THOUGHTS ON RULES FOR CONSTRUCTION OF MODULES for Multiscale Modeling. Draft for critiquing. J Bassingthwaighte 22jul08

GOAL: Automated construction of models from component modules archived in standardized form.

RATIONALE: Modules developed by labs around the world could be reused by investigators formulating new models. But they have to be correct and understandable. Roy Kerckhoffs' comments on the previous standards list were particularly helpful, especially with respect to the feeling that the first requirements were too tough to be realistic. Thus major revisions are in "Standards.list.8jul08".

INTRODUCTION:

A "module" can be defined as computer code for performing a function or providing on operation. While it might be internally complicated, its number of "connections" to the region external from it are limited.

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.

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 system point of view it may be necessary only to identify externally 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 to the inputs and return the information to the external domain. The external domain can then take the information, along with that from other modules and integrate 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, and more time-consuming and unrewarding under our current granting system to create manuals and tutorials. 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.

Reproducible models

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 success: it is almost impossible to fulfill all the requirements for the "CLass 4" biophysically-based models described there. 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 to test run the models, and none of these have automated unit balance checking, discovering all errors is difficult, even in the hands of their very experienced group. Of the first 87 models for example, 19 do not compile in JSim because of a JSim shortcoming (in handling events from SBML, though JSim does handle events from CellML), and of the 68 remaining only 8 compile as passing unit balance checking. Though there could be errors in the translation program from Biomodels to JSim's MML (mathematical modeling language), and we haven't checked all of the 60 failures, the first 5 of these have definitive unit balance errors.

Errors persist even though the curators are the proponents 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 to

know the time course of the fluxes of Na and K : these are the currents provided as the outputs of the computation. External to the central or inner machinery of the module, 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 concentrations of Na and K on either side of the membrane.

In this scenario, following the description above for modular code, the concentrations of Na and K inside and out, the temperature, and the time course of 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.

References:

Winslow R.L., Rice J., Jafri S., Marban E., and O'Rourke B., Mechanisms of altered excitation-contraction coupling in canine tachycardia-induced heart failure, II: model studies. Circ Res. 1999 Mar 19;84(5):571-86.

Greenstein J.L., Wu R., Po S., Tomaselli G.F., and Winslow, R.L., Role of the Calcium-independent transient outward current Ito1 in shaping action potential morphology and duration. Circ Res, 1026-1033, 2000.

Nicolas Le Novère, Andrew Finney, Michael Hucka, Upinder S Bhalla, Fabien Campagne, Julio Collado-Vides, Edmund J Crampin, Matt Halstead, Edda Klipp, Pedro Mendes, Poul Nielsen, Herbert Sauro, Bruce Shapiro, Jacky L Snoep, Hugh D Spence & Barry L Wanner. Minimum information requested in the annotation of biochemical models (MIRIAM). Nature Biotechnology 23, 1509 - 1515 (2005)