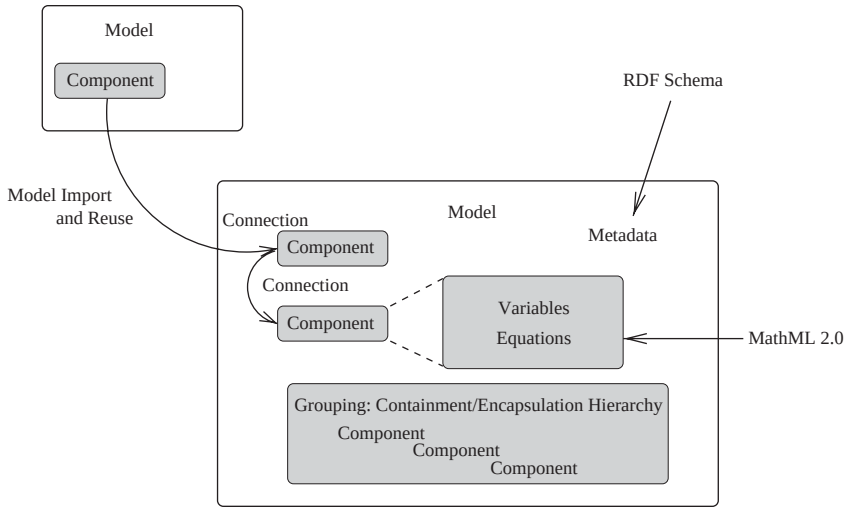

Fig. 1. A schematic diagram showing the general structure of a CellML model.

3.3. Mathematics

Mathematical equations are expressed using content MathML 2.0 (Ausbrooks et al., 2001), an XML-based language which is embedded within the CellML framework. In a CellML model mathematical expressions are defined within components under the MathML namespace. In addition to its extensive set of predefined containers and operators, MathML also provides a method for combining them in mathematical expressions. Although a CellML document may contain any MathML content markup elements, to encourage interoperability between software applications, we have defined a CellML subset of MathML content elements. This subset was introduced to facilitate interoperability between different software applications, and it is currently sufficient to describe the mathematics of most biological models.

3.4. Connections

Although a variable is declared and defined in one component, it is possible to map its value to other components in a model. However, it should be noted that the value of a variable that is declared with either a public_interface or a private_interface of *in*, may not be mathematically altered inside the current component, as its value is imported from a variable that belongs to another component.

The mapping of variables between components occurs via *connections*. These connections allow information to be exchanged between components in the network. A connection consists of a set of variable mappings between two components, and there can only be one connection between any two components in a model. Validating this rule helps to avoid establishing inconsistent or duplicate variable mappings between any two components.

3.5. Units

CellML requires that all variables and numbers be declared with a set of units. This feature of the CellML language ensures the robustness and reusability of the CellML components and models, as components and models containing variables with different units but the same dimensions (e.g., ounces and kilograms) can be connected. CellML provides a dictionary of standard units based on the International System of Units (SI) (BIPM, 1998, 2000; http://www1.bipm.org/utils/en/pdf/si-brochure.pdf, http://www1.bipm.org/utils/common/pdf/si-supplement2000.pdf), with additional SI-derived, and other non-SI units that are frequently used in the types of biological models described by CellML. User-defined units may be expressed in terms of the units already defined in the dictionary, allowing model authors to work with their units of choice while ensuring that their models can still be integrated with models which use other units.

3.6. Groups

Grouping adds structure to a model by defining named relationships between components. There are two predefined types of grouping in CellML: these are *encapsulation* and *containment*. Encapsulation is a logical type of grouping, or effectively a modelling convenience. Encapsulation allows the modeller to hide a group of components from the rest of the model by using a single component as an interface through which variables are exchanged between the hidden *encapsulated set* and the rest of the model components. If the current component is encapsulated, the encapsulating component is referred to as the *parent*, and all the components encapsulated by the same parent are the *sibling set.*

Encapsulation requires that each variable has two types of interfaces:

- A public_interface refers to the variable exchange interface exposed to components in the parent and sibling sets. An interface value of *out, in* or *none* implies the direction in the exchange of the variable value, i.e., variable export, or import, or no movement. If no direction is specified, the interface takes a default value of *none*.
- A private_interface refers to the interface exposed to components in the encapsulated set. Like the public_interface, a variable in the private_interface can also have a value of *out* or *in*.

Containment is used to describe the physical, or geometric, organisation of a model, such as biological structure. This type of grouping specifies that components are physically nested within their parent component. For example, a component representing a cell membrane channel maybe physically contained within its parent membrane component, but the cell membrane channel component does not have to be encapsulated by the membrane component, especially if there are variables exported from the membrane channel components that are used in other components in the model.

3.7. Import

All the features which have been discussed so far are common to both CellML versions 1.0 and 1.1. However, CellML 1.1 has the additional characteristic of *import.* Combined with the

component based structure of CellML 1.0, the feature of import allows CellML 1.1 to import components, connections and units from other existing models, and reuse them in the current importing model. This is a valuable feature of a modelling language which allows earlier models to serve as a framework onto which new components can be added and new models built.

The value of the reuse feature of CellML 1.1 is illustrated by the following example: Rice et al. (2000) presents a detailed model of the short-term interval-force relations in cardiac muscle. This detailed model is derived from two previously published models: the Jafri et al. (1998) electrophysiological model describing the action potential of a ventricular myocyte, and the Rice et al. (1999) model of force generation in cardiac muscle. By combining these two models into one, force generation is added to an electrophysiological model. The models are linked via a calcium transient provided by the cardiac cell model, which provides the stimulus for Ca^{2+} binding to troponin, and subsequent force generation by the myofilaments.

Rather than having to code up this new model in its entirety, it was possible to import components, variables and units from the two contributing models and combine them in a single model, as seen in Fig. 2.

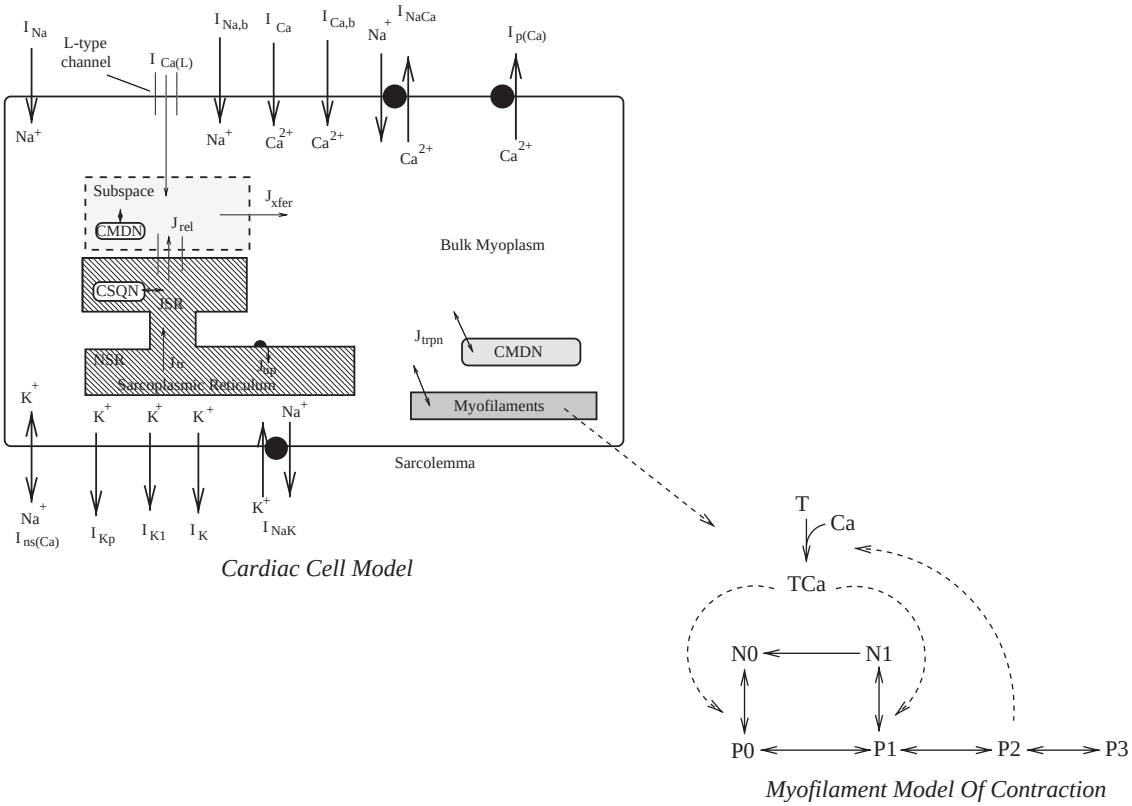

Fig. 2. A schematic diagram of a detailed, single ventricular cell model which is able to simulate action potentials, Ca^{2+}-handling and isometric force generation by myofilaments (http://www.cellml.org/examples/examples/import/rice_model_doc.html).

3.8. Metadata

Metadata are embedded in the CellML models using the W3C approved RDF standard (Cuellar et al., 2002; Lassila and Swick, 2003, http://www.w3.org/TR/2003/PR-rdf-syntax-grammar-20031215/). (It should be noted that where possible, CellML uses existing standards and languages in order to maximise the interoperability of CellML models.) Metadata are a machine readable form of information which provide context for a model by describing the literature reference on which the model is based, identifying the model creator, and supplying biological information about the model, such as the specific cell type for which the model has been developed.

As the CellML repository has grown to include over 200 models, finding a particular model of interest has become challenging. In order to make this process easier we have developed a "repository search" facility which allows you to search for a particular model author, or date, or keyword in which you are interested (see Repository Search, http://www.bioeng.auckland.ac.nz/physiome/php/repository_search.php). These searches are based on the metadata content of the model.

3.9. Model validation

We argue that CellML has been developed to eliminate the errors which arise during the process of model publication (see Section 2.1). CellML is an exchange language that facilitates the machine translation of working models into presentation formats, so it will remove the possibility of human error in representing a model that they have running in a simulation. With any representation language, there needs to be validation to ensure that it conforms to the syntax and semantics defined for producing or interpreting it. As an XML document, we can easily check for conformance, both as an XML 1.0 compliant document, and also as a CellML 1.0 or 1.1 compliant set of XML structures.

While model structure is easily checked through DTDs or XML-Schemas, or both, the semantics of the representation require other Schemas and processing tools. Ontologies are being developed to represent various levels of these semantics. All CellML models need to conform to the rules of the language as they set out in the specification (see CellML Specification), which include rules for connections, encapsulation, units, etc. These rules can be described using ontologies and then used by validation processors to check models. CellML models may also advertise that they conform to particular biological and mathematical concepts or to particular outcomes in simulation. Examples of these are particular mathematical formulations, biochemical pathways, simulation outputs, etc. This can be checked by validation processors that know how to validate a particular model against an ontological description. Such uses and toolsets are part of the intention of these logic based ontological languages such as OWL.

4. Specific examples of CellML models

The flexibility of CellML has allowed us to represent a wide range of models of cellular function. Over 200 models of cellular function are stored in the CellML repository (see CellML

Model Repository, http://www.cellml.org/examples/respository/index.html). Included among these are examples of electrophysiological, signal transduction, metabolic pathway and mechanical models. These different types of model are described in more detail below.

4.1. Electrophysiological models

In CellML, models are considered to be networks of discrete, but connected, components. These components may correspond to real biological, physical entities such as an organelle, or an ion channel or pump, or they can represent a convenient modelling abstraction, such as a component used to store all the constants in a model. Variables can be passed between the various components via their interfaces and connections.

Using the Dokos et al. 1996 model of the pacemaker activity of sinoatrial node cells as an example of an electrophysiological model (see Fig. 3), the type of cellular processes captured by the model equations can be seen. Sodium, calcium and potassium ions are exchanged between the intracellular and extracellular environments through channels, the sodium–potassium pump and

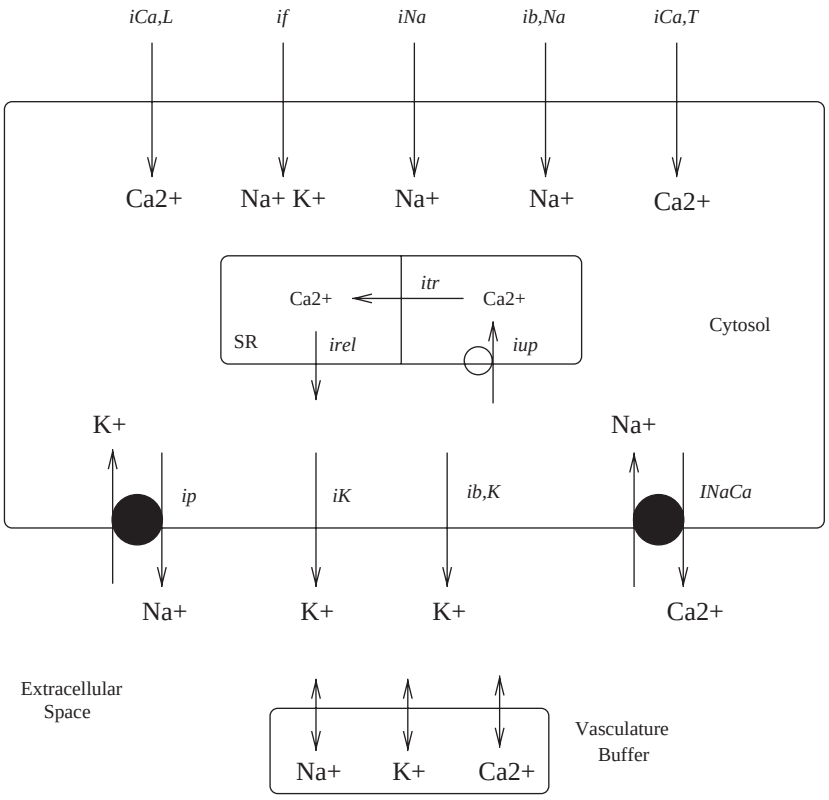

Fig. 3. A schematic diagram of the Dokos et al. (1996) mathematical model of the SA node cell. Sodium, calcium and potassium ions are exchanged between the intracellular and extracellular environments through channels, the sodium–potassium pump and the sodium–calcium exchanger. Calcium is transferred between the cytosol and the sarcoplasmic reticulum (SR), and between the local regions of the SR (http://www.cellml.org/examples/repository/dokos_model_1996_doc.html).

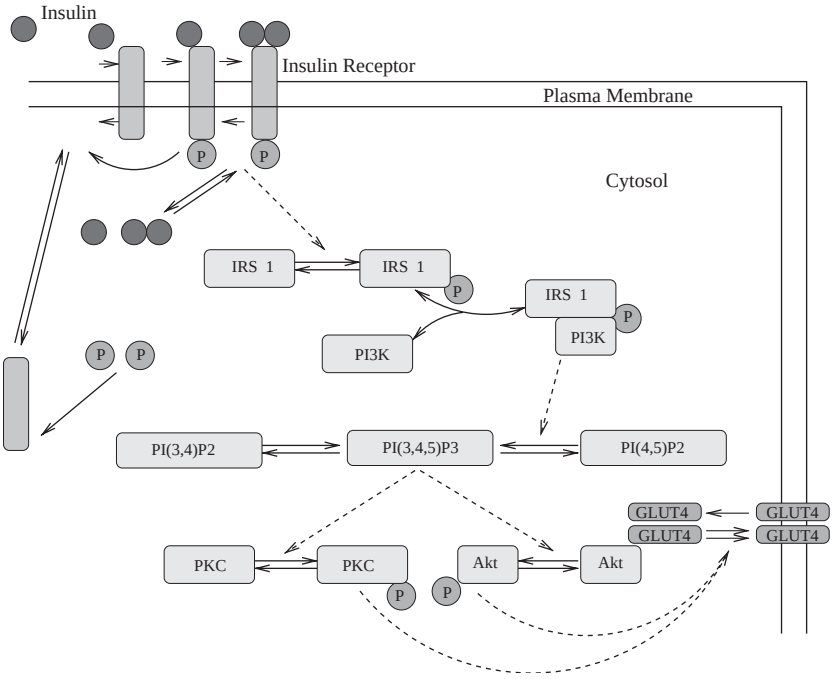

Fig. 4. A schematic diagram describing the reactions captured by Sedaghat et al.'s (2002) mathematical model of metabolic insulin signalling pathways (http://www.cellml.org/examples/repository/sedaghat_model_2002_doc.html).

the sodium–calcium exchanger. Calcium is transferred between the cytosol and the sarcoplasmic reticulum (SR), and between the local regions of the SR.

4.2. Reaction pathway models

When representing a biochemical reaction such as a metabolic pathway or a signal transduction pathway, the *reaction* element may be used to store information associated with a reaction within a CellML model. Associated with this reaction element are variables declaring the names and *roles* of the substrates involved in the reaction, such as *reactant*, *product*, *catalyst* and *inhibitor*. There is also a variable declared as *rate*, which often has a MathML expression associated with it to define the rate equation of the reaction. CellML also provides the ability to define the stoichiometry, directionality, and reversibility of a reaction.

A schematic diagram of the reactions between the stimulus insulin, membrane-bound protein receptors, and the intracellular reactants, products and enzymes, involved in an insulin-triggered signal transduction pathway, can be seen in Fig. 4.

4.3. Qualitative models

The above are examples of *quantitative* models, which have been derived from articles published in peer-reviewed journals and contain parameter sets which are based on experimental data.

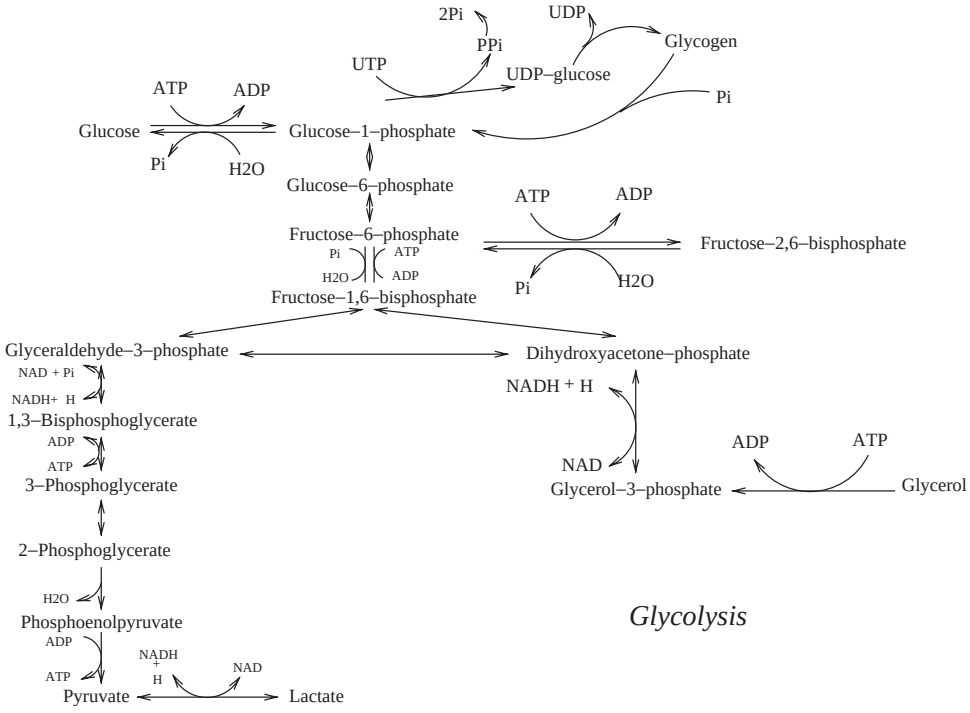

Fig. 5. Glycolysis metabolic pathway diagram. Defined in a textbook (Bronk, 1999) and used as the basis for a qualitative CellML model (http://www.cellml.org/examples/repository/qualitative/metabolic_models_doc.html).

Sometimes the detailed reaction kinetics of a pathway have not been characterised in a published mathematical model, but the pathway topography and the type of reaction kinetics have been elucidated. This is often the case with metabolism (see Fig. 5.), which is a qualitatively well-understood area of biology that is relatively poorly mathematically quantified. Since the relations between the reactants, products, enzymes and inhibitors of the reaction pathway are known, these reaction pathways can be translated into CellML as *qualitative* models.

5. Future directions

5.1. The integration of ontologies with CellML

CellML is an evolving language. Members of the CellML development team are currently publishing CellML version 1.1 (Cuellar et al., 2003), and CellML version 2.0 is now under development.

The evolution of CellML will integrate ontologies into its structure. As previously defined, an ontology is a formal representation of the concepts within a domain of interest, their properties, and the relations between them. For a modelling language such as CellML, an ontology can apply rules that specify how components are represented and connected.

One of the first ontologies to be integrated into CellML will be a reactions ontology. Biochemical reactions are well characterised and documented (Bronk, 1999). Associated with each reaction type there is a predefined rate equation, essential parameters and possibly additional associated species such as enzymes or inhibitors.

A reaction's ontology serves as an interface between the experimental biologist and the computer programmer. For example, a biologist can describe a reaction with the Michaelis–Menten kinetics, and since this reaction type forms part of the reaction ontology, a machine process can infer that the reaction must involve at least one reactant, product and enzyme, and a particular template for the reaction's rate equation (Figs. 6a and b).

The dual interface of such an ontology is an effective method for promoting communication between experimental biologists and mathematical modellers.

In addition to a reaction's ontology, it is intended that CellML 2.0 will be able to integrate other existing ontologies such as TAMBIS, UMLS and aMAZE. Another ontology that is being developed is an anatomical ontology. This ontology defines structural attributes across a range of spatial scales, such as organs systems, organs tissues and cells, as well as modelling concepts closely associated with anatomy, such as myocardial activation, thoracic cavity wall mechanics, etc. Initially, for CellML, the anatomical ontology will help to define the geometric containment hierarchy of a model. For example, the ontology may specify that components of the type "sarcoplasmic reticulum" cannot be physically placed within a component of type "nucleus" in a containment hierarchy. Both these organelles are present within the cytosol of eukaryotic cells,

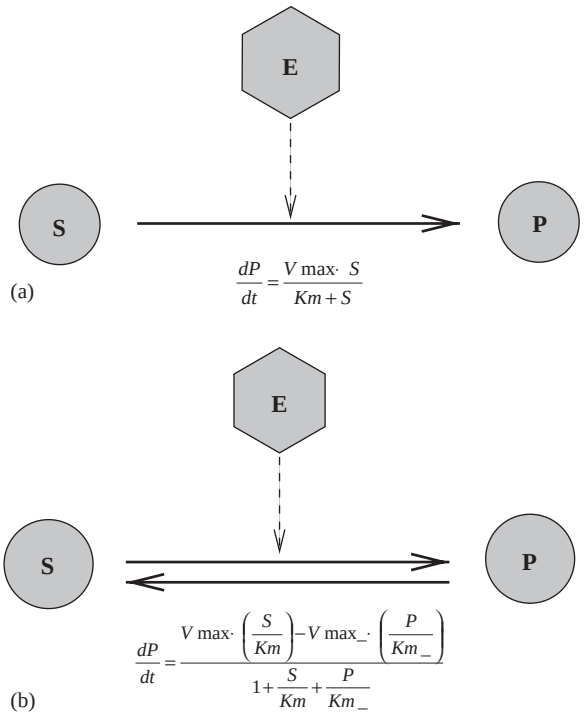

Fig. 6. Schematic diagram illustrating a standard Michaelis–Menten rate equation for (a) an irreversible reaction and (b) a reversible reaction.

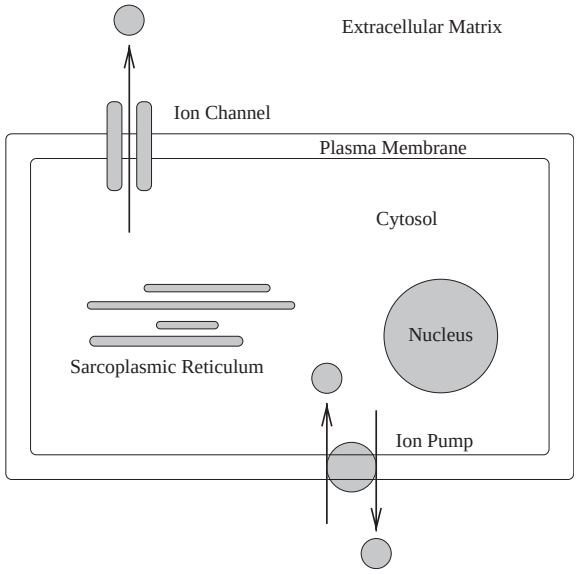

Fig. 7. This diagram is a schematic representation of a cell. It illustrates the positions of the channels, pumps and organelles relative to each other. With the integration of an anatomical ontology, biologically unrealistic containment hierarchies such as the physical grouping of the sarcoplasmic reticulum (SR), within the nucleus, within the cytosol would not be permitted in a CellML model. Instead, the SR and the nucleus would both be contained within the cytosol, but the SR would not be allowed to be physically contained by the nucleus.

but they are never nested within each other. These rules may be used by CellML processing software to help modellers avoid constructing biologically invalid models (Fig. 7).

At this point it is important to emphasise the fact that ontologies are evolving structures. If the model author wishes to represent something which is not in the current CellML ontology, this is possible. However, the only consequence is that this new element would not be subject to validation, as the ontology will only be able to check it against existing rules.

Besides providing structure for CellML models, an anatomical ontology offers a framework for organising the repository of cellular models. It will become possible to retrieve all models which include a specific ion channel, or all models which are written by a particular author, from the model repository.

In recognition of the growing need for a single, unified anatomical ontology, an international collaboration of several research groups, including The University of Edinburgh, The Stanford Medical Informatics Group, The Washington University Informatics Group, The Jackson Laboratory in Maine, and the Bioengineering Institute at The University of Auckland, has been established. We intend to incorporate this anatomical ontology into CellML.

5.2. Ontologies and software tools

The integration of ontologies with CellML will play a major role in the development of software tools. These tools include public model databases with human- and machine-readable

interfaces, including inference capabilities, tools for model visualisation, authoring and validation, and workflow systems for reviewing new or modified models.

Databases with reasoning capabilities are an important development for data modelling and repository systems. Their foundation brings together the mathematical roots of data representation systems and mathematical logic into a system that provides a natural way to represent concepts and relationships, and also expressive reasoning capabilities that help to retrieve sets of data matching complex relations. An important part of our philosophy is to provide interfaces that conform to standards developed by the open community for the purpose of data modelling, storage, knowledge representation and mining. These standards include DAML + OIL, OWL and RDF + RDF-S representations (Horrocks and Patel-Schneider, 2003), and OKBC compliant interfaces (see OKBC).

While ontology languages have firm roots in semantic systems, standards such as OWL and OKBC seek to bring this semantic modelling domain into a domain of modelling and standards used by software developers, the aim being that semantic data systems can be integrated into various end-user toolsets. For example, a visual tool that interfaces with a reaction pathway ontology, such as aMAZE or BioPAX (see aMAZE and BioPAX), could allow a biologist to visualise and select reactions that satisfy particular properties. A CellML model editor could use these to select between templates that can be used to base or extend models being developed.

A core objective in developing ontology databases is to provide interfaces between the human, who understands the concepts and the machine, which aims to provide unambiguous, consistent and accurate representations of models of biological processes. The process of developing and evolving ontology and model databases is an iterative cycle that necessarily brings together specialists across different fields of biology, mathematics and computer science. To establish this kind of international and cross-discipline cooperation, we are trying to establish a framework that attracts researchers to using CellML and its associated ontologies in their core work practices. Our effort is twofold: firstly, to establish a public CellML working group centred around mailing-lists and specification development; and secondly, to provide and collaborate on toolsets to facilitate the use of CellML, such as editors, validators, core language and simulation APIs, visualisation tools, and database interfaces to promote a workflow for submitting, reviewing and integrating models in repository systems.

5.3. The integration of CellML with other XML-based languages

The XML-based format of CellML allows other well established XML-based languages to be integrated into a CellML model. For example, we have already mentioned that content MathML is imported into the CellML framework to describe the mathematical relationships. In addition, it would also be possible to incorporate elements from other related XML-based languages, such as descriptions of small molecules using the Chemical Markup Language (CML).

Currently, CellML does not have a method for representing spatial elements in a model. However, we recognise that this is an important element of many biological processes, e.g., calcium will diffuse over an electrochemical gradient over time, and the length of this diffusion pathway will influence the model's behaviour (see Sneyd et al. 1998, for an example of a model which includes spatial variables). FieldML (see FieldML, http://www.physiome.org.nz/fieldml/pages/) is an XML-based language for describing spatially and temporally varying fields.

Although FieldML is still in its infancy, when it is fully developed and integrated with CellML, it will provide a spatial element to the mathematical models.

Integration of cellular models with mathematical models at the tissue and organ levels will also be made possible as CellML and the markup language AnatML are more closely integrated. AnatML (see AnatML, http://www.physiome.org.nz/anatml/pages/) is an XML-based language for describing the anatomical relationships between various types of biological objects.

6. Conclusion

The complexity of biological systems, combined with the availability of vast amounts of experimental data, can be effectively represented by mathematical models. CellML is an XML-based markup language designed to describe models of cellular function. It has evolved to meet the needs of the mathematical modelling community for standards with which they can exchange data and develop compatible models. CellML facilitates the exchange of models across the Web, and this form of communication effectively reduces the errors which are introduced by the traditional publication process of translating computer code into printed mathematical equations.

CellML is sufficiently flexible to be able to represent a wide range of model types, including electrophysiological, mechanical, signal transduction and metabolic pathway models. The import feature of CellML 1.1 also allows an existing model to act as a foundation onto which new components can be implemented and new models built.

CellML is an evolving language. Future developments include the integration of ontologies, and the creation of new toolsets. CellML is an open-source, free standard, which provides the potential for accelerated involvement from both the mathematical and biological communities. We would like to encourage the involvement of the wider modelling community in the development of both the language and toolsets. Further information can be obtained from the CellML website (http://www.cellml.org).

Acknowledgements

The authors would like to acknowledge Autumn Cuellar, David Nickerson, David Bullivant, Warren Hedley, Melanie Nelson and Peter Hunter; members of the CellML project team who, as part of the CellML project, have invested time and considerable expertise in the development and promotion of the concept.

References

Ausbrooks, R., Buswell, S., Dalmas, S., Devitt, S., Diaz, A., Hunter, R., Smith, B., Soiffer, N., Sutor R., Watt, S., 2001. Mathematical Markup Language (MathML) Version 2.0. W3C Recommendation 21 February.

Berners-Lee, T., Hendler, J., 2001. Publishing on the semantic web. Nature 410, 1023–1024.

BIPM, Bureau International des Poides et Mesures, 1998. The International System of Units (SI).

BIPM, Bureau International des Poides et Mesures, 2000. The International System of Units. Supplement 2000: addenda and corrigenda to the 7th edition (1998).

Bronk, J.R., 1999. Human Metabolism: functional diversity and integration. Addison-Wesley Longman, Harlow.
Cuellar, A.A., Nielsen, P.F., Bullivant, D.P., Nickerson, D., Hedley, W., Nelson, M., Lloyd, C.M., 2003. CellML Specification 1.1. Draft—30 September 2003.
Cuellar, A.A., Nelson, M., Hedley, W. CellML Metadata 1.0 Specification. Working Draft 16 January 2002.
Crampin, E.J., Halstead, M., Hunter, P., Nielsen, N., Noble, D., Smith, N., Tawhai, M., 2004. Computational physiology and the Physiome Project. Exp. Physiol. 89, 1–26.
Dokos, S., Celler, B., Lovell, N., 1996. Ion currents underlying sinoatrial node pacemaker activity: a new single cell mathematical model. Theor. J. Biol. 181, 245–272.
Fensel, D., 2000. The semantic web and its languages. IEEE Intell. Syst. 15 (6), 67–73.
Finney A., Hucka, M., 2003. Systems Biology Markup Language (SBML) Level 2: structures and facilities for model definitions, 28 June.
Gruber, T.R., 1995. Toward principles for the design of ontologies used for knowledge sharing? Int. J. Human–Comput. Stud. 43 (5–6), 907–928.
Hedley, W.J., Nelson, M.R., Bullivant, D.P., Nielsen, P.F., 2001. A short introduction to CellML. Philosophical Transactions of the Royal Society. Mathematical, Phys. Eng. Sci. 395 (1783), 1073–1089.
Horrocks, I., Patel-Schneider, P.F., 2003. Three theses of representation. In: Proceedings of the 12th International World Wide Web Conference. Budapest, Hungary, pp. 39–47.
Hucka, M., Finney, A., Sauro, H.M., Bolouri, H., Doyle, J.C., Kitano, H., et al., 2003. The Systems Biology Markup Language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics 19 (4), 524–531.
Hunter, P.J., Borg, T.K., 2003. Integration from proteins to organs: the Physiome Project. Nature Rev. Mol. Cell Biol. 4 (3), 237–243.
Hunter, P., Robbins, P., Noble, D., 2002. The IUPS human physiome project. Eur. J. Physiol. 445 (1), 1–9.
Jafri, M.S., Rice, J.J., Winslow, R.L., 1998. Cardiac calcium dynamics: the roles of ryanodine receptor adaptation and sarcoplasmic reticulum load. Biophys. J. 74, 1149–1168.
Kamel Boulos, M.N., Roudsari, A.V., Carson, E.R., 2002. Towards a semantic medical Web: HealthCyberMap's tool for building an RDF metadata base of health information resources based on the Qualified Dublin Core Metadata Set. Med. Sci. Monitor 8(7), 124–136.
Lassila, O., 1998. Web metadata: a matter of semantics. IEEE Internet Comput. 2 (4), 30–37.
Lassila, O., Swick, R.R., 2003. RDF/XML syntax specification (revised) W3C Proposed Recommendation, 15 December.
Loew, L.M., Schaff, J.C., 2001. The Virtual Cell: a software environment for computational cell biology. Trends Biotechnol. 19 (10), 401–406.
Normille, D., 1999. Complex systems: building working cells 'in Silico'. Science. 284 (5411), 80–81.
Rice, J.J., Winslow, R.L., Hunter, W.C., 1999. Comparison of putative cooperative mechanisms in cardiac muscle: length dependence and dynamic responses. Am. J. Physiol. 276, H1734–H1754.
Rice, J.J., Jafri, M.S., Winslow, R.L., 2000. Modeling short-term interval force relations in cardiac muscle. Am. J. Physiol. 278, H913–H931.
Sedaghat, A.R., Sherman, A., Quon, M.J., 2002. A mathematical model of metabolic insulin signalling pathways. Am. J. Am. Physiol. 283, E1084–E1101.
Sneyd, J., Wilkins, M., Strahonja, A., Sanderson, M.J., 1998. Calcium waves and oscillations driven by an intercellular gradient of inositol (1,4,5)-triphosphate. Biophys. Chem. 72 (1–2), 101–109.