What Can Formal Methods Bring to Systems Biology?

Nicola Bonzanni, K. Anton Feenstra, Wan Fokkink, and Elzbieta Krepska

Vrije Universiteit Amsterdam, Department of Computer Science
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{bonzanni,feenstra,wanf,ekr}@few.vu.nl

Abstract. This position paper argues that the operational modelling approaches from the formal methods community can be applied fruitfully within the systems biology domain. The results can be complementary to the traditional mathematical descriptive modelling approaches used in systems biology. We discuss one example: a recent Petri net analysis of *C. elegans* vulval development.

1 Systems Biology

Systems biology studies complex interactions in biological systems, with the aim to understand better the entirety of processes that happen in such a system, as well as to grasp the emergent properties of such a system as a whole. This can for instance be at the level of metabolic or interaction networks, signal transduction, genetic regulatory networks, multi-cellular development, or social behaviour of insects.

The last decade has seen a rapid and successful development in the collaboration between biologists and computer scientists in the area of systems biology and bioinformatics. It has turned out that formal modelling and analysis techniques that have been developed for distributed computer systems, are applicable to biological systems as well. Namely, both kinds of systems have a lot in common. Biological systems are built from separate components that communicate with each other and thus influence each other's behaviour. Notably, signal transduction within a cell consists of cascades of biochemical reactions, by which for instance genes are activated or down-regulated. The genes themselves produce the proteins that drive signal transduction, and cells can be connected in a multicellular organism, making this basically one large, complex distributed system. Another, very different, example at the organism level is how ants in one colony send stimuli to each other in the form of pheromones.

Biological systems are reactive systems, as they continuously interact with their environment. In November 2002, David Harel [11] put forward a grand challenge to computer science, to build a fully animated model of a multi-cellular organism as a reactive system; specifically, he suggested to build such a model of the *C. elegans* nematode worm, which serves as a one of the model organisms in developmental biology.

Open questions in biology that could be addressed in such a modelling framework include the following, listed in order from a detailed, molecular viewpoint to a more global view of whole organisms:

- How complete is our knowledge of metabolic, signalling and regulatory processes at a molecular level?

- How is the interplay between different pathways or network modules organized and regulated?
- How is the interaction between intra-cellular processes and inter/extra-cellular processes organized?
- How do cells self-organize?
- How do cells differentiate?
- How are self-organization and differentiation of cells connected?
- How does self-organization and differentiation lead to the formation of complex structures like organs (e.g. the eye, brain, kidney)?

One grand open question that pervades the whole of biological research is, how could all of this *evolve*? This is exemplified by the title of the 1973 essay by Theodosius Dobzhansky [4] that "Nothing in biology makes sense except in the light of evolution". Some recent theoretical work [5] highlights an interesting possibility, that flexibility in regulation is a necessary component of evolution, but has itself been evolved in biological systems.

2 Formal Models of Biological Systems

Why would a biologist want to use formal models? First of all, formal models can be an excellent way to store and share knowledge on biological systems, and to reason about such systems. Furthermore, in vivo experiments in the lab tend to take an awfully long time, and are labour intensive. In comparison, in silico experiments (i.e. computer experiments) can take relatively little time and effort. And for instance genetic perturbations can be difficult (or unethical) to perform in the lab, while they may require trivial adaptations of a formal model.

The time is ripe for exploiting the synergy between (systems) biology and formal methods. First of all we have reached the point where biological knowledge of for instance signal transduction has become so detailed, that enough information is available to start building sensible formal models. Second, the development of analysis techniques for formal methods, and the power of the underlying computer hardware, has made it possible to apply formal methods to very complex systems. Although we are certainly not (and possibly never will be) at a level where a full-fledged formal analysis of the entire genetic regulatory network of one cell is within reach, we can definitely already study interesting, and challenging, fragments of such networks.

It is important to realise that biology (like e.g. physics, chemistry, sociology, economics) is an empirical science. This is basically orthogonal to the standard application of formal methods in computer science, where a formal analysis is used to design and prove properties of a computer system. If a desired property of a computer system turns out to fail, then we can in principle adapt the system at hand. In contrast, biological systems are simply (and quite literally) a fact of life, and formal models 'only' serve to better understand the inner workings and emergent properties of such systems. So while in computer science model validation typically leads to a redesign of the corresponding computer system, in systems biology it leads to a redesign of the model itself, if in silico experiments on the model do not correspond with in vivo experiments on the real-life

biological system. A nice comparison between these two approaches can be found in the introduction of [18].

Fisher and Henzinger [6] distinguish two kinds of models for biological systems: operational versus denotational (or, as they phrase it, computational versus mathematical). On the one hand, operational models (such as Petri nets) are executable and mimic biological processes. On the other hand, denotational models (such as differential equations) express mathematical relationships between quantities and how they change over time. Denotational models are in general quantitative, and in systems biology tend to require a lot of computation power to simulate, let alone to solve mathematically. Also it is often practically impossible to obtain the precise quantitative information needed for such models. Operational models are in general qualitative, and are thus at a higher abstraction level and easier to analyse. Moreover, Fisher and Henzinger, as well as Regev and Shapiro [17], make a convincing case that a good operational model may explain the mechanisms behind a biological system in a more intuitive fashion than a denotational model.

Metaphorically one can ask the question whether molecules in a cell, or cells themselves, solve differential equations to decide what to do in a particular situation, or rather when they encounter one another follow simple sets of rules derived from their physical interactions. In that respect, one may consider the continuous, mathematical models as an approximation of the discrete molecular processes, rather than viewing the qualitative model as a course-grained abstraction of a continuous reality.

An operational model progresses from state to state, where an event at a local component gives rise to a state transition at the global system level. Fisher et al. [7] argue that (unbounded) asynchrony does not mimic real-life biological behaviour properly. Typically, asynchrony allows that one component keeps on executing events, while another component is frozen out, or executes only few events. While in real life, all components are able to execute at a certain rate. Bounded asynchrony, a phrase coined by Fisher et al. [7], lets components proceed in an asynchronous fashion, while making sure that they all can proceed at their own rate. A good example of bounded asynchrony is the maximally parallel execution semantics of Petri nets; we will return to this semantics in Section 3.

We briefly mention the three modelling paradigms from the formal methods community that are used most frequently for building operational models of biological systems.

Petri nets are well-suited for modelling biochemical networks such as genetic regulatory pathways. The places in a Petri net can represent genes, protein species and complexes. Transitions represent reactions or transfer of a signal. Arcs represent reaction substrates and products. Firing of a transition is execution of a reaction: consuming substrates and creating products. Cell Illustrator [15] is an example of a Petri net tool that targets biological mechanisms and pathways.

Process calculi, such as process algebra and the π-calculus, extended with probabilities or stochastics, can be used to model the interaction between organisms. Early ground-breaking work in this direction was done by Tofts [21] in the context of process algebra, with regard to ant behaviour. The Bioambients calculus [16], which is based on the π-calculus, targets various aspects of molecular localisation and compartmentalization.

Live sequence charts are an extension of the graphical specification language message sequence charts; notably, they allow a distinction between mandatory and possible behaviour. They have been used successfully by Harel and his co-workers to build visual models of reactive biological systems, see e.g. [12].

Model checking is in principle an excellent methodology to verify interesting properties of specifications in any of these three formalisms. And as is well-known, abstraction techniques and distributed model checking (see e.g. [1]) can help to alleviate the state explosion problem. However, in view of the very large scale and complexity of biological systems, so far even these optimisation techniques cannot push model checking applications in this area beyond toy examples. Simulations methods are commonly used to evaluate complex and high-dimensional models, and are applicable in principle to both operational and denotational models. Well-known drawbacks, compared to model checking, are that this approach can suffer from limited sampling due to the high-dimensional state space, and that there may be corners of the state space that have a biological relevance but that are very hard to reach with simulations. Still, in spite of these drawbacks, for the moment Monte Carlo simulations are currently the best method to analyse formal specifications of real-life biological systems.

In our view, for the successful application of formal methods in the systems biology domain, it is expedient to use a simple modelling framework, and analysis techniques that take relatively little computation power. This may at first sound paradoxical, but simplicity in modelling and analysis methods will make it easier to master the enormous complexity of real-life biological systems. Moreover, it will help to communicate with biologists on the basis of formal models, and in the hopefully not too far future will make it attractive for biologists to start using formal modelling tools.

3   A Petri Net Analysis of *C. elegans* Vulval Development

Petri nets representing regulatory and signalling networks We recall that a Petri net is a bipartite directed graph consisting of two kinds of nodes: places that indicate the local availability of resources, and transitions which are active components that can change the state of the resources. Each place can hold one or more tokens. Weighted arcs connect places and transitions. In [13] we explained a method to represent biological knowledge as a Petri net. As explained before, places represent genes and protein species, i.e., bound and unbound, active and inactive, or at different locations, while transitions represent biological processes. Firing of a transition is execution of a process, e.g. consuming substrates or creating products. The number of tokens in a place is interpreted as follows. For genes as a boolean value, 0 means not present and 1 present. For proteins, there are abstract concentration levels 0-6, going from not present, via low, medium, and high concentration to saturated level. The rationale behind this approach is to abstract away from unknown absolute molecule concentration levels, as we intend to represent relative concentrations and rates. If desired, a modeller could fine-tune the granularity of the model by adjusting the number of available concentration levels.

Biological systems are highly concurrent, as in cells all reactions can happen in parallel and most are independent of each other. Therefore, in [13] we advocate to use what is called maximal parallelism [3]. A fully asynchronous approach would allow one part

of the network to deploy prolonged activity, while another part of the network shows no activity at all. In real life, all parts can progress at roughly the same rate. Maximal parallelism promotes activity throughout the network. The maximal parallel execution semantics can be summarised informally as execute greedily as many transitions as possible in one step. A maximally parallel step leaves no enabled transitions in the net, and, in principle, should be developed in such a way that it corresponds to one time step in the evolution of the biological system. This is possible because the modeller can capture relative rates and concentration levels using appropriate weights on arcs. Typically, if in one time unit a protein A is produced four times more than a protein B, then the transition that captures production of A should have a weight that is four times as large as the weight of the one that captures B production.

In nature a cell tends to saturate with a product, and as a result the reaction slows down or stops. To mimic this behaviour, each place in the Petri net has a predefined maximum capacity of six. To guarantee that the highest concentration level can be attained, we introduced bounded execution with overshooting. A transition can only fire if each output place holds fewer than six tokens. Since each transition can possibly move more than one token at once into its output places, each transition can overshoot the pre-given capacity at most once.

C. elegans vulval development C. elegans is a round worm, about 1mm in length, living in soil. In order to lay eggs, the *C. elegans* hermaphrodites grow an organ called vulva. The complexity and universality of the biological mechanisms underlying the vulval development (e.g. cell-cell interactions, cell differentiation, cross-talk between pathways, gene regulation), and the intensive biological investigations undertaken during the last 20 years [19] make this process an extremely appealing case study [8,9,10,14,20]. In particular, the considerable amount of descriptive biological knowledge about the process joint with the lack of precise biochemical parameters, and the large number of genetic perturbations tested in vivo, welcome the research of alternative modelling procedures. These approaches should be able to express the descriptive knowledge in a formal way, abstract the processes enough to overcome the absence of fine-grained biochemical parameters, and check the behaviour of the system with a sound methodology.

Recently we developed a Petri net model of the process that leads to the formation of the vulva during *C. elegans* development [2], using the Petri net framework described above. It comprises 600 nodes (places and transitions) and 1000 arcs. In this network we could identify different modules. These correspond to different biological functions, such as gene expression, protein activation, and protein degradation. It is possible to reuse modules corresponding to a function, like small building blocks, to compose more complex modules, and eventually build a full cell. The cell itself is a module that can be reused, as can other modules like pathways or cascades.

To analyse the Petri net model, we applied Monte Carlo simulations. We simulated 64 different genetic perturbations. Twenty-two experiments previously selected in [9] were used for model calibration. Thirty perturbations were used for validation: 26 from [9], three from [19], and one from [22]. The remaining twelve simulations constitute new predictions that invite further in vivo experiments.

This case study shows that the basic Petri net formalism can be used effectively to mimic and comprehend complex biological processes.

4 Conclusions

Transforming 'data' into 'knowledge' is a holy grail in Life Sciences. Sometimes we have much data but relatively little descriptive knowledge, e.g. a whole genome sequenced and protein interaction data, but little information about the single genes and their functions. At other times we have excellent descriptive knowledge about a biological process but lack the biochemical details to simulate or explain accurately the phenomenon. For instance, we may know the response of an organism to a certain stimulus but we do not know which molecules are responsible, or we may know the molecules but not all the biochemical parameters to reproduce the behaviour of the organism in silico.

Reaching the sweet spot in between abstraction and biological significance is one of the big challenges in applying formal methods to biology. On the one hand, a fine-grained approach potentially gives more detailed predictions and a better approximation of the observed behaviour, but it has to cope with a huge number of parameters that are largely unknown and could not be effectively handled by, for instance, model checking techniques. On the other hand, a coarse-grained approach developed at a higher level of abstraction needs fewer detailed parameters and is computationally cheaper, but it might have to be tailored to answering a single question, lowering the overall biological significance of the model. Therefore, it is crucial to choose the appropriate abstraction level and formalism in respect to the biological questions that the modeller wants to address.

To pick up the right questions is a pivotal choice, and to understand their biological significance is essential. In order to accomplish these two goals, it is necessary to establish a clear and unambiguous communication channel between 'biologists' and 'computer scientists'. Furthermore, it is necessary to expand the application of formal methods beyond the manageable but only moderately interesting collection of the toy examples. Although several different formal methods can achieve such objectives, in our experience the intuitiveness of its graphical representation, tied with the strict formal definition of Petri net, contributed greatly to establish a common ground for 'biologists' and 'computer scientists'.

References

1. Barnat, J., Brim, L., Cerná, I., Drazan, S., Safránek, D.: Parallel model checking large-scale genetic regulatory networks with DiVinE. In: Proc. FBTC 2007. ENTCS, vol. 194(3), pp. 35–50. Elsevier, Amsterdam (2008)
2. Bonzanni, N., Krepska, E., Feenstra, A., Fokkink, W., Kielmann, T., Bal, H., Heringa, J.: Executing multicellular differentiation: Quantitative predictive modelling of *C. elegans* vulval development. Bioinformatics (in press)
3. Burkhard, H.-D.: On priorities of parallelism. In: Salwicki, A. (ed.) Logic of Programs 1980. LNCS, vol. 148, pp. 86–97. Springer, Heidelberg (1983)
4. Dobzhansky, T.: Nothing in Biology Makes Sense Except in the Light of Evolution. American Biology Teacher 35, 125–129 (1973)
5. Crombach, A., Hogeweg, P.: Evolution of evolvability in gene regulatory networks. PLoS Comput. Biol. 4, e1000112 (2008), doi:10.1371/journal.pcbi.1000112

6. Fisher, J., Henzinger, T.: Executable cell biology. Nature Biotechnology 25(11), 1239–1249 (2007)
7. Fisher, J., Henzinger, T., Mateescu, M., Piterman, N.: Bounded Asynchrony: Concurrency for Modeling Cell-Cell Interactions. In: Fisher, J. (ed.) FMSB 2008. LNCS (LNBI), vol. 5054, pp. 17–32. Springer, Heidelberg (2008)
8. Fisher, J., Piterman, N., Hajnal, A., Henzinger, T.A.: Predictive modeling of signaling crosstalk during *C. elegans* vulval development. PLoS Comput. Biol. 3, e92 (2007)
9. Fisher, J., Piterman, N., Jane Albert Hubbard, E., Stern, M.J., Harel, D.: Computational insights into Caenorhabditis elegans vulval development. P. Natl. Acad. Sci. USA 102, 1951–1956 (2005)
10. Giurumescu, C.A., Sternberg, P.W., Asthagiri, A.R.: Intercellular coupling amplifies fate segregation during *Caenorhabditis elegans* vulval development. P. Natl. Acad. Sci. USA 103, 1331–1336 (2006)
11. Harel, D.: A grand challenge for computing: Towards full reactive modeling of a multicellular animal. Bulletin of the EATCS 81, 226–235 (2003)
12. Kam, N., Harel, D., Kugler, H., Marelly, R., Pnueli, A., Jane Albert Hubbard, E., Stern, M.: Formal modeling of *C. elegans* development: A scenario-based approach. In: Priami, C. (ed.) CMSB 2003. LNCS, vol. 2602, pp. 4–20. Springer, Heidelberg (2003)
13. Krepska, E., Bonzanni, N., Feenstra, A., Fokkink, W., Kielmann, T., Bal, H., Heringa, J.: Design issues for qualitative modelling of biological cells with Petri nets. In: Fisher, J. (ed.) FMSB 2008. LNCS (LNBI), vol. 5054, pp. 48–62. Springer, Heidelberg (2008)
14. Li, C., Nagasaki, M., Ueno, K., Miyano, S.: Simulation-based model checking approach to cell fate specification during *Caenorhabditis elegans* vulval development by hybrid functional Petri net with extension. BMC Syst. Biol. 3, 42 (2009)
15. Nagasaki, M., Saito, A., Doi, A., Matsuno, H., Miyano, S.: Using Cell Illustrator and Pathway Databases. Springer, Heidelberg (2009)
16. Regev, A., Panina, E., Silverman, W., Cardelli, L., Shapiro, E.: BioAmbients: An abstraction for biological compartments. Theoretical Computer Science 325(1), 141–167 (2004)
17. Regev, A., Shapiro, E.: Cellular abstractions: Cells as computation. Nature 419, 343 (2002)
18. Sadot, A., Fisher, J., Barak, D., Admanit, Y., Stern, M., Hubbard, J.A., Harel, D.: Toward verified biological models. IEEE/ACM Transactions on Computational Biology and Bioinformatics 5(2), 223–234 (2008)
19. Sternberg, P.W.: Chapter on vulval development in *Wormbook* (June 2005), http://www.wormbook.org/chapters/www_vulvaldev/vulvaldev.html
20. Sun, X., Hong, P.: Computational modeling of *Caenorhabditis elegans* vulval induction. Bioinformatics 507, i499–i507 (2007)
21. Tofts, C.: Describing social insect behaviour using process algebra. Transactions of the Society for Computer Simulation 9(4), 227–283 (1992)
22. Yoo, A.S., Greenwald, I.: LIN-12/Notch activation leads to microRNA-mediated downregulation of Vav in *C. elegans*. Science 310, 1330–1333 (2005)