Model Orchestration: addressing the model management challenges of systems biology

Anthony Finkelstein, James Hetherington, Peter Saffrey and Anne Warner

CoMPLEX centre for interdisciplinary research
University College London
London, NW1 2HE
email: a.finkelstein@cs.ucl.ac.uk
{j.hetherington,p.saffrey,a.warner}@.ucl.ac.uk

March 31, 2003

Abstract

Biological modelling is an increasingly complex and diverse field. As well as developing new models for biological phenomena, there is a need to integrate existing models and *scale up* to investigate higher level behaviour. To achieve this integration, techniques and tools are required to catalogue and understand existing models, and to support the development of new models ready for integration. We describe our approach to this problem and validate the approach with examples.

1 Motivation

One of the major challenges of contemporary science is to 'scale-up' our knowledge of micro-level phenomena to yield an understanding of macro-level phenomena. This challenge is particularly evident in biology where our growing knowledge of molecular and cell biology has still to be harnessed in such a way as to give a better understanding of gross physiological issues such as the behaviour of organs. Obviously such an understanding would be important in medicine and drug design.

To address the challenge of scaling up, the principal method used by scientists is the construction of models. From a scientific standpoint the primary problem that has attracted attention is the validity of the models, that is the extent to which the behaviour of the models, and the assumptions the models embed, correspond to the 'real-world' as established by experiment and observation. From a

computational standpoint the primary problem that has attracted attention has been the performance and scaling demands associated with composing fine grain models. This problem fits well with interests in large-scale distributed computing environments and middleware, broadly characterised as GRID architectures.

This paper takes a different stance. It looks instead at the 'managerial' problems inherent in scaling up modelling. The paper envisages a modelling 'environment' in which a wide variety of models are produced within a common domain of interest. These models may be at different levels of abstraction; may deploy different representations; may focus on different, albeit interacting phenomena. Further, the process of modelling and of validation may give rise to model versions and variants that will require management.

The ultimate goal of this work would be the integration and synergy of models addressing phenomena down at the level of individual cell features scaling up through tissue and organ models to a model of a complete organism, such as a human being. Models at every level of this structure could be developed and validated by groups and individuals all over the world using a plethora of techniques mathematical, computational and experimental.

2 Current Approaches

Current approaches to modelling do not take account of the potential plethora of different models nor to how to 'orchestrate' the resulting models - that is how to use them in a synergetic manner. Instead, they assume standalone models or small collections of models with arbitrary handcrafted integration mechanisms. These integration mechanisms are commonly at the program code level. A good example of this approach is the work on the heart carried out by Denis Noble and his team [10]. The models on which this work is based have yielded significant insight and confirmed experimental findings. The complexity and ongoing evolution of this work has however drawn attention to the size of the systems biology challenge and to the pressing limitations of the ad-hoc approach.

The Systems Biology Workbench Project is an example of a response to this [8]. It comprises two distinct components: the Systems Biology Markup Language (SBML) and the Systems Biology Workbench (SBW). SBML is an XML based language for representing biochemical network models. The use of SBML is a significant step forward, potentially making the exchange of models between different tools much simpler. However, the current Level I proposal for SBML does not provide metadata support and SBML does not address model management problems directly. SBW is a software framework that supports the integration of the heterogeneous tools and resources used in biological modelling. This is achieved by way of a relatively low-level message passing and brokering architecture. This tool interoperability constitutes an important practical step, but SBW has nothing to say about the relationships between models nor how

these relationships are managed. The Systems Biology Workbench project is driven forwards by the need to integrate particular tools and models. As such it is a pragmatic step but it does not, nor could it expect to address the larger orchestration problems of systems biology.

The Physiome Project [9] is a leading effort in the area of systems biology. It aims to collect together models categorise them and associate with them a small amount of static metadata. To facilitate this it has developed an XML based language known as CellML. CellML is well designed and makes good use of the XML namespace mechanism in order to integrate with other markup languages such as MathML. It has taken a sensible approach to metadata. Indeed, current proposals for SBML incorporate the CellML metadata definitions. Though CellML is a more 'principled' attack on model-management it is less widely used than SBML. It could be said to fall between two stools — less useful for tool integration than SBML but still taking a limited view of the full scope of the challenge of managing and integrating models. CellML is a step forward but does not in our view constitute a systematic attack on the problems of model management entailed in scaling-up.

Both approaches rely on the relative scarcity of models, the homogeneity of the way in which such models are constructed, and the extent to which the models are 'orthogonal', or at any rate very loosely coupled to each other; an assumption we believe to be false or at least invalid.

As well as modelling approaches, there is significant work in the cataloguing of biological information in online databases. There are now a number of repositories for biological information. These include MEDLINE for medical publications [3], the GenomeNet Database Service [2] for genomic information, BioCyc [1] for pathway/genome information as well as others. Products such as 'Life Science Connect' [4] attempt to provide a front-end for these diverse sources to allow cross queries between them. For raw biological information, these sources are invaluable, but they do not address biological modelling.

3 Work Context

The work described in this paper is taking place as part of a large-scale systems biology project, involved in building an *in-silico* model of the human liver, scaling up the model from the level of gene-expression through individual cells. One of the aims of the project is *vertical and horizontal integration*. This may include, for example, composing together many instances of a model of a single cell (horizontal integration) and incorporating results and insights from this composed model into a separately constructed model of a complete liver lobule (vertical integration)[1].

[1]Note that composing together simple models of individual cells is often trivial, it is scaling up with relation to the large scale behavioural changes caused by, for example, gene expression which presents the greater challenge.

Obviously these ambitious goals motivate the need to organise and integrate models quickly and effectively and therefore our model orchestration work.

In building a fully integrated model of the liver, existing models of various components must be used along with newly devised models constructed using state-of-the-art techniques. Our approach must allow flexibility not only in the growth and evolution of current models of liver function but also in the creation of new models and new modelling techniques. These new techniques may be closely related to existing modelling paradigms, or not related at all. Our approach to the orchestration and integration issues must be developed with these goals in mind, with a view to a more generic contribution to systems biology.

To address the problems of model orchestration we draw on very substantial experience from software engineering on the construction of integrated modelling environments. Building complex software systems requires sophisticated modelling and analysis and presents serious problems of scale and model orchestration. This has been recognised for a considerable period and there are some established design principles and architectures for such environments. During this paper we will outline the use of this experience and expertise in the context of systems biology.

4 Meta-modelling

If you want to understand an activity and ultimately build a software system to support it you need to model it. So, if you want to understand modelling . . . you model it, constructing by way of this process a meta-model. Meta-modelling is the first step in addressing model orchestration.

4.1 Meta-Modelling Requirements

Below we describe the set of informal requirements for model orchestration and therefore our meta-model. These are motivated partly by the needs of the liver project and partly by the strengths and weakness of both CellML and SBML/SBW.

The requirements are divided into two categories: model organisation, how to understand and catalogue existing models and model integration, the challenge of collecting and linking potentially disparate models.

4.1.1 Model Organisation

Understanding and cataloguing Many biological models already exist, and many will be created in the future using a wide variety of ideas, assumptions and paradigms. It should be possible to understand the aims, principles and concepts of each model quickly and how it fits into the overall gamut of biological modelling.

Modularity and Encapsulation A model of any significant size is always built from other constituent models. An orchestration approach must be able to treat a model as a module which can be easily composed with other models. The composition of two models must also constitute a model, which can in turn be composed with other models, if appropriate. The concept of encapsulation, where the details of one model are hidden from another, is also important here.

4.1.2 Model Integration

Linking There are many ways in which models can be linked together. It may be necessary to connect models which are similar or differ greatly in method of construction, level of abstraction or overall aim. An understanding of the features and subtleties of each model should allow many diverse models to be integrated quickly and without unwanted side-effects.

Flexibility/Extensibility As we have described, biological modelling can use a diverse range of modelling paradigms. Our approach must not only incorporate all existing modelling paradigms, but also be extensible enough to accommodate new paradigms which may be applied in future. Ideally, an orchestration approach should be able to deal with any paradigm used in biological modelling, no matter how obscure or unexpected.

Compatibility Existing techniques such as SBML and CellML have already been used to describe a number of existing models and are now known and understood by the biological modelling community. An effective orchestration approach must work alongside and in conjunction with these techniques, rather than as a direct alternative.

Ease of use To appeal to those who build and design models, our approach must be simple and intuitive to use.

Interoperability Our approach must address the need for interoperability between modelling tools.

Dynamic and static models Just as a set of differential equations can be a model of a biological system, so can a diagram or textual representation. An orchestration approach should be able to incorporate these *static* models into an overall framework. Static models will be described in greater detail in section 4.4.

4.2 Meta-model Presentation

Below we set out a meta-model for systems biology modelling. This meta-model presents a very high level 'logical' analysis, constituting the core of a potentially

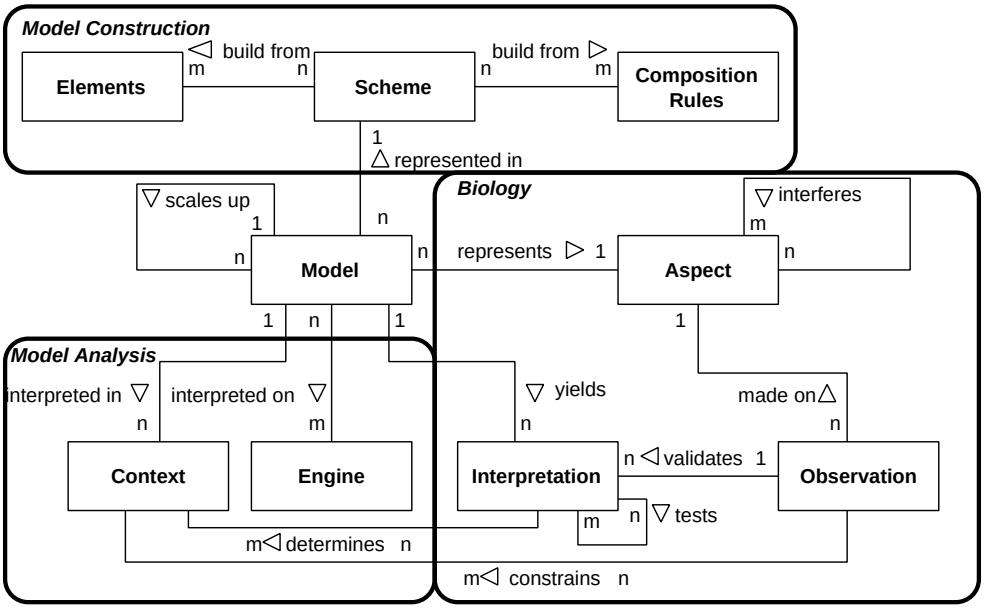

Figure 1: Systems biology meta-model

much more elaborate model. We use a very rudimentary meta-modelling representation, essentially entity-relationship (ER) modelling. The ER approach is very familiar to software developers and there is no scope in context of this paper to provide a detailed tutorial. The diagrams are however quite easy to read and we have used the most stripped down version of the notation. We provide, for scientists, a short account of the key elements. Each box represents an entity class (a class of things (or *objects*). The lines between boxes represent relationships (associations between entity classes). Each entity class, entity for short, may have properties, known as attributes. For simplicity we have not put these attributes on our initial diagram but they are discussed below. Relationships have names given as labels by the lines. The small arrow heads show the direction in which these relationships can be read. Each relationship has a cardinality (or multiplicity) that represents the number of the entity instances that can be tied together by an instance of the relationship; this should become clearer from the discussion below.

The model is set out in Figure 1. Note that the entities are separated into three categories. In traditional modelling, model construction, model analysis and biology (both experimental and theoretical) are carried out mainly by mathematicians, computer scientists and biologists respectively. However, the boundaries between these disciplines are becoming increasingly blurred.

In the remainder of this section, we will describe our meta-model with reference to a simple example.

4.3 Examples

For clarification we have devised a stripped-out biological example, along with how it may be modelled, which we will refer to throughout our descriptions. The example we have chosen is that of an isolated pair of cells communicating solely via a gap-junction, opened on binding with a particular molecule. There are a variety of techniques that could be used to model the behaviour of this system; we have constructed models for the following three:

1. A chemical rate-equation approach, where mass-action kinetics [5] are used to obtain differential equations governing the concentration of each species participating in a reaction.

2. A probabilistic approach, where the behaviours of individual molecules, including a molecule representing the gap-junction, are specified in terms of their relative positions and probabilities to change state.

3. A process algebra approach, where the movement of molecules is modelled by the transmission of discrete messages between processes and each process is represented by a finite state machine. We have chosen to use Promela [7] with its accompanying model checker Spin [6].

The models we have constructed are too small to be of any biological interest, but will serve to illustrate the concepts salient to model orchestration. For brevity, the example will be referred to as 'cell-pair'.

4.4 Models and Aspects

The central concept in our meta-model is, of course, Model. A Model is a description from which detail has been removed in a systematic manner and for a particular purpose. A simplification of reality intended to promote understanding. In systems biology a model may perform many roles - explanatory, exploratory or experimental.

A Model represents, or perhaps seeks to represent, an Aspect. An Aspect can be thought of as a coherent set of properties (or phenomena) of biological interest. In representing an Aspect a model embeds assumptions about these properties; this process of making assumptions is central to modelling.

In our example, the Aspect being modelled is the pair of cells, communicating via the gap-junction. The embedded assumptions contained within this aspect include that the cells can have no other form of communication and indeed that modelling a pair of cells without external influences can provide any biological insight. Many further assumptions may be made in implementing this model, as will be seen in section 4.5.

The three types of model described in section 4.3 are examples of *dynamic models*, models executed to produce results. There is also a notion of a *static*

model such as a diagram, textual description or graph, commonly used to help construct or understand a dynamic model. Although a static model cannot be executed, it is still a model because it constitutes a simplification of reality intended to promote understanding.

A Model can only be relevant to a single Aspect, though as in our example, there may be many Models that represent that Aspect. This is denoted by the cardinality n to 1 given to the relationship represents. We may make Observations on an Aspect, by experiment or otherwise. Such an Observation can only be considered well-formed if it relates to a single Aspect - you may make many Observations on a single Aspect but any Observation can be related to one, and only one, Aspect. The role of observations in understanding and validating models will be discussed further in section 4.6.

Aspects may interfere with one another. In other words sets of properties are not independent (orthogonal) but rather interact. In our example, the aspect is concerned only with the communication between the paired cells, and not any incidental behaviour as a result of this. The species used to form the communication may interact with other Aspects of cell behaviour, which may be important to other models.

4.5 Models and Representations

A Model is represented in a (representation) Scheme or language appropriate to the expression and analysis of properties of particular interest. Such a Scheme is built from Elements which are the meaning-bearing components of the language. The Elements are assembled by way of Composition Rules that describe how the Elements can be composed. A Model can be constructed in a Scheme by using the Elements and Composition Rules. Intra-model relationships — mechanisms that provide structure within a Model — are treated as instances of these Composition Rules.

The three models of the cell-pair each use a different Scheme:

In a differential equation Scheme, the Elements are variables, terms and equations. The Composition Rules are those of mathematics: the connections of variables and terms via arithmetic operators into a system of equations.

An individual-based Scheme also uses variables as Elements, here recording the state and position of each molecule, and the actions governing how a molecule may move or change state. The Composition Rules for this Scheme are how and when the actions are applied.

In the process algebra Scheme the elements are the processes that represent the major entities in the system, in the cell-pair example the cells, and the messages that represent the flow of information between these entities. Communication channels between processes compose these Elements.

4.6 Model Interpretation

A Model is interpreted in a Context. A Context is the data required to produce an instance of the Model, in general the inputs to the Model and the initialisation data for that Model. Clearly there may be many such Contexts in which a Model can be interpreted (1 to n). A Model may, of course, be static and not require a Context. The Model can be interpreted on (or by) an Engine. An Engine is a 'procedure' for generating Interpretations — behaviours — from a Model in a Context. The same Engine may be used for many Models and vice versa (n to m). Again, a static model does not require an Engine.

In the differential equation Scheme, the Context is the initial values for the rate constants and concentrations of each species in each cell. The Engine in this case is the extremely rich background of techniques and tools, both simulation based and analytical, to manipulate and reason about differential equations. The simplest Interpretation of this Scheme would be the change in species concentration over time, displayed as raw data, or as a chart. This data could be further interpreted to provide, for example, a period of oscillation of concentrations, or the length of time until an equilibrium is reached. Our example is relatively simple, but there may still be scope for more advanced mathematical analysis, such as the identification of bifurcations, phase-transitions, boundary values and so on.

In the individual-based Scheme, the Context is the initial states of each molecule and the probabilities for various state changes, such as a molecule binding to cause a gate opening, or molecule movement. The engine could be some algorithm to update the states of each molecule at each time step, based on the probabilities, most likely implemented as a piece of computer software. Analytical techniques also exist for individual-based methods. As with the differential equation Scheme, the Interpretation of this Scheme could be the concentrations of molecules in each cell over time. It may also be possible to derive other values such as the probability of gate opening/closing in relation to the concentrations.

For the process algebra Scheme the Context will be initial values for the concentrations of species in each cell. The Engine will depend on the particular process algebra used, but in the case of our example is the model checker Spin. The Interpretation used here may depend on the verification of temporal properties, to identify the existence or absence of behaviours of interest. Simulations are also possible, and repeated random simulations may be used to derive probabilities for various concentrations, in the manner of a Monte Carlo approach.

An observation may validate a model only in some particular Context; it may constrain the Context. This can lead to the 'inversion' of Context and Interpretation, where we may wish to use a previous Interpretation as a Context to analyse how certain results are obtained.

The interpretation of one model may determine the context for another. For example, an experimental observation of a diffusion constant, which constrains

a partial-differential-equation model diffusion, is used, together with the application of an analytical (i.e. non computational) engine to the individual based model of diffusion to obtain the diffusion-probability-per-unit-time parameter of the individual based model.

Some engines are restricted to analysing a model only in a given context. Other techniques, such as the powerful analytical techniques of bifurcation theory in dynamical systems, or the application of computing power to scan over or sample from the possible contexts of a model, allow one to examine a model in a multiplicity of contexts, and to explore the structure of the space of contexts of a model.

A model Interpretation is largely what motivates the construction of a model in the first instance. The Interpretation may lead to further models to examine new phenomenon uncovered, or to further experimental investigation.

In addition our objective is always to ensure that the Interpretations are biologically meaningful and relate to the Observations. The Observations serve to test and possibly invalidate the Interpretations, or indeed the whole model. Of course with many possible Interpretations we can use one Interpretation to test another. In the cell-pair example, we could, for example, compare the flow rates derived from the individual based model to the differential equation model.

4.7 Scaling up

A Model scales up a number of 'lower-level' Models. The concept of scale deserves some examination. It is a heavily overloaded term with many meanings. In biology it is often used to mean two distinct things - 'extent' and 'level of scrutiny'. Thus an organ like the liver has greater spatial extent than the lobules that compose it. Processes within it have greater temporal extent than those within the cells that make it up. The level of scrutiny is, by contrast, the granularity at which a system is viewed. For the liver, in terms of components of cells (gap junctions and so on), in terms of cells or in terms of lobules. The potential for confusion is obvious. In general, phenomena with large extent are best viewed in terms of relatively coarse grain elements. If not, the models themselves become large and awkward to build and analyse. Further, phenomena of significant extent may be relatively insensitive to changes in fine grain elements and it may makes sense to build coarse grain models to determine gross behaviour. We use the term scale to denote level of scrutiny alone and in a manner directly analogous to the way in which computer scientists use the term abstraction.

If we have Models representing the same Aspect but in different Schemes or at different levels of scrutiny, we use the scales up relationship. Scales up provides inter-model structuring. This distinction between intra-model and inter-model structuring may seem complex but is key to our meta-model: without making a distinction between relationships inside and outside a particular model, it would not be possible to understand a model as a distinct component, or to integrate it

with other models, which may not necessarily be represented in the same Scheme or model the same Aspect.

5 Implementation

The meta-model we have outlined is a significant first step toward providing a comprehensive environment for scaling modelling in systems biology. The construction of such an environment will be a major task and we intend to approach it incrementally. To further validate our meta-model we will implement it as a set of XML-based languages. These XML languages are intended as a layer associated with existing approaches in each area.

With our approach, each model has associated with it an XML description. This describes the model in terms of its Scheme, Aspect and the other features described in section 4. It will also contain a variety of meta-data, including information such as literature references and those groups or individuals responsible for the model.

A scaled model can be described with a hierarchy of XML documents. The top level provides the most general description of the Aspect and aims of a particular model. Below this, further XML documents refine this into the details of assumptions made, implementation decisions and other specifics. In this way, any model can be understood at a variety of levels of abstraction and in the context of other models closely related and associated with it.

Analysis of an the XML description for a model can provide high level information on the aims and principles behind the model as well as more detailed information on equations used, assumptions made and tools employed. The use of XML makes this information easy to index, search, display and cross-reference with existing tools. An XML description can eventually be converted into input files for simulation and analysis tools in various modelling paradigms.

XML is ideal for our purposes because it is flexible, extensible and allows easy embedding of data in other formats. A model described with our approach could contain a description of that model using SBML or CellML simply using the XML namespace system. Links to data sources and executable models can be included using XLinks.

6 Conclusions and Further Work

We have described our method for tackling the understanding and integration of biological models, with a view to integration. We have presented an example as viewed by our approach and demonstrated how it can be understood with reference to three separate modelling paradigms. We have also discussed the implementation of our approach, and ensuring compatibility with other generic

modelling schemes.

As part of our work on modelling the liver, we are developing a prototype model integration framework, based on the approach described in this paper. The framework aims to explore the effectiveness of our approach in a real world setting particularly when used by model builders themselves. At present, the framework is aimed largely at the modularisation and integration requirements within a single modelling Scheme (differential equations), but extensibility and the later inclusion of other modelling Schemes is a strong consideration in development.

7 Acknowledgements

This work was carried out with funding from the Department of Trade and Industry under their Beacon initiative.

References

[1] Biocyc knowledge library. `http://www.biocyc.org/`.

[2] Genomenet database service. `http://www.genome.ad.jp/`.

[3] National library of medicine online database. `http://www.ncbi.nlm.nih.gov/PubMed/`.

[4] Solutions for life science it. `http://www.equait.com/index.html`.

[5] P.W. Atkins. *Physical Chemistry*. Oxford Univesity Press, 1998.

[6] Gerard Holzmann. The model checker spin. *IEEE Transactions on Software Engineering*, 23(5):279–295, May 1997.

[7] Gerard Holzmann. *Promela Language Reference*. Bell Labs, MAY 1997.

[8] M. Hucka, A. Finney, Herbert M. Sauro, H. Bolouri, J. Doyle, and H. Kitano. The erato systems biology workbench: Enabling interaction and exchange between software tools for computational biology. In *Pacific Symposium on Biocomputing 2002*, pages 450–461, January 2002.

[9] Peter J. Hunter and Thomas K. Borg. Integration from proteins to organs: the physiome project. In *Nature Reviews Molecular Cell Biology 4*, pages 237–243, March 2003.

[10] Denis Noble and Yoram Rudy. Models of cardiac ventricular action potentials: iterative interaction between experiment and simulation. *Philosophical Transactions: Mathematical, Physical & Engineering Sciences*, 359(1783):1127–1142, 2001.