Title: Modularity (requiring ontologies and standards) is the lynchpin for collaborative multi-scale modeling

Authors: WG10 members
.....J.Bassingthwiaghte 9aug10 ... as a starter jbb2@uw.edu 206-685-2012,
.....Addition by D. Beard 18aug
.....Revision and addition by JB on 20aug10 Search on ?? for your name

Abstract:
Modular construction is the natural form of biological systems at all levels and is the efficient mode of construction of models of biology. Interdisciplinary collaboration is flourishing and is advancing the rates of scientific progress. Models developed in different laboratories on different platforms can be brought together if they adhere to common standards and ontologies. The strategies used for multi-scale model building serve also for substituting modular elements of differing complexity and robustness for one another in order to meet varied demands, e.g. for computational speed, simplicity of operation, robustness, precision, or mechanistic denouement. A particular goal is the automated construction of models from component modules archived in standardized form. We propose herein some recommendations for defining the characteristics of modular components and suggest some of the requirements for automating constructing higher level models from available modules.

Key Words: modular modeling, ontologies, multi-scale, model standards, databases, SBML, CellML JSim,, computational methods, spatial and temporal continua, stochastic calculation.

INTRODUCTION:

Generally speaking, anything beyond a one-compartmental model can be regarded as being composed of modules. These might be each of a series of enzymatic reactions in a biochemical network, particular functions of intracellular organelles, sequences of steps in a feedback system, or a whole organ. This raise the question of whether or not the modular nature can be formalized to the extent useful for broad usage in constructing multi-scale systems.

A "module" in a model is computer code for performing a function or providing an input/output operation. Modules developed by labs around the world could be reused by investigators formulating new higher level integrated models. But the modules have to be correct and understandable. While a module may be internally complicated, its number of "connections" to the region external from it are always limited. Modules with the fewest connectors are generally the easiest to define, to connect and to maintain.

Large models are difficult to maintain. If they are useful they are continuously being updated as information is gathered and as purposes diverge. Einstein's admonition that "models should be as simple as possible but not too simple" makes sense if the goal is to capture the essence of the idea, but when referring to comprehensive systems models, defined as describing evolving situations where the models grow to absorb the latest discoveries, it is not so appropriate. Biophysical modules, on the other hand, can be simpler to define and the model easier to maintain, since they remain relatively unchanging as the science advances. A complex model composed of many modules could be automatically updated when any module is improved. The connections for a new version of a module usually do not change even though the intramodular construction and internal equations change; then the substitution of the new for the old is simple, as nothing is changed in the master model.

An example of a complex, single level model is one for the regulation of the ionic concentrations in an excitable cell. Consider each channel, ion pump, transporter, or exchanger as an independent module. The "environmental" conditions for all of them are the composition of the external and internal milieux and the transmembrane voltage. Given their instantaneous conditions the time and voltage dependence of the internal conditions, the conductances and then the fluxes can be calculated from the electrophysiological equations. Since the modules are all, in this case, totally independent of one another, except through their varied influences on the membrane potential and the transmembrane concentration differences, their internal calculation are uninfluenced by other modules. This is then an ideal situation in which variants of a chosen module can be inserted in order to determine the influence on the overall system.

A particular example is that of the Iks channel of the cardiomyocyte as demonstrated by Silva and Rudy (2010). The module for this channel could provide the kinetics of the normal channel or that of the abnormal mutation (KcnQ1) giving rise to the LongQT syndrome, in which the repolarization of the membrane is slowed. These authors also determined the time and voltage dependencies of the channel using computational molecular dynamics, so in principal their supercomputer calculations could also serve as an equivalent module; this makes it obvious that module simplification or reduction is the usual goal in making a substitution.

(Their modeling is a masterpiece of integrative systems modeling, going from the gene sequence to the protein conformational states, to the channel conductances and current flows, to the spread of excitation and the susceptibility to arrhythmia in the intact contracting heart. The Long QT Syndrome is, I think, the first disease, causing sudden death in young athletes, whose mechanisms have been clearly elucidated from gene to organ and organism in humans. Even cystic fibrosis is not so nicely defined.)

The automated construction of a model for cellular ionic regulation is now a reality, given a set of modules for channels, pumps, transporters, and exchangers using a common ontology and consistently formatted module code that can be interpreted for assembly and aggregation. The ontology must be complete (perhaps using synonyms) for the parser to recognize the identity of the elements and to preserve their uniqueness. The parameters must also be uniquely named for each of the modules, especially if they are to be merged into a single master program rather than maintained in isolation within a subroutine.

Defining a module:

(A) A module is a well-defined unit, and has a defined singular purpose.
(B) A module can be archived, always available,
(C) A module is reproducible, thoroughly documented, verified for numerical accuracy, and validated for a variety of situations, and thereby documents a part of a multicomponent model.
(D) A module serving a particular purpose can come in different forms of varied complexity, accuracy, or computational speed. Reduced forms may be of validity limited to a fraction of parameter space. Such a set of alternative modules allows great flexibility in designing a model for a particular purpose.
(E) A preformed module, is analogous to standard mathematical routines like sin(x) or log(Y). It may be written in re-entrant code allowing multiple uses in different parts of an integrated models is facilitated by the existence ;

Modularity is the lynchpin for integrative modeling

The lynchpin (or linchpin) is the pin inserted through the axle of the cart to prevent the wheel slipping off the axle; It holds the whole contraption together, and is the key to making progress. Without it, the cart collapses. The cart is inherently modular, and cannot be constructed without affirming the nature of its modularity. The arguments for modules being central to multi-scale modeling are similar to those for making carts from parts or integrated circuits from components, only one or a few of which can be regarded as lynchpins:
1. Distribution of labor: Different modules can be develop by different people, living in different places.
2, The expertise of a particular group can be captured and transferred through the production of the module.
3. Diverse expertise of a variety of groups can be collated and shared to advance the science more rapidly.
4. Code verification and evaluation for validity can be done by groups outside of the original design group.

5. Collaboration is facilitated not just by the common usage of modules but by the acts of critiquing, testing, validating, and using in different ways by different groups.
6. Public archiving for free distribution enables investigators world wide to build upon the work refined into a modular element.
7. Databases of archived modules are essential for the furthering of of collaborative research and of further independent research starting in new areas.

Defining the internal structure of reproducible and sharable modules:

The context for modular construction of large models in biology is describable by three levels:
1. The domain: This relates to the anatomy, e.g. the cell as a mixing tank and the extracellular fluid another mixing chamber. An enzyme is restricted to a location in one chamber, but solutes pass from chamber to chamber. The "domain" of a module might be the membrane, thereby requiring it to have links to one or both external domains (e.g. as for a channel), or in a defined solution within a region (e.g. intramitochondrial or inside the nucleus or in the plasma)

2. The species: In a biochemical setting the species are the reactant solutes and the enzymes which facilitate reactions. An enzyme "operates" on a substrate to produce a product, or vice versa, and while the enzyme may take a variety of forms in the process, from the external system point of view it may be only necessary to identify three measures: the rate of substrate usage, the rate of product formation, and the amount of solute (substrate or product) bound to the enzyme. The three information items allow the calculation of mass conservation, used to verify the the model computation as mathematically reasonable.

3. The operator: The enzymatically facilitated reaction transforms A to B or A+B to C+D or other reaction type. How it does that is internal to the operator, thus allowing a separation of computer code for a module into 2 types: **internal** and **external.** The **internal code** comprises the body, or innards, of the module. The **external code** provides the links to the overall multimodular domain.

For channels, pumps, transporters, exchangers and other mechanisms for permeating membranes, the external code needs only the rates of exchange for each of the substrates or products in order to calculate the concentrations of each of the species. This is convenient, for there may be many simultaneous influences on the concentration of a solute, and all need to be accounted for in the **domain** common to those various operators. Thus in the **external code** the modular code provides what is need for the domain calculation.
The internal code defines the operation. It uses the **external conditions** defined within the domain, the **parameter values** for the operator and a set of initial conditions for the internal variables. (The default initial conditions are could be simply the steady state conditions for the operator under the external conditions, or could be as if the external concentrations had been zero. This arbitrariness is a potential source of error.)
The role of the module's internal code is to determine the physical-chemical response or provide a precalculated descriptive response to the inputs and to return the output information to the external domain. The external domain can then takes the information, along with that from other modules and integrates it appropriately.

Sharing and Disseminating Model Code

Most model code never leaves its programmer's institution. It is difficult to document and explain computer code. The time-consuming effort to create manuals and tutorials is unrewarded and unfunded under our current granting system . The first level of success in modeling is a model useful in research, and therefore publishable. To take a substantial model, e.g. Noble's model of the human cardiac action potential (2004?) from the stage where it is verified to be mathematically correct and validated scientifically as an analog of the real-life phenomena, to a distributable, understandable, proven reproducible model that others can build upon is a huge task. The labour is 10 to 20 times that of producing the original model. Journals do not require, yet, that the model should be available to the reviewers of the article, or to require that it be archived and available to the public or to the readers of the journal. But the funding agencies now are clear in asking that the results of scientific work be shared. This means that adherence to minimal standards will soon be required. It is not appealing to think that adherence to quality standards is to imposed on the scientist, but it is appealing to feel that the hard work of making models reproducible will be appreciated. Having models to provide to others is a great start on a fruitful collaboration.

Achieving Reproducibility in Reporting on Models

There are surprisingly few models that can actually be reproduced from the original published paper. Hodgkin and Huxley (1952) set a high standard: their figures can be reproduced from the equations and parameters they provide. The field of electrophysiology is exceptional in this regard: the classic papers of Noble (1962) and of Beeler and Reuter (1977) are likewise reproducible.

Reproducibility has twin aspects: utility and transparency. Adherence to notational and formatting standards makes for ease of utility. Clarity of presentation and using step by step logic in explaining the model, it's principal function, its perspective and what can be done with it as a building block all help to make it useful as a stepping stone for others. A set of "Standards for Biophysical Models" is available at www.physiome.org/Models/standards.html. These set a high bar, for it is almost impossible to fulfill all the requirements for the "Class 4" biophysically-based models described there. At a minimum there should be unitary balance (Chizeck et al The problem is the difficulty in demonstrating exact mass balance, charge balance, energy balances and osmotic balance, and in fact most models do not need to adhere to ALL of these. However at the top of that list are unitary balance and mass balance. Unitary balance is mandatory and without it there are errors, almost always. Mass balance, that is , conservation of mass, volume by volume and species by species is easier to attain, and is a critical part of the verification that the model is correctly computed.

The initial keys to model reproducibility are logical construction of the model and clear presentation in the publication. ALL of the equations and parameters should be in the published article, without typographical errors, with units on everything, and with source references for all of the parameter values. One way of achieving this state of blessedness is to have the journal's reviewers test the model, and reproduce the figures.

An early example of a collaborative success in this approach occurred with the publication of the action potential model of Winslow et al (1999; Greenstein, 2000). As a reviewer, and having a well-written manuscript in hand, we coded the model in JSim from their tables and equations. On finding a few problems we communicated with the authors, corrected the equations while they corrected the manuscript, and through a couple of iterations achieved consilience between our code in JSim, their code, and the manuscript presentation. The paper was then published, released on a Thursday afternoon at 4 PM coincident with our release of their model on the Physiome website. (www.physiome.org/Models/CellPhysiology/ActionPotential).

The BioModels database (www.ebi.ac.uk/Biomodelsliters) is an excellent repository of models which have been well curated. Many of these can be downloaded and run directly in JSim, but a significant fraction, starting with their first model, do not comply with scientific requirements in that they violate unitary balance. For example, an equation for the rate of change of an amount (moles/s) is calculated as a flux (moles/s) divided by a volume (liters). They tested the model on MathSBML, where it apparently runs without the detection of a fundamental flaw. Since the Biomodels curation uses only a few simulation systems, and none of these have automated unit balance checking, discovering all errors is difficult, even in the hands of their very experienced group and even though they are the designers of MIRIAM (Minimum Information Requested In the Annotation of biochemical Models) (LeNovere et al 2005). The intent of MIRIAM is to make sure that selected published models are archived correctly, and that they can be downloaded and used, so the emphasis is on matching the model and the publication; improving the models to represent the biology better is not a part of their effort, and even though the 150 models provided are of uneven quality they do represent a big improvement over the source in the SBML database. Neither does the Biomodels effort attempt to impose scientific stands equivalent to those Standards proposed for the multiscale modeling effort (imagwiki.org/mediawiki ... This site).

Black Box Modeling:

Ideally, the operational equations and the internal parameters of the module can remain hidden. For example, take the Hodgkin-Huxley action potential model. The action potential is the event dominating our view of nerve ionic currents, and we tend ignore the roles of the pumps and exchangers that are required for homeostasis. When we model the action potential, none of the parameters for the time- and voltage-dependent conductances need be seen externally. We need only to know the time course of the fluxes of Na and K : these are the currents provided as the outputs from the computation. These currents are summed with any other currents (e.g. calcium current and the currents due to the ionic pumps like the NaKATPase) to obtain the total net charge transfer. From this one calculates, externally so to speak, the change in transmembrane voltage, Em, and from the ionic fluxes, the changes in the concentrations of Na and K on either side of the membrane.

In this scenario, following the description above for modular code, the changing values of the concentrations of Na and K inside and out and the Em are the inputs to the model. The fluxes are the conductance parameters are untouched and can remain hidden, even while available for adjustment.

Reusing modules:

Ideally, the modules should be reusable or re-entrant, so that the code is not rewritten for each instantiation. A compromise necessitated by the flat non-modular nature of JSim's compiled code is to automate the renaming all the code within a module being used a second or third time. Gary Raymond has devised a program for doing this, so that multiple versions of the same operator are given new names for internal parameters with each use. This is not so much of a problem in procedural languages that allow reentrant code. (FURTHER COMMENTARY NEEDED)

Dissemination and Reuse of Modules (Dan Beard)

Given the stated goal of integrating validated models together to achieve increasingly realistic representations of biological function, the distribution and reuse of component modules is of more practical value than the dissemination of published models. This is because integration of models naturally occurs at the lower "modular" level, rather than at the high level of the integrated models. As a concrete example, consider two biochemical systems models each simulating the kinetics of a set of enzymes catalyzing an integrated network of chemical reactions. These two models may overlap in terms of the reactions that they treat and therefore likely invoked different versions of individual reaction modules. Furthermore, reactant concentrations treated as state variables in one model may be represented by fixed parameters in the other. Therefore there is not likely to exist one unique integration of these two models.

When models representing the kinetic mechanism of the individual enzyme-catalyzed reactions in each module are available, however, these reaction modules may be combined in arbitrary ways. The Biochemical Simulation Environment (BISEN) package [Bioinformatics 2009 25:836-837] facilitates this sort of modular biochemical systems model building. Using this package, constructing a model from existing enzyme/reaction modules is the computational equivalent of choosing a set of enzymes from laboratory shelf, reconstituting them in a defined environment, and adding substrates at concentrations defining an initial condition.

BISEN is able to apply modular model design and reuse effectively, in part, due to its relatively narrow application domain. It is build on a strictly defined set of rules for defining modules and constructing integrated models. (See [Bioinformatics 2009 25:836-837] and user manual available at http:// bbc.mcw.edu/BISEN.) Similarly well-defined and community standards for module definition, description, and exchange will be required to achieve practical modular construction and reuse for a broader class of multi-scale models of biological systems.

Module Substitution: (concerns multiple forms of modules, various levels of approximation, variants to fit different species or different cell types, analogy to different types of enzymatic reaction in Michaelis-Menten form, etc.).

Modules for a given function may take a variety of forms to serve diverse processes. A primary reason for having module variants is to gain computational speed by using a functionally almost equivalent module that is faster to compute. Another is that a reduced form of a module may be good over a limited range of state-space, with alternative variants needed for other ranges. This sets up a situation where module substitution might be automated during computational runs, not just in model construction.

In multi-scale modeling, as one enlarges the scale of integration of a system, the system may develop complex dynamics, even overt chaos, while at the same time may become more stable, in the sense of becoming more robust, more resistant to perturbation by external inputs or internal failures, while cycling on an attractor. This stabilization on the attractor may allow further simplification of individual modules and allow further reduction in module complexity.

Modules for Multi-scale versus mono-scale models: Steve??

Ontologic issues: Max Neal??

The Common Module Format --- Module Standard Form:

Robustness and the need to return to complex module forms when the position in state space moves out of the circumscribed range of accurate operation of a reduced form module. How to recognize inadequacy in the module behaior, How to regroup on the Ify, while maintaining computational capability to optimize the model to a continuously acquired signal, e.g. as in monitoring a patient in the ICU and operating the model to adjust the IV inflow or call a nurse.

Modularity in Spatially distributed models: (FE and FV modeling especially)

Examples of automated model construction from modules (brief summaries of how each works by Gary and Max)

- SemGen (Max Neal??)
- FORTMOD (Gary Raymond??)
- Anything done with PCEnv? (Alan Garny??)

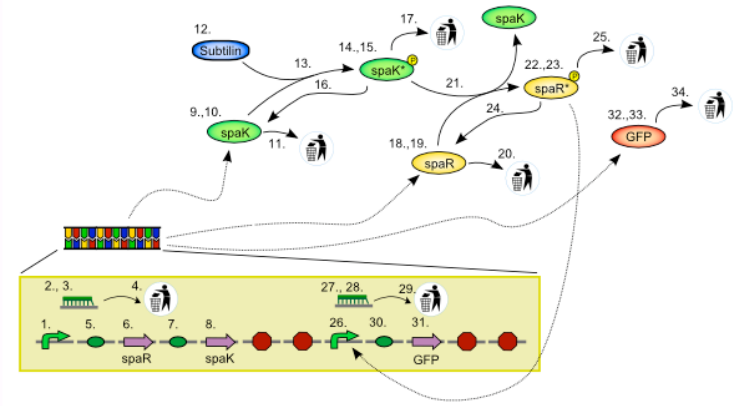
Archival forms and storage sites for Modules

CellML: CellML is primarily an archival XML form of a model definition. While it defines algebraic and differential algebraic and ordinary differential equations, it does not define the computational platform or the numerical methods or provide data for model validation. It does lend itself to network approaches as suggested in the Bugbuster System Model in the Figure. The methods of deriving equations from the diagram is not so clear: each one has to be individually defined. These elements do not seem to fit the definition of a module in the sense of the module defining an operator. Rather these elements are the objects (proteins, substrates) which are being operated upon, and the processes are the numbers in the figure:

Bugbuster Systems Model

Model Overview

This is a Systems Model for the gold medal-winning machine 'Bugbuster' produced by the team from Newcastle University for the iGEM 2008 competition. Details about the project can be found here: http://2008.igem.org/Team:Newcastle_University.



A constitutive promoter leads to the production of proteins spaK and spaR. Subtilin activates spaK which in turn activates spaR. Activated spaR is a transcriptional regulator for GFP. The system senses and reports subtilin concentrations via GFP.

SBML and SBW: Herb Sauro??

JSIm and Physiome.org repository Butterworth??

Application to Cellular Electrophysiology:

The cell membrane potential is defined by the net charge difference across a membrane and the capacitance of he membrane. The charge balance is governed by a set of ionic currents carrying charges across the cell memranes. The integral proteins involved include ion-selective channel proteins, exchangers or transporters, and energy-coupled pumps. The Hodgkin-Huxley (1952) model for the action potential in the squid giant axon pioneered quantitative modeling of ion fluxes and action potentials in excitable cells. Models for cardiac cells followed (Nobel 1962), though were soon found to be more complex, having for example, calcium channels (Beeler Reuter 1977) which had not been noted in the nerve studies.

The regulation of a cell's ionic milieu is an ideal application of modular methods. Each channel is highly selective. The conductances of the channels are time-and voltage dependent, but not dependent on the concentrations of the ions.The fluxes are driven by the electrochemical gradient and are therefore dependent on concentrations of the particular ions. Given the independence of each of the individual charge-carrying entities, each can be defined as a module and coded as a complete model. (In order to demonstrate the time- and voltage-dependence of the kinetics of the conductance changes, one would use a voltage clamp approach.) Then the integrating a selected set of entities into a call model can be automated. This has been accomplished with both SemGen and FORTMOD and the methods are being further evaluated and refined.

Application to Compartmental Systems

PKPD type models jb?? anybody??

Application to Membrane transport in axially distributed systems

FORTMOD has succeeded in incorporating bidirectional competitive transporters in covection-diffusion-permeaation-reaction systems using PDEs as well as with systems of ODEs. ??Gary??

SUMMARY:

??

References:

683. Beeler GW Jr and Reuter H. Reconstruction of the action potential of ventricular myocardial fibres. J Physiol (Lond) 268: 177-210, 1977.

10632. Chizeck HJ, Butterworth E, and Bassingthwaighte JB. Error detection and unit conversion. Automated unit balancing in modeling interface systems. IEEE Eng Med Biol 28(3): 50-58, 2009.

6795. Greenstein JL, Wu R, Po S, Tomaselli GF, and Winslow RL. Role of the calcium-independent transient outward current Ito1in shaping action potential morphology and duration. Circ Res 87: 1026-1033, 2000.

700. Hodgkin AL and Huxley AF. A quantitative description of membrane current and its application to conduction and excitation in nerve. J Physiol 117: 500-544, 1952.

7534. Noble D. A modification of the Hodgkin-Huxley equations applicable to Purkinje fibre action and pace-make potentials. J Physiol 160: 317-352, 1962.

8127. Le Novère N, Finney A, Hucka M, Bhalla US, Campagne F, Collado-Vidas J, Crampin EJ, Halstead M, Klipp E, Mendez P, Nielsen P, Sauro H, Shapiro B, Snoep JL, Spence HD, and Wanner B. Minimum information requested in the annotation of biochemical models (MIRIAM). Nature Biotech 23: 1509-1515, 2005.

8126. ten Tusscher KHW J, Noble D, Noble PJ, and Panfilov AV. A model for human ventricular tissue. Am J Physiol, Heart Circ 286: H1573-1589, 2004.

6646. Winslow RL, Rice J, Jafri S, Marba'n E, and O'Rourke B. Mechanisms of altered excitation-contraction coupling in canine tachycardia-induced heart failure, II: Model studies. Circ Res 84: 571-586, 1999.

Pedersen Michael. Modular Languages for Systems and Synthetic Biology. PhD Thesis University of Edinburgh. 2010

Frank T. Bergmann and Herbert M. Sauro. SBW - a modular framework for systems biology. In WSC '06: Proceedings of the 38th conference on Winter simulation, pages 1637–1645. Winter Simulation Conference, 2006

Lucian P. Smith, Frank T. Bergmann, Deepak Chandran, and Herbert M. Sauro. Antimony: a modular model definition language. Bioinformatics, 25(18):2452–2454, 2009.

M. T. Cooling, V. Rouilly, G. Misirli, J. Lawson, T. Yu1, J. Hallinan and A. Wipat Standard virtual biological parts: a repository of modular modeling components for synthetic biology. BIOINFORMATICS 26 (7): 925–931, 2010. doi: 10.1093/bioinformatics/btq063

CellML Model Repository at Auckland NZ http://models.cellml.org/cellml

SBML and the BioModels Database at EBI, Cambridge UK

JSim : http://nsr.bioeng.washington.edu/jsim

The Physiome Model Repository: http://nsr.bioeng.washington.edu/Models