

Original research

BioModel engineering for multiscale Systems Biology

Monika Heiner^{a,*}, David Gilbert^b^a Department of Computer Science, Brandenburg University of Technology, Postbox 10 13 44, 03013 Cottbus, Germany^b School of Information Systems, Computing and Mathematics Brunel University, Uxbridge, Middlesex UB8 3PH, UK

ARTICLE INFO

Article history:

Available online 12 October 2012

Keywords:

BioModel engineering
 Multiscale modelling
 Systems Biology
 Synthetic Biology
 Biomolecular networks
 Coloured qualitative/stochastic/continuous/
 hybrid Petri nets
 Model checking

ABSTRACT

We discuss some motivational challenges arising from the need to model and analyse complex biological systems at multiple scales (spatial and temporal), and present a biomodel engineering framework to address some of these issues within the context of multiscale Systems Biology. Our methodology is based on a structured family of Petri net classes which enables the investigation of a given system using various modelling abstractions: qualitative, stochastic, continuous and hybrid, optionally in a spatial context. We illustrate our approach with case studies demonstrating hierarchical flattening, treatment of space, and hierarchical organisation of space.

© 2012 Elsevier Ltd. Open access under [CC BY license](http://creativecommons.org/licenses/by/3.0/).

1. Motivation

BioModel Engineering. Biology is increasingly becoming an informational science. This revolution has been driven by technological advances which have supported the development of studies at many levels of intra- and intercellular activity. These advances have facilitated the analysis of how the components of a biological system interact functionally – namely the field of Systems Biology (Hood, 2003).

At the heart of this field lies the construction of models of biological systems, see Fig. 1. These models are used for description of acquired understanding, or analysis which should ideally be both *explanatory* of biological mechanisms and *predictive* of the behaviour of the system when it is perturbed by, e.g., mutations, chemical interventions or changes in the environment. Furthermore, models can be used to help make genetic engineering easier and more reliable, serving as design templates for novel synthetic biological systems – an emerging discipline known as Synthetic Biology (Endy, 2005, Heinemann and Panke, 2006). Central to both Systems and Synthetic Biology is BioModel Engineering (BME) which is the science of designing, constructing and analysing computational models of biological systems (Breitling et al., 2010).

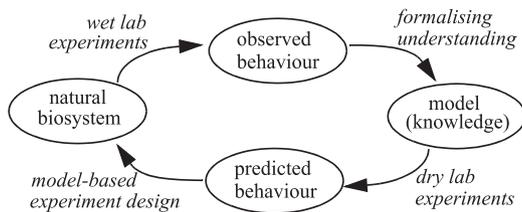
Modelling means abstraction. A general discussion of the relationships between computational modelling and abstraction can be found in (Melham, 2013). We can vary the degree of abstraction and the specific information abstracted away from along each of the following dimensions.

1. *Hierarchical organisation of components* – like molecules, organelles, cells, tissues, organs, organisms, i.e., the abstraction levels of the model objects are chosen as appropriate; indeed several levels can be mixed within one model.
2. *Function* – the model operations (atomic events) can be abstracted to their essential effects and can refer to a wide variety of events. In the most abstract case the relationship between objects being modelled can be reduced to *interactions*, e.g., between genes or proteins. More detailed descriptions can be given at the level of, e.g., chemical reactions with precise or abstract stoichiometries, or conformational change of a protein, transport of a molecule, etc.
3. *Granularity of description* – i.e., the resolution within a particular level of abstraction, depending on the completeness of our knowledge within one level.
4. *Time* – governs everything; however, we may abstract away from time to obtain simplified qualitative models with corresponding analysis, specifically if insufficient kinetic information is available. This kind of abstraction typically yields a conservative approximation of the behaviour, and considers more behaviour than is actually possible under some given timing constraints.

* Corresponding author.

E-mail addresses: monika.heiner@informatik.tu-cottbus.de (M. Heiner), david.gilbert@brunel.ac.uk (D. Gilbert).

Systems Biology: modelling as formal knowledge representation



Synthetic Biology: modelling for system construction

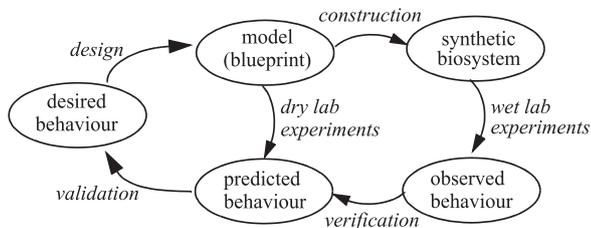


Fig. 1. The role of formal models in Systems and Synthetic Biology, adapted from (Heiner and Gilbert, 2011).

validated – does the model indeed behave as expected? Deviations of expected and observed behaviour may be caused by bugs in the model or the software tools used. Never blindly trust numerical simulation results.

Multiscale modelling in Systems Biology is the field of solving physical problems which have important features at multiple scales, particularly multiple spatial and/or temporal scales. It goes far beyond the traditional approach of modelling at just one spatial/temporal scale. The specific computational challenges caused by multiple scales and the state of the art of potential solutions how to bridge the gaps are reviewed in (Qu et al., 2011).

Until now most models have been at the intracellular level, and indeed have largely ignored locality within the cell; however there is a need to increase specifically the spatial scope of models of biological systems to enable descriptions at the intercellular (cell–cell), tissue, organ and even whole organism scales. The motivation has come both from the increasing need for life scientists to use computational models to facilitate the investigation and understanding of multicellular systems, and the greater variety of data available at different scales.

The challenges for modellers include the development of suitable paradigms and associated tools to create coherent descriptions of biological systems by integrating several spatial scales, and methods for the simulation, analysis, and checking of the models in order ultimately to use them to predict the behaviour of the biological system when disturbed by e.g. mutations, drugs or stress; for an introductory illustrative example see Fig. 2 which is discussed in some more details in Section 6.

More specifically, the scenarios introduced by modelling biological systems beyond one spatial scale, which need to be addressed, include:

5. *Individual versus population behaviour* – requiring a choice between the stochastic versus deterministic modelling paradigm. The latter considers the average case, smoothing out any stochasticity. The averaged behaviour might be representative for a population, but may ignore special behaviour triggered by rarely occurring stochastic events.
6. *Space* – either hierarchical organisation of, for instance, cells in a tissue, or measured space at different resolutions, yielding a distance measure.
7. *Shape* – of, for instance, molecules; e.g., regarding protein structure, protein folding is reduced to the different states (conformations).
8. *Volume* – even if the system behaviour evolves in space, we may abstract away from the volumes of individual molecules and consider the number of molecules per space position instead.
9. *Observables* – often we abstract away from the phenotype and treat the underlying molecular mechanism as the read-out (observable), see, e.g., the *Drosophila* wing case study briefly discussed in Section 6. This contributes to bridging the gap between models and observations; see Section 5 on model checking.
10. *Biosystem dynamic development* – this is considered in developmental biology; however model structure in Systems Biology is often currently assumed to be fixed.

There are two obvious consequences for sound engineering of models.

- (1) *All assumptions underlying a model need to be explicitly stated.* Generally, a model and the derived conclusions are only valid as long as the underlying assumptions are justified.
- (2) *Models need to be validated before they are used for behaviour prediction.* Modelling and programming have many things in common. Both require abstract reasoning and produce condensed descriptions of behaviour which may not always coincide in all aspects with the intended one. Thus, modelling should be done – as programming – with great care. Models need to be developed step-wise with slowly increasing model size and complexity. Each modelling step should be carefully

1. *Repetition of components* – e.g. the need to describe multiple cells each of which has a similar definition.
2. *Variation of components* – sets of similar components with defined variations, e.g. mutants.
3. *Organisation of components* – e.g. how cells are organised into regular or irregular patterns over spatial networks in one, two or three dimensions.
4. *Communication between components* – in general communication is constrained to occur between immediate neighbours, but this may be further constrained according to the relationship between neighbours, and the position of a component within a spatial network.
5. *Mobility/Motility of components* – e.g. transport of components within a system, or actively motile cells.
6. *Hierarchical organisation of components* – enabling the description of (possibly repeated) components which contain repeated sub-components; for example, cells containing several compartments. This feature enables the use of abstraction regarding the level of detail used to describe components.
7. *Replication of components* – e.g. cell division.
8. *Deletion of components* – e.g. cell death.
9. *Irregular/semi-regular organisation of components* – for example a not exact honeycomb grid.
10. *Dynamic grid size* – for example alter size and/or topology of grid to model development. Also required for ability to insert/remove items.
11. *Differentiation of components* – for example, differentiation of embryonic stem cells or immune cells makes a less specialised cell more specialised.
12. *Pattern formation of components* – organising a number of cells in appropriate one, two or three dimensional structures in space and time.

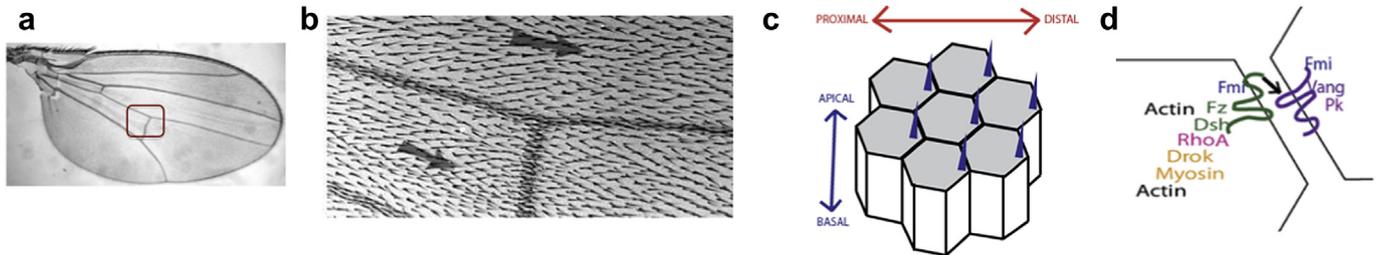


Fig. 2. Planar Cell Polarity, *Drosophila*. (a) Wing. (b) Wing tissue with hairs. (c) Epidermal cell arrangement. (d) Inter- and intracellular signalling cartoon.

Components could be molecules, organelles, cells, tissues, organs, organisms.

A drawback of current modelling approaches are their limitation to relatively small networks. Biological systems can be represented as networks which themselves typically contain regular (network) structures, and/or repeated occurrences of network patterns. This organisation occurs in a hierarchical manner, reflecting the physical and spatial organisation of the organism, from the intracellular to the intercellular level and beyond (tissues, organs, etc.).

Although such network models can be designed using standard modelling approaches, so far there is no dedicated support for such structuring; it becomes impractical as the size of the networks to be modelled increases. Besides the purely technical aspect due to the impracticality of handling large flat networks, humans need some hierarchy and organisation in what they are designing in order to be able to conceptualise the modelled object. Thus, models should explicitly reflect the hierarchical organisation in complex biological systems.

2. Representation formalism – just a matter of taste?

One of the well-established design principles in engineering is to keep everything as simple as possible. This raises the question: Do we actually need different formalisms for different abstraction levels and/or different scales? In the following we summarise our criteria which a modelling formalism should ideally fulfill.

- **Readability**, having an intuitive representation to make model descriptions easily comprehensible. Readability is often the key to fault-avoidant model construction.
- **Unambiguity**, having a formal semantics, uniquely defining derived descriptions required for some established analysis techniques such as stoichiometric matrix, stochastic reaction networks, or Ordinary Differential Equations (ODEs). Note, however, that ODEs are not a suitable core representation because the corresponding reaction network is only uniquely defined under specific conditions, see (Soliman and Heiner, 2010, Heiner and Gilbert, 2011) for more detailed discussions.
- **Abstraction**, allowing the unambiguous representation of various types of biological processes at different levels of abstraction with appropriate resolution of detail in the same model, ranging from the conformational change of a single molecule to the macroscopic response of a cell, the development of a tissue, or even the behaviour of a whole organism. (Marwan et al., 2011).
- **Local context**, so that an event only causes changes in its immediate environment. This allows us to read and understand a model by going through all its local effects, and supports efficiency gains in some analysis algorithms, which are otherwise not possible.
- **Causality**, permitting reasoning about the order of occurrence of events in terms of “what has to happen first, before something else can happen afterwards”.

- **Concurrency**, an inherent property of biochemical processes, requiring a clear distinction to be made between alternative and concurrent behaviour, see (Heiner et al., 2004).
- **Compositionality**, enabling the construction of models by the composition of smaller modules, preserving essential properties. This facilitates the establishment of module libraries and their reuse in automated model construction. One aim here is the use of models as designs in Synthetic Biology, which should ultimately enable the compilation of genetic components from design descriptions, and the automatic construction of the desired biosystems.
- **Executability**, the visualisation of which allows the modeller to better comprehend the net behaviour, and facilitates the communication between wet lab experimentalists and dry lab (computational) theoreticians.
- **Analysability**, supporting the wide range of established analysis techniques, comprising both static and dynamic analyses including model checking in various paradigms. Standard analysis techniques, which are commonly used in many Computer Science approaches, need to construct the partial or full state space (expressed as LTS or PO prefix). Complementary to these dynamic analysis techniques, static analysis techniques might be of help, which permit the decisions of general behavioural properties without state space construction, usually.

In addition, modelling biosystems at multiple scales requires:

- **Notions of space** in 1, 2 and 3 dimensions (1D, 2D, 3D) – and possibly over the 4th dimension of time (spatial models which describe system development over time).
- **Hierarchical organisation** in terms of both physical and conceptual levels.
- **Hybrid descriptions** ranging from individuals to populations, and also over different time-scales.

Just as with uniscale modelling, multiscale modelling formalisms need to be supported by appropriate model construction, execution and analytical techniques.

How dynamic has a model to be. Model modifications over time may include:

- addition/subtraction of model components,
- rewiring yielding new structures,
- parameter modification (e.g. triggered by mutation),
- model translocation (model passing, nets in nets),
- reorganising the hierarchical structure, adding, removing levels; e.g. there is re-organization on the organ level in insect development from larva to adult, the metabolism may change as well.

Remark: different stages in the life cycle of an organism can involve different sets of genes that are transcribed into proteins, thus changing the resulting biochemical networks over time.

Currently we wish to focus on modelling the dynamic behaviour of biological systems, with an emphasis on the underlying mechanisms, for example gene transcription networks, metabolic pathways and signal transduction networks. Thus we do not consider – for the time being – systems which evolve over time and their corresponding models, for example developmental pathways, because of the computational challenges involved in designing, simulating and analysing models in which the topology or kinetics exhibit temporal changes.

3. Framework

The following framework can be realised using a variety of modelling approaches; see (Melham, 2013) for a review of possible candidates. We prefer to employ Petri nets because they are a natural and established notation for describing biochemical reaction networks both share the bipartite property: species and reactions form two types of nodes in biochemical networks which can be mapped onto places and transitions in Petri nets.

Petri nets can be used to perform all major modelling and analysis approaches central to Systems and Synthetic Biology. They may serve as an umbrella formalism integrating qualitative and quantitative (i.e. stochastic, continuous, or hybrid) modelling and analysis techniques; see (Heiner and Gilbert, 2011) for a fuller treatment, (Heiner et al., 2008) for related formal definitions of the core terminology by means of a running example from Systems Biology, and (Gilbert et al., 2008) for an example of application to Synthetic Biology.

In the following we present a framework, see Fig. 3, to give a formal structure over qualitative and quantitative descriptions of biochemical networks using Petri nets incorporating individual and population-based scenarios in a hybrid manner. The framework comes with associated analytical techniques including model checking, which we focus on later.

Interaction paradigm (\mathcal{IN}): (for the sake of readability not included in Fig. 3) the most abstract representation of a biochemical network is at the level of the relationships between the objects being modelled i.e., their interactions. These are typically represented by undirected graphs. All other paradigms discussed below are at a more detailed level, and include specific information about the reactions. \mathcal{IN} can be derived by abstraction from reaction networks, but not vice-versa. In Fig. 3, the \mathcal{IN} node could be placed

in the centre with a pair of opposite abstraction/extension arcs going to all other nodes.

Qualitative paradigm (\mathcal{QPN}): the most abstract representation of a biochemical reaction network is qualitative and is minimally described by its topology, usually as a bipartite directed graph with nodes representing biochemical entities and reactions, or in Petri net terminology *places* (represented as circles) and *transitions* (represented as boxes), respectively. Arcs can be annotated with stoichiometric information. The qualitative description can be further enhanced by the abstract representation of discrete quantities of species, achieved in Petri nets by the use of *tokens* at places. These can represent the number of molecules or the level of concentration of a species, or simply the presence of, e.g., a gene. A particular arrangement of tokens over a network specifies the current system state. The state of the system changes by the firing of transitions. A transition can fire if all its pre-conditions are fulfilled, i.e., its pre-places carry enough tokens. Upon firing of a transition, tokens from all its pre-places are removed, and tokens are added to all its post-places, each according to the corresponding arc weights.

Fig. 4a) shows a small, but realistic Petri net as an introductory example. The presence of one gene allows the generation of proteins without consuming the gene (each firing of *generate* adds a token to *protein*), while generated proteins can degrade (each firing of *degrade* subtracts one token from *protein*). This basic behaviour can be extended by allowing the gene to be blocked by another protein, which makes a building block called gene gate; see Fig. 4b). When genes repress each other in a circular manner, we obtain a gene regulatory cycle – the repressilator (Blossey et al., 2008); see Fig. 5, Fig. 6. More details regarding the definition and semantics of Petri nets can be found in (Heiner et al., 2008).

The behaviour of such Petri nets forms a discrete state space, which can either be captured as (1) a Labelled Transition System (LTS) to describe the net behaviour by all (totally ordered) interleaving sequences in the style of transition-labelled automata (interleaving semantics), or as (2) a maximal branching process to describe the net behaviour by all partially ordered transition sequences (partial order (PO) semantics). Animating a Petri net by sequentially firing individual transitions generates a path through the LTS.

Both LTS and PO descriptions of behaviour can be analysed for the purpose of model verification. If the state space is finite, this is

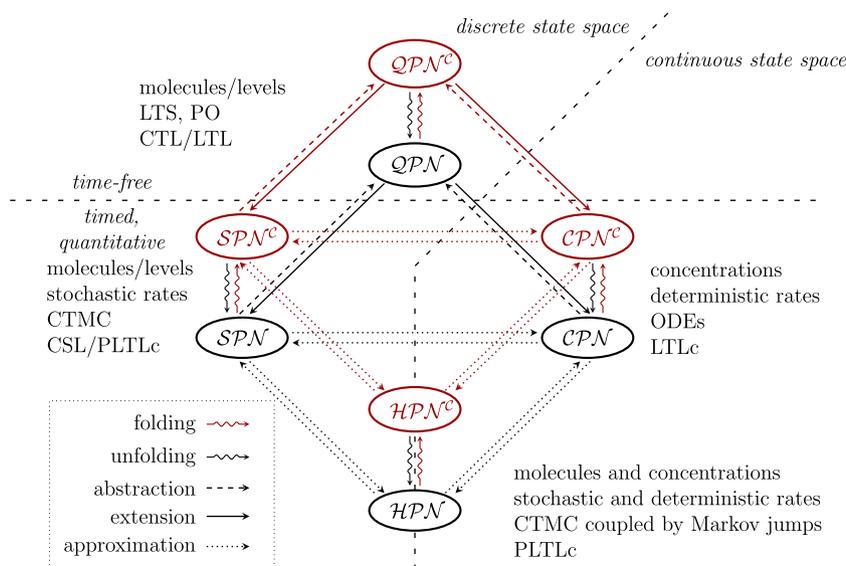


Fig. 3. A unifying framework integrating various abstraction levels, adapted from (Heiner et al., 2012).

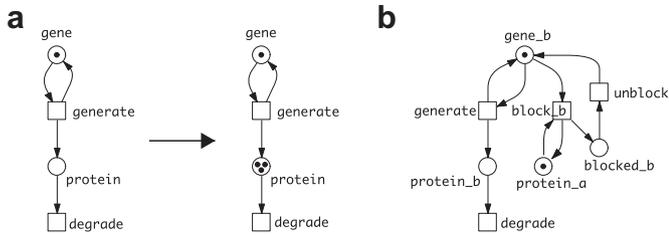


Fig. 4. Introducing Petri nets. **a)** The system state on the right is reached after firing a transition sequence containing **generate** exactly three times more than **degrade**. **b)** A gene gate according to [14]: gene *b* may be blocked by protein *a*.

best done using model checking techniques, where the properties of interest are expressed by, e.g., a branching time temporal logic, one instance of which is Computational Tree Logic (CTL) (Clarke et al., 2001), or a linear-time logic (LTL) (Pnueli, 1977).

The standard semantics for *QPN* do not associate a time with transitions or the sojourn of tokens at places, and thus these descriptions are time-free. The qualitative analysis considers however all possible behaviour of the system under any timing. Thus, the *QPN* model itself implicitly contains all possible time-dependent behaviours.

Timed information can be added to the qualitative description in two ways – stochastic and continuous.

Stochastic paradigm (SPN): preserves the discrete state, but in addition associates an exponentially distributed firing rate (waiting time) with each reaction. The firing rates are typically state-dependent and specified by rate functions. All reactions, which occur in the *QPN*, can still occur in the *SPN*, but their likelihood depends on the probability distribution of the associated firing rates. Thus all qualitative properties valid in the *QPN* are also valid in the *SPN*, and vice versa. The underlying semantics is a Continuous-Time Markov Chain (CTMC), and stochastic simulation generates a random walk through the CTMC. For example, assigning rates to any of our repressilator Petri nets (see Blossey et al., 2008) for suitable parameters) generates sustained oscillation for all proteins, with each run behaving differently.

Special behavioural properties can be expressed using, e.g., Continuous Stochastic Logic (CSL), a stochastic counterpart of CTL which was originally introduced in (Aziz et al.) and extended in (Katoen et al.), or PLTLc, a probabilistic extension of LTL with constraints (Donaldson and Gilbert, 2008).

Continuous paradigm (CPN): replaces the discrete values of species in the *QPN* or *SPN* with continuous values, and hence is not able to describe the behaviour of species at the level of individual molecules, but only the overall behaviour via concentrations. Timed information is introduced by the association of a particular deterministic firing rate with each transition, permitting the continuous model to be represented as a set of ODEs which are typically non-linear, requiring numerical analysis methods. Unlike

in the *SPN*, the concentration of a particular species in such a model will have the same value at each point of time for repeated computational experiments. The state space of *CPN* models is continuous and linear, and can be analysed by, for example, using Linear Temporal Logic with constraints (LTLc) e.g., (Calzone et al., 2006).

Moving between stochastic and continuous paradigms. The same quantitative model can be read either stochastically or continuously, no changes being required (up to some scaling in the rate functions for higher order reactions). In (Gilbert et al., 2007) we discuss in more detail how the stochastic and continuous models are mutually related by approximation. However, such a purely syntactic translation will not always preserve the behaviour; e.g. our stochastic repressilator will lose its oscillation when considered continuously, which calls for the hybrid paradigm, or a much more sophisticated derivation of the continuous counterpart; see (Blossey et al., 2008) how to do it for the repressilator.

The qualitative and stochastic models consider all possible behaviour under any timing, whereas the continuous model is constrained by its inherent determinism to consider a subset. This may be too restrictive when modelling biochemical systems, which by their very nature exhibit variability in their behaviour. Thus, moving between stochastic and continuous paradigms may come along with counter-intuitive effects, see (Heiner and Gilbert, 2011) for some examples.

Hybrid paradigm (HPN): combines all features of *SPN* and *CPN*. It is specifically useful for the description of systems which are characterised by multiple temporal scales (extremely stiff systems), where reactions with low rates are considered stochastically, and those with high rates are considered continuously. Another use is when stochasticity is crucial for the total system behaviour, as it is the case for the repressilator example. To preserve oscillation we need to keep the (discrete) on/off switch of the genes, while the number of proteins can be continuously approximated, if the numbers become too high. In summary, *HPN* provide a trade-off between computational costs and accuracy of the computed results (Herajy and Heiner, 2012).

Colouring: can be applied both to interaction networks as well as to reaction networks. Colouring of interaction networks permits the description of repeated interactions within a spatial context (i.e., at the most abstract level – repeated objects in space) which can also be refined to give a notion of location.

Colouring of reaction networks yields a form of high-level Petri nets which permit the description of similar network structures in a concise way using colours grouped in colour sets (synonym for discrete data types) to describe repeated elements. This allows for the discrimination of species (molecules, metabolites, proteins, secondary substances, genes, etc.). In addition, colours can be used to encode locality, for example to distinguish between sub-populations of a species in different locations (cytosol, nucleus and so on). The colouring principle can be equally applied to

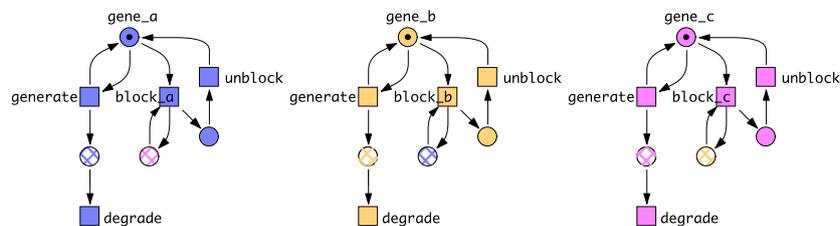


Fig. 5. The repressilator (Blossey et al., 2008), Petri net for three genes in a regulatory cycle represented using logical nodes (here places, cross-hatched) to preserve gene-centred modules. Logical nodes with identical names serve as connectors; they are multiple representations of the same node used for layout clarity. See Fig. 6 for an alternative representation without logical nodes.

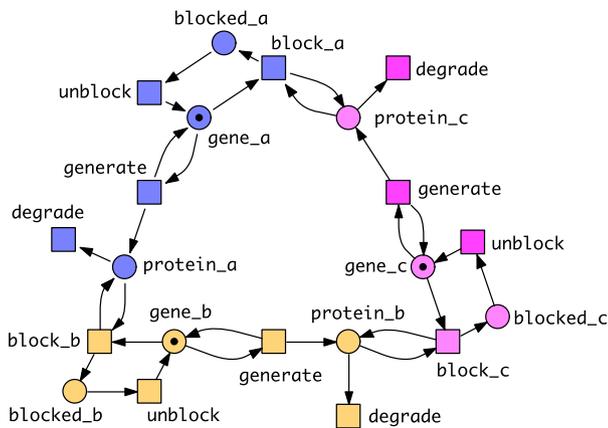


Fig. 6. Alternative representation of the Petri net in Fig. 5 by fusion of logical places with the same name.

qualitative, stochastic, continuous and hybrid Petri nets; we denote the coloured counterparts by QPN^C , SPN^C , CPN^C , and $HCPN^C$, respectively.

Coloured Petri nets can be constructed from uncoloured Petri nets by folding, when partitions of places and transitions are given. These partitions define the colour sets of the coloured net. For illustration, we fold the Petri net given in Fig. 5 and 6 with *ColourSet GeneSet* = {a,b,c}, see Fig. 7. An attractive advantage of this representation is its scalability; changing the number of genes involved in the regulatory cycle just requires to adapt *GeneSet* appropriately. Vice versa, coloured Petri nets with finite colour sets can be automatically unfolded into uncoloured Petri nets, which then allows the application of all of the existing powerful standard Petri net analysis techniques. For example, unfolding of the coloured Petri net in Fig. 7 generates the Petri net in Fig. 6 (with some fine-tuning of the automatic layout).

Hierarchically coloured Petri nets ($HCPN$): impose a hierarchical structure over colours to reflect the hierarchy inherent in a system being modelled. Hierarchically structured colouring brings, e.g., abstraction (at physical levels) over network motifs. We can (trivially) obtain locality from colour; however the true

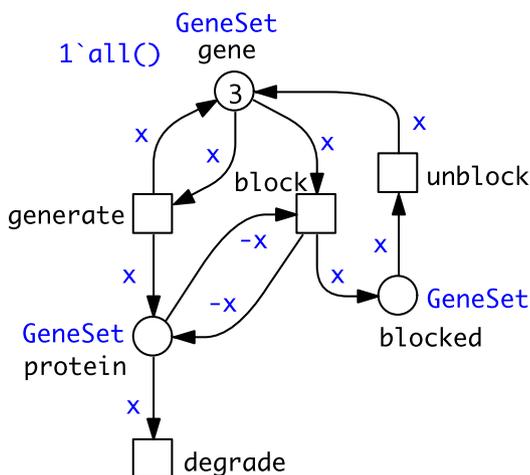


Fig. 7. Alternative representation of Fig. 5 and 6 by folding of similar subnets into a coloured Petri net with *ColourSet GeneSet* = {a,b,c}, x is a variable of type *GeneSet*, $-x$ refers to the (modulo) predecessor in *GeneSet*, and $1`all()$ specifies the initial marking as one token of each colour in *GeneSet*. The number 3 in place *gene* gives the total amount of tokens.

integration of hierarchy and colour brings hierarchical locality which enables us to construct multiscale models, see (Gilbert and Heiner, 2011) for more details, and (Gao et al., 2012) for a detailed example, briefly discussed in Section 6.

Tools and technology. BioModel Engineering of non-trivial case studies requires adequate tool support. We deploy a sophisticated toolkit covering the whole reaction network framework:

- **Snoopy** (Rohr et al., 2010, Heiner et al., 2012) is a platform to support the construction and animation/simulation of all the types of Petri nets discussed above, with an automatic conversion between them. Obviously, there may be a loss of information in some directions (cf. arrows labelled with *abstraction* in Fig. 3). The conversion between coloured and uncoloured net classes involves user-guided folding or automatic unfolding. Snoopy supports several data exchange formats, among them to the following analysis tools in this list, as well as SBML import/export, which opens the door to a bunch of tools popular in Systems and Synthetic Biology.
- **Charlie** (Wegener et al., 2011) permits the analysis of standard properties and techniques of Petri net theory, expanded by explicit CTL and LTL model checking.
- **Marcie** (Schwarick et al., 2011) is a symbolic analysis tool of standard Petri net properties, and CTL model checking for QPN and CSL model checking for SPN . Exact analyses are complemented by approximative PLTc model checking built on fast adaptive uniformisation and distributed Gillespie simulation.
- **MC2(PLTLc)** (Donaldson and Gilbert, 2008) is a Monte Carlo Model Checker for properties written in (PLTLc). MC2(PLTLc) can operate with stochastic/deterministic simulation output, deterministic parameter scan output or even wet lab data.

The Petri net tools are publicly available at <http://www-dssz.informatik.tu-cottbus.de>, and MC2(PLTLc) at <http://www.brc.dcs.gla.ac.uk/software/mc2/>.

4. Algorithmic model construction

In this section we discuss how to obtain models of biological systems in an algorithmic manner, as opposed to by hand, and specifically explore the particular challenges associated with constructing multiscale models. Humans have great strengths in developing conceptual models, giving meaning to models and relating them to the behaviours of known biological systems. The advantage of an automated approach is that it facilitates exploration of model space, permitting the discovery of models of natural systems which are unexpected or as yet unknown, for example two organisms in a mutualistic or parasitic relationship, or of those which do not exist in nature, which could serve as a blueprint for the design and construction of novel synthetic biosystems. Manual construction is a very limiting approach in that, for example, it is very difficult to manually perform multiparameter optimisation or to create all possible topologies even from a small model.

The general challenge in algorithmic construction is the inverse problem – deriving information, often a model, about a system from observations of its behaviour. There is a large body of work regarding the construction of models of biochemical systems from data, see e.g. (van Riel, 2006) for a review. Data can be derived from many individual small-scale experiments ('bottom up' approach) or from large high-throughput data sets ('top down').

There is substantial body for uniscale scenarios, but none for multiscale. Algorithmic reconstruction of multiscale models is more challenging than for uniscale models because of various factors:

- the target behaviours (e.g. observations) may be at a different level to that of the part of the model that is being fitted, and thus may be of different types (observing the orientation of epithelial hairs in Planar Cell Polarity in *Drosophila* wing, but attempting to construct the underlying intra-cellular and inter-cellular biochemical model),
- the temporal or spatial scale of observations can be different to that of the model – for example there may be much longer time intervals for physiological changes at a high level compared with the time scale at the biochemical level; in the extreme there may be only one observation – for example the final physiological state.

In general, there are three aspects of uniscale models that require fitting during construction: (1) the model structure, i.e. the players and their relationships, (2) concentrations of the players at least at some initial time point, and (3) the kinetics. A modular approach to model construction has proved to be promising at the uniscale, for example (Wu et al., 2012) who use an evolutionary algorithm to select and compose pre-defined building blocks from a library of atomic models, and employ a global optimisation algorithm to fit the kinetic rates in order to produce a Continuous Petri Net model. In the multiscale case there may be different models for different spatial scales, and then there is the additional challenge of fitting the functions that interface between these models.

5. Model checking over multiscale models

Model checking is a well established technique applied to uniscale models and has been used for model validation, to drive model construction and to generate predictions from models. Additionally it can be used to perform model comparison and model searching (Heiner and Gilbert, 2011). In this section we discuss the challenges encountered when extending model checking to multiple scales when space and hierarchy are involved. Although there are many techniques for model analysis, all of which need to be developed for multiscale models, we focus here on model checking because it relates the observed behaviour or desired behaviour of the target system to the behaviour of the model. This is an essential first step because models need to be validated before performing model based behaviour prediction of potential perturbations to the biological system (see Section 1).

Model checking is a technique in computer science which is used to automatically check whether a model of a system meets a given specification, and has mostly been applied to hardware or software systems. In order to solve the problem algorithmically, both the model and the specification are formulated with precise mathematical descriptions; if they are given in logic then the task is to verify that a given logical formula is satisfied by a given structure. Because of the temporal ordering inherent in the causal relationships in the behaviour of the systems being modelled, specifications are often given in the form of temporal logic.

Analytical model checking is subject to the state-space explosion, especially when using explicit state space exploration. However, the state space of models of biological systems can be so large or even infinite that the only effective approach is to use simulative model checking – i.e. to check if a temporal logic description of the specification is satisfied by the behaviour generated from simulating the model. Simulative approaches can be off-line, where the model is simulated for some predefined time and then the behaviour is checked, or in-line where the simulation and checking are more tightly coupled. Both approaches are inexact due to the halting problem; however an advantage of the in-line approach is that the simulation can be terminated as soon as the formula being checked is satisfied. Simulative model checking can

be adapted to check the behaviour of the biological system under consideration, assuming that sufficient time series data can be generated by experimental procedures. Given the variability inherent in experimental data due to experimental error as well as the possible stochastic nature of the biological system, such behaviour checking is best performed using a description in a probabilistic temporal logic.

In multiscale model checking we have to deal with several different types of discontinuities. These include discontinuity in the level of behaviour which can be observed in the model and the level which can be observed in the wet lab – e.g. in fly wing, model checking considers protein time course series, while wet lab observation is hair orientation at single time points – and discontinuity in resolution of the time course: time course versus single time points. Clustering of time course data in a multiscale context can require density-based approaches and feature reduction, see e.g., (Maccagnola et al., 2012) who have also developed a pattern mining approach for generating high-level classificatory descriptions of the behaviour of the clusters in temporal logic. The challenges raised by multiscale modelling include the introduction of a new category of properties which have not traditionally been considered in uniscale model checking. These include shapes and patterns in different geometries; colour, density, location and distances in space, all in 1, 2 or 3 dimensions; see Section 6 for related case studies.

Thus we need appropriate languages to express these kinds of properties, and propose the development of temporal logics over geometrical and spatial descriptions, and relationships between them including similarities, equalities and inequalities.

6. Case studies

A recent survey (Baldan et al., 2010) has shown how Petri nets can be applied to transcriptional, signalling, and metabolic networks, or combinations of them, illustrating this with a rich set of case studies. Most of the published case studies focus on the (uniscale) molecular level; however examples at the multi-cellular level include the signal-response behaviour of an organism (Marwan et al., 2005), and developmental processes in multi-cellular pattern formation (Bonzanni et al., 2009, Chen et al., 2009, Liu, 2012).

In (Gilbert and Heiner, 2011) we go one step further and discuss how a computational (Petri net) framework needs to be extended to permit adequate support of multiscale issues. In the following we briefly summarise some lessons learnt from two case studies undertaken within this project with focus on spatial multiscale aspects. In our current modelling approach we discretise space, and in continuous models this corresponds to discretising partial differential equations. See (Herajy and Schwarick, 2012) for an example addressing temporal multiscale aspects with *HPN*.

Phase variation in bacterial colony growth (Gilbert and Heiner, 2011). A common microbial stochastic mechanism is phase variation, in which gene expression is controlled by a reversible genetic mutation, re-arrangement, or modification. Phase variation has traditionally been considered in the context of ‘contingency genes’ in which a sub-population is continuously generated which is pre-adapted to repeated environmental transitions, often to immune selective changes. However recent re-consideration suggests an important potential role in bacterial specialisation and differentiation, and the generation of structured bacterial populations. The diversification of mutation-mediated phase variation is context independent, thus the process can be observed and studied in in-vitro culture conditions and occurs within bacterial colonies. This is the simplest culture condition and can be used to assess basic properties of the stochastic process, such as rate of variation (mutation rate) or contributions of differing

fitness. It also helps to establish the underlying framework for future more complex modelling and determination of local population sub-structures [private communication NS].

The modelling challenge here is to design a generic spatial model for bacterial colonial phase variation, which allows to substantially extend the method developed in (Saunders et al., 2003) to computationally predict rates of phase variation. Previously, phase variation has been characterised by deterministic models, which describe synchronous growth in cell colonies without reflecting how a colony develops in space. Our Petri nets adopt an asynchronous modelling approach so that cells divide individually, and explicitly consider spreading in space.

We have developed \mathcal{SPN}^c models which take into account:

- phase variation between two genotypes;
- the displacement of cells in space, with no cell division, where the surface of the growth medium is modelled by a rectangular grid;
- the displacement of dividing cells, where the parent remains in-situ, and the offspring may displace by one grid position or stay in its current location according to local cell density;
- controlled thickness and speed of spread of the colony.

We assume that the 3D colony is represented by a 2D grid with a finite capacity on each grid position, and there is an equal maximal height over all of the cell colony (i.e., all grid positions have the same capacity). This model could easily be extended to incorporate death of the bacteria, which is different to zero activity (quiescence).

Model parameters include: grid size, mutation rates, fitness, preference of an offspring to stay with its parent, and total number of cells (colony size). All computational experiments are done on the automatically unfolded Petri nets. Unfolding our coloured Petri net for a 100×100 grid yields a plain Petri net with about 30,000 places and 360,000 transitions.

The analysis considers the development over time of the proportion of the given genotype in the total population, and the patterning into characteristic segments. This requires converting the stochastic simulations into 2D representations, see Fig. 8, and performing model checking over 2D shapes. The model permits the prediction of mutation rates and fitness by counting and measuring mutation segments.

Currently, our model predicts behaviour which has not been measured so far in the wet lab — the model generates a time series description of the evolution of the patterns in cell colony (indicated in Fig. 8), while wet lab data just give a snapshot of the final state.

The issues highlighted by this example include:

- multiple scales: individual level (highly abstract representation of phase variation, mutation with cell division/replication of components) to colony level,

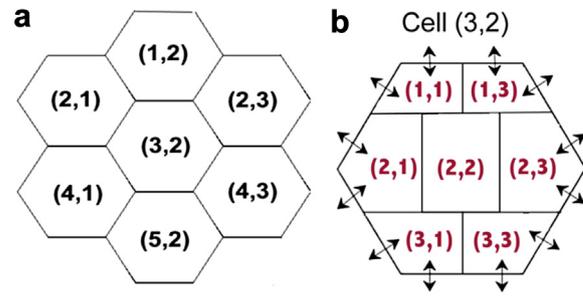


Fig. 9. *Drosophila* wing epithelial cells. (a) Fragment of wing tissue; coordinates represent honeycomb grid position; (b) Cell with seven virtual compartments, arrows denote inter-cellular communication with adjacent neighbouring cells.

- mobility of components,
- states (quiescent or active) of components,
- 2D pattern formation, characterised by size and shape.

Planar cell polarity in *Drosophila* wing (Gao et al., 2012).

Planar cell polarity (PCP) refers to the orientation of cells within the plane of the epithelium, orthogonal to the apical-basal polarity of the cells. This polarisation is required for many developmental events in both vertebrates and non-vertebrates. Defects in PCP in vertebrates are responsible for developmental abnormalities in multiple tissues including the neural tube, the kidney and the inner ear (reviewed in (Simons and Mlodzik, 2008)). The fruit fly *Drosophila melanogaster* has been used extensively as a model to study the signalling mechanisms underlying PCP. The adult *Drosophila* wing comprises about 30,000 hexagonal cells each of which contains a single hair pointing in an invariant distal direction. This hair comprises actin bundles and is extruded from the membrane at the distal edge of the cell during pupal development, at the conclusion of PCP signalling. Preceding this ultimate manifestation of PCP, signalling occurs such that the proteins adopt an asymmetric localisation within each cell. At the initiation of PCP signalling Fmi, Fz, Dsh, Vang and Pk are all present symmetrically at the cell membrane. At the conclusion of PCP signalling Fmi is found at both the proximal and distal cell membrane, Fz and Dsh are found exclusively at the distal cell membrane and Vang and Pk are found exclusively at the proximal cell membrane. The causal mechanism for PCP has yet to be positively identified, but could be due to an unknown and as yet un-identified secreted morphogen signal to which the PCP proteins respond, or a biased transport mechanism within cells.

We have created a multiscale model of PCP (Gao et al., 2011, Gao et al., 2012) based on descriptions of the intracellular signalling pathway within cells, formation of protein complexes between cells across the intercellular gap leading to intracellular communication, and patterns of communication between the hexagonal epithelial

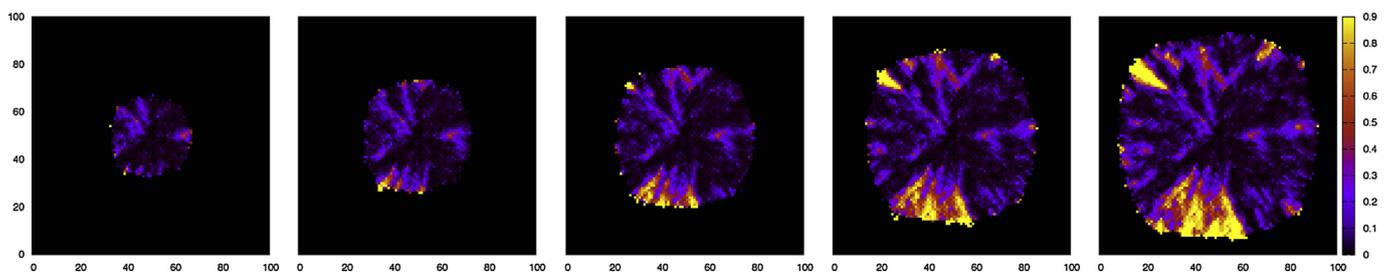


Fig. 8. 2D representation of the development of a binary phase variation cell colony over time and space. Density of the two phenotypes is represented by yellow and dark blue, respectively.

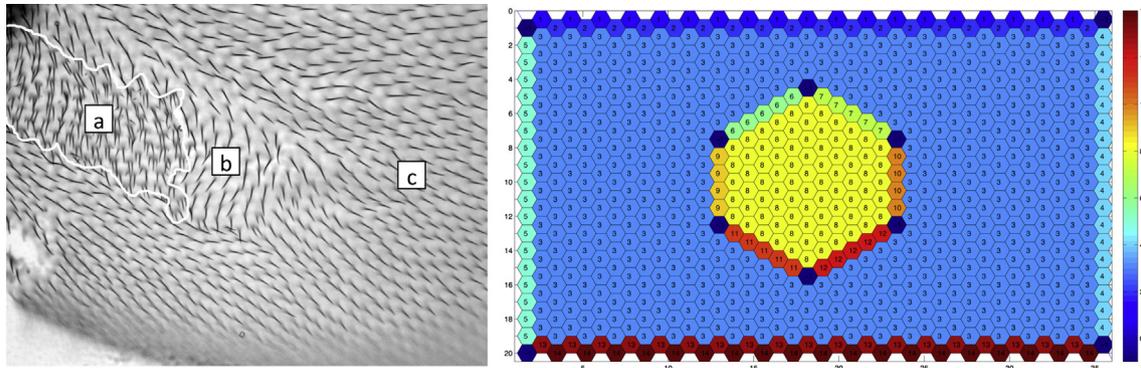


Fig. 10. Left: Image of *Drosophila* wing. (a) Fz- clone, (b) wild-type cells distally adjacent to clone, (c) wild-type cells far from clone. Right: Clustering result for simulation of a circle-like Fz- clone (Gao et al., 2012).

cells in the wing tissue. Each cell has been symmetrically divided into six virtual compartments at the interior of the cell membrane, and one compartment representing the interior of the cell with the nucleus, see Fig. 9 which gives the overall model structure at the tissue and cell level for the epidermal tissue shown in Fig. 2.

The model has been encoded in Hierarchically Coloured Petri nets which we have considered both continuously and stochastically, where the core of the model comprises a description of the PCP signalling network within one virtual compartment, and colours employed to describe the relationship between adjacent compartments in neighbouring cells. Our model uses a two-layer hierarchy where the upper layer describes the layout of the hexagonal cells in the epithelial tissue, and the lower layer describes the layout of the virtual compartments within one cell. The hierarchical mechanism is encoded by two tuples, the first of which represents the coordinates of a cell in the tissue, and the second of which represents the position of a virtual compartment within the cell. Thus the central compartment of the central cell in the fragment of wing tissue in Fig. 9 is addressed by the colour tuples (3,2)(2,2).

Fig. 10 compares our computational results with wet lab data, showing the influence of a patch of mutant Fz cells on neighbouring wild-type cells.

The issues highlighted by this example include

- illustrating how colours can be used to encode both space and hierarchy,
- computational challenges due to the fact that currently simulations must be performed at the unfolded level rather than at the coloured level: the largest model we have constructed so far consists of 800 cells each with 7 virtual compartments, comprising 19 places and 23 transitions as a CPN which when unfolded yields 164,000 places (biochemical entities, each of which is described by an ordinary differential equation) and about a quarter million transitions (reactions); unfolding on a Mac Quad-core 2.26 GHz Intel Xeon takes 2 min and simulation takes 2 h.
- fitting the multiscale model to data, where problems are due to (i) the lack of reliable time-series data for concentrations of key biochemical entities due to the fact that the time period being modelled is during the pupal stage when it is difficult to obtain such data, (ii) data about hair orientation is at the tissue level whereas the kinetic parameters being fitted are at the biochemical level in the virtual compartments, and (iii) these data are at one time point (the adult – i.e. the final state).

7. Summary

In this paper we have presented a discussion of some challenges arising from the need to model and analyse complex biological systems at multiple scales, both spatial and temporal. These challenges have motivated the extension of our former biomodel engineering framework to address some of these issues within the context of multiscale Systems Biology.

We have based our methodology on the use of a structured family of Petri net classes which enables the investigation of a biological system using various modelling abstractions: qualitative, stochastic, continuous and hybrid, optionally in a spatial context. We have illustrated our approach with two major case studies specifically demonstrating multiscale issues in a spatial context, and have highlighted many open issues which need to be addressed in future programmes of research.

Acknowledgement

This research has been partly funded by EPSRC Research Grant EP I036168/1. The phase variation in bacterial colony growth case study has been provided by Nigel Saunders [NS] of Brunel University.

References

- A. Aziz, K. Sanwal, V. Singhal, R. Brayton, Model checking continuous time Markov chains, *ACM Transactions on Computational Logic* 1(1).
- Baldan, P., Cocco, N., Marin, A., Simeoni, M., 2010. Petri nets for modelling metabolic pathways: a survey. *Journal of Natural Computing* 9, 955–989.
- Blossey, R., Cardelli, L., Phillips, A., 2008. Compositionality, stochasticity, and cooperativity in dynamic models of gene regulation. *HFSP Journal* 2 (1), 17–28.
- Bonzanni, N., Krepska, E., Feenstra, K., Fokkink, W., Kielmann, T., Bal, H., Heringa, J., 2009. Executing multicellular differentiation: quantitative predictive modelling of *c. elegans* vulval development. *Bioinformatics* 25, 2049–2056.
- Breitling, R., Donaldson, R., Gilbert, D., Heiner, M., 2010. Biomodel engineering – from structure to behavior. *Lecture Notes Bioinformatics* 5945, 1–12.
- Calzone, L., Chabrier-Rivier, N., Fages, F., Soliman, S., 2006. Machine learning biochemical networks from temporal logic properties. *Transactions on Computational Systems Biology VI*, LNBI 4220, 68–94.
- Chen, L., Masao, N., Kazuko, U., Satoru, M., 2009. Simulation-based model checking approach to cell fate specification during *c. elegans* vulval development by hybrid functional Petri net with extension. *BMC Systems Biology*, 42.
- Clarke, E., Grumberg, O., Peled, D., 2001. *Model Checking*. third printing. MIT Press 1999.
- Donaldson, R., Gilbert, D., 2008. A model checking approach to the parameter estimation of biochemical pathways. LNCS 5307. *Proc. 6th International Conference on Computational Methods in Systems Biology (CMSB 2008)*. Springer, pp. 269–287.
- Endy, D., 2005. Foundations for engineering biology. *Nature* 438 (7067), 449–453.
- Gao, Q., Liu, F., Gilbert, D., Heiner, M., Tree, D., 2011. A multiscale approach to modelling planar cell polarity in *Drosophila* wing using hierarchically coloured

- Petri nets. In: Proc. 9th International Conference on Computational Methods in Systems Biology (CMSB 2011). ACM digital library, pp. 209–218.
- Gao, Q., Gilbert, D., Heiner, M., Liu, F., Maccagnola, D., Tree, D., 2012. Multiscale modelling and analysis of planar cell polarity in the *Drosophila* wing (PrePrints). IEEE/ACM Transactions on Computational Biology and Bioinformatics 99, 1–1.
- Gilbert, D., Heiner, M., 2011. Petri Nets for Multiscale Systems Biology. Brunel University, Uxbridge/London. <http://multiscalepn.brunel.ac.uk/>.
- Gilbert, D., Heiner, M., Lehrack, S., 2007. A unifying framework for modelling and analysing biochemical pathways using Petri nets. LNCS/LNBI 4695. Proc. CMSB. Springer, pp. 200–216.
- Gilbert, D., Heiner, M., Rosser, S., Fulton, R., Gu, X., Trybilo, M., 2008. A case study in model-driven synthetic biology. In: Proc. 2nd IFIP Conference on Biologically Inspired Collaborative Computing (BICC). IFIP WCC 2008, Milano, pp. 163–175.
- Heinemann, M., Panke, S., 2006. Synthetic biology – putting engineering into biology. Bioinformatics 22 (22), 2790–2799.
- Heiner, M., Gilbert, D., 2011. How might Petri nets enhance your systems biology toolkit. LNCS 6709. Proc. PETRI NETS 2011. Springer, pp. 17–37.
- Heiner, M., Koch, I., Will, J., 2004. Model validation of biological pathways using Petri nets – demonstrated for apoptosis. BioSystems 75, 15–28.
- Heiner, M., Gilbert, D., Donaldson, R., 2008. Petri nets for systems and synthetic biology. LNCS 5016. Schools on Formal Methods (SFM). Springer, pp. 215–264.
- Heiner, M., Herajy, M., Liu, F., Rohr, C., Schwarick, M., 2012. Snoopy – a unifying Petri net tool. LNCS. Proc. PETRI NETS 2012, vol. 7347. Springer, pp. 398–407.
- Herajy, M., Heiner, M., 2012. Hybrid representation and simulation of stiff biochemical networks. Journal of Nonlinear Analysis: Hybrid Systems 6 (4), 942–959.
- Herajy, M., Schwarick, M., 2012. A hybrid Petri net model of the eukaryotic cell cycle. In: Proc. of the 3rd International Workshop on Biological Processes & Petri Nets (BioPPN), Satellite Event of Petri Nets 2012, Vol. 852 of CEUR Workshop Proceedings, CEUR-WS.org, pp. 29–43.
- Hood, L., 2003. Systems biology: integrating technology, biology, and computation. Mechanisms of Ageing and Development 124 (1), 9–16.
- Katoen C. Baier, B. Haverkort, H. Hermans, J.-P. Katoen, Model checking algorithms for continuous-time Markov chains, IEEE Transactions on Software Engineering 29(6).
- Liu, F. Colored Petri nets for systems biology, Ph.D. thesis, Brandenburg University of Technology Cottbus, Department of Computer Science (January 2012).
- Maccagnola, D., Messina, E., Gao, Q., Gilbert, D., 2012. A machine learning approach for generating temporal logic descriptions of complex model behaviours. In: Proc. Winter Simulation Conference 2012.
- Marwan, W., Sujathab, A., Starostzik, C., 2005. Reconstructing the regulatory network controlling commitment and sporulation in *Physarum polycephalum* based on hierarchical Petri net modeling and simulation. Journal of Theoretical Biology 236 (4), 349–365.
- Marwan, W., Wagler, A., Weismantel, R., 2011. Petri nets as a framework for the reconstruction and analysis of signal transduction pathways and regulatory networks. Journal of Natural Computing 10 (2), 639–654.
- Melham, T., 2013. Modelling, abstraction, and computation in systems biology: a view from computer science. Progress in Biophysics and Molecular Biology 111 (2-3), 129–136.
- Pnueli, A., 1977. The temporal logic of programs. In: Proceedings of the 18th IEEE Symposium on the Foundations of Computer Science (FOCS-77). IEEE Computer Society Press, Providence, Rhode Island, pp. 46–57.
- Qu, Z., Garfinkel, A., Weiss, J., Nivala, M., 2011. Multi-scale modeling in biology: how to bridge the gaps between scales? Progress in Biophysics and Molecular Biology 107 (1), 21–31.
- Rohr, C., Marwan, W., Heiner, M., 2010. Snoopy – a unifying Petri net framework to investigate biomolecular networks. Bioinformatics 26 (7), 974–975.
- Saunders, N., Moxon, E., Gravenor, M., 2003. Mutation rates: estimating phase variation rates when fitness differences are present and their impact on population structure. Microbiology 149 (2), 485–495.
- Schwarick, M., Rohr, C., Heiner, M., 2011. Marcie – model checking and reachability analysis done efficiently. In: Proc. 8th International Conference on Quantitative Evaluation of SysTems (QEST 2011). IEEE CS Press, pp. 91–100.
- Simons, M., Mlodzik, M., 2008. Planar cell polarity signaling: from fly development to human disease. Annual Review of Genetics 42, 517–540.
- Soliman, S., Heiner, M., 2010. A unique transformation from ordinary differential equations to reaction networks. PLoS ONE 5 (12), e14284.
- van Riel, N.A.W., 2006. Dynamic modelling and analysis of biochemical networks: mechanism-based models and model-based experiments. Briefings in Bioinformatics 7 (4), 364–374.
- Wegener, J., Schwarick, M., Heiner, M., 2011. A Plugin system for Charlie. In: Proc. CS&P 2011. Bialystok University of Technology, pp. 531–554.
- Wu, Z., Yang, S., Gilbert, D., 2012. A hybrid approach to piece-wise modelling of biochemical systems. In: LNCS 7491, Proc. Parallel Problem Solving from Nature (PPSN XII), pp. 519–528.