Proceedings of the 2006 Winter Simulation Conference
L. F. Perrone, F. P. Wieland, J. Liu, B. G. Lawson, D. M. Nicol, and R. M. Fujimoto, eds.

COMPREHENSIVE AND REALISTIC MODELING OF BIOLOGICAL SYSTEMS

David Harel

Department of Computer Science and Applied Mathematics
The Weizmann Institute of Science
Rehovot 76100, ISRAEL

ABSTRACT

In comprehensive modeling the main purpose is to understand an entire biological system in detail, utilizing in the modeling effort all that is known about the system, and to use that understanding to analyze and predict behavior in silico. In realistic modeling the main issue is to model the behavior of actual elements, making possible totally interactive and modifiable realistic executions/simulations that reveal emergent properties. I will address the motivation for such modeling and the philosophy underlying the techniques for carrying it out, as well as the crucial question of when such models are to be deemed valid, or complete. The examples I will present will be from among the biological modeling efforts my group has been involved in: T cell development in the thymus, lymph node behavior, embryonic development of the pancreas, the C. elegans reproduction system and a generic cell model.

1 OVERVIEW

This is an abstract of an invited lecture to be given at the Winter Simulation Conference in Monterey, in December 2006, as part of the Track on Modeling and Simulation in Computational Biology. The lecture is about comprehensive and realistic modeling of natural systems, with particular emphasis on modeling biology. It emphasizes the two adjectives "comprehensive" and "realistic", as applied to modeling system from Nature, and the questions it tries to deal with include the following:

- What kinds of systems should we model?
- Why do we want to model?
- How should we model?
- When are we done?

In *comprehensive* modeling the goal is to model an entire organ, an entire organism, or even an entire population, in a variety of linked and zoomable levels of abstraction. This is to be contrasted with more conventional types of modeling where one is interested in a specific aspect of a system and the modeling is aimed at getting particular results or making particular predictions. The motivation for comprehensive modeling is mainly to gain a true and extremely deep understanding of the entire system, including its development and behavior over time. However, we also want to be able to "go wild" with it, testing (and thus predicting) its behavior under varying circumstances, etc.

It is obvious that comprehensive modeling, if carried out successfully, can yield far-ranging benefits for biology and for science in general. However, its immediate benefits may be somewhat limited, since it is not designed to be a short term effort aimed at solving a particular problem, but rather to greatly broaden our understanding of biology, deepening knowledge and insight.

The notion of *realistic* modeling is a key issue, and it means several things. First, a model must capture not only some kind of overall average-case stochastic behavior of the system, but also, and more importantly, the behavior of the individual entities, their inter-relationships via cooperation, competition, cause-effect, etc., including subtle issues of concurrency and time-criticality. In fact, it is best if the model is such that the overall emergent picture is the *result* of the combined behavior of the individually modeled entities.

Second, a realistic model must also be *fully executable*, which is more than the ability to carry out a probabilistic computation and generate probable outcomes. Rather, we want the ability to execute the "program" of the system any way we want; which, just like running any computer program, should be doable on various inputs, in deterministic, non-deterministic or stochastic fashion, in a one-step-at-a-time debugging fashion, in ways that highlight the behavior of individual pieces, in best, worst and average case fashion, and much more. Model execution should be the true analogue of running a conventional computer program. And in the same vein, the kinds of analysis we want to be able to do are to be the model analysis analogue of program verification, validation and complexity analysis.

1-4244-0501-7/06/$20.00 ©2006 IEEE

A third aspect of realism in modeling has to do with ease of comprehension — both of the model itself and of its dynamics during execution. We want the experts of the subject matter (biologists, in this case) to be able to model on their own, or at the very least to comprehend and modify existing models. Thus, heavy use of differential equations, logic or algebraic calculi in the modeling has the disadvantage of being unfitting for these experts, and indeed it might alienate them.

In way of illustrating the "realistic" facet of modeling, the lecture goes on to describes the general approach to modeling taken by our group. We view the biological artifacts to be modeled as *reactive systems* (Harel and Pnueli 1985), and use for their modeling and simulation *visual formalisms* (Harel 1988). These are graphical, diagrammatic languages that are both intuitive and mathematically rigorous, and are supported by powerful tools that enable full model executability. They are linkable to object diagrams and GUIs, and other structural descriptions of the system under development and its front-end, as well as to full animation by an idea we call *reactive animation* (Efroni, Harel, and Cohen 2005). At present, such languages and tools — often based on the object-oriented paradigm — are being strengthened by verification modules, making it possible not only to execute and simulate the system models (test and observe) but also to verify dynamic properties thereof (prove) (Sadot et al. 2006). They are also linkable to tools for dealing with the system's continuous aspects (e.g., Matlab) in a full hybrid fashion.

One of two visual formalism approaches that we use is *state-based*, encouraging an *intra-object* style of specification. It uses the language of *statecharts* (Harel 1987, Harel and Gery 1997) to describe the system's behavior by objects. One powerful tool supporting this approach is Rhapsody from I-Logix, but there are many statechart tools. (Matlab has also adopted statecharts for its discrete aspects, in its StateFlow tool.) Another, more recent approach is *scenario-based*, and *inter-object* in spirit. It uses the language of *live sequence charts* (LSCs) (Damm and Harel 2001) and allows one to play in the behavior directly from the system's GUI and to then play it out just as if it were an intra-object model (Harel and Marelly 2003). In both cases, the model's objects are considered to exist as individual entities, and when executed they interact with others in ways that are appealingly realistic.

The lecture then goes on to discuss a *grand challenge* that was proposed a few years ago to the computer science and systems biology community (Harel 2003), which is to fully model an entire multi-cellular organism. We actually have a particular one in mind, the *Caenorhabditis elegans* nematode worm, better known simply as *C. elegans*, a suggestion that is in line with the extraordinarily insightful 40-year old proposal of Sydney Brenner, who chose this creature to challenge biologists with the task of discovering the entire development and neurobiology of a living creature. (For this proposal and the tremendously influential work that he and others did following it, Brenner shared the 2002 Nobel Prize in Physiology or Medicine.)

This challenge — which we estimate to require many years of work by many research groups with diverse backgrounds, and which might never really be achieved — is to construct a full, true-to-all-known facts 4-dimensional model of this worm (or of a comparable multi-cellular animal), which is easily extendable as new facts are discovered. The front end would be an anatomically correct, animated graphical rendition, tightly linked to a reactive system model of the entire creature. The model would be fully executable, flexible, interactive, comprehensive and comprehensible. It would enable realistic simulation of the worm's development and behavior over time (the fourth dimension), which would help uncover gaps, correct errors, suggest new experiments and help predict unobserved phenomena. It would be zoomable, enabling easy switching between levels of detail (reaching down at least to the cellular level, and possibly the molecular level at some points), and allowing researchers to see and understand the organism and its behavior in ways not otherwise possible.

The underlying computational framework would be not only rigorous and realistic, but would be set up in such a way that biologists would be able to enter new data themselves as it is discovered, and even plug in varying theses about aspects of behavior that are not yet known, in order to see their effects.

In order to lend support to this outlandish idea, the lecture then goes on to illustrate some facets of the general approach by describing briefly a number of modeling efforts that have been made, or are being made, in our research group. They include:

1. T-cell development in the thymus (Efroni, Harel, and Cohen 2003, 2005).
2. Vulval cell fate determination in C. elegans (Kam et al. 2002, Fisher et al. 2005).
3. Embryonic development of the pancreas (unpublished yet).
4. Cell behavior and development of the lymph node (Swerdlin, Cohen, and Harel 2006).
5. Generic cell behavior and specialization (current work).

Finally, the particularly interesting question of how we know when we are done will be addressed. In other words, when is a comprehensive, realistic model deemed complete, or valid? Since the modeling is not done in order to answer some particular questions, but to understand in general, it is not clear when such a model can be labeled "good".

Here a sort of Turing test is proposed, but with a Popperian twist: a model of an entire biological system is

complete and valid if a team of professionals cannot tell the difference between the model and the real thing; see (Harel 2005). There are many difficulties that have to be overcome for such a test to be even conceivable, such as devising the "buffer" that must be set up to prevent the interrogating team from knowing the difference simply by peripheral things like sight and smell, or by the time difference between a computerized model answering a query and a lab experiment set up to do the same. And the levels of detail must be clearly agreed upon in advance, so that when modeling a worm or a fly the interrogators don't ask questions about quarks or galaxies. The Popperian twist comes from the fact that once such a model passes the test, it will inevitably change over time as science develops and we learn more about the system we are modeling — all this in the good spirit of Popper's philosophy of science.

This test might be too outlandish to be taken totally seriously, but it does appear to capture the notion of prediction-confirmation taken to the limit. And it does try, just like Turing's original test for computerized intelligence (Turing 1950), to put an upper bound on what is needed for us to say that we have really and truly managed to model a natural system.

REFERENCES

Damm, W., and D. Harel. 2001. LSCs: breathing life into message sequence charts. *Formal Methods in System Design* 19 (1): 45–80. (Early version in *Proceedings of the 3rd IFIP International Conference on Formal Methods for Open Object-Based Distributed Systems (FMOODS'99)*, 293–312. Kluwer.

Efroni, S., D. Harel, and I. Cohen. 2003. Towards rigorous comprehension of biological complexity: modeling, execution and visualization of thymic T cell maturation. *Genome Research* 13:2485–2497.

Efroni, S., D. Harel, and I. Cohen. 2005. Reactive animation: realistic modeling of complex dynamic systems. *IEEE Computer* 38 (1): 38–47.

Fisher, J., N. Piterman, E. Hubbard, M. Stern, and D. Harel. 2005. Computational insights into C. elegans vulval development. *Proceedings of the National Academy of Sciences* 6 (102): 1951–1956.

Harel, D. 1987. Statecharts: a visual formalism for complex systems. *Science of Computer Programming* 8 (3): 231–274.

Harel, D. 1988. On visual formalisms. *Communications of the Association for Computing Machinery* 31 (5): 514–530.

Harel, D. 2003. A grand challenge for computing: towards full reactive modeling of a multi-cellular animal. *Bulletin of the European Association for Theoretical Computer Science (EATCS)* (81): 226–235. (Reprinted in *Current Trends in Theoretical Computer Science: The Challenge of the New Century*, Algorithms and Complexity, Vol I, ed. Paun, Rozenberg and Salomaa, 559–568, World Scientific, 2004).

Harel, D. 2005. A turing-like test for biological modeling. *Nature Biotechnology* 23:495–496.

Harel, D., and E. Gery. 1997. Executable object modeling with statecharts. *IEEE Computer* 23 (7): 31–42.

Harel, D., and R. Marelly. 2003. *Come, let's play: scenario-based programming using lscs and the play-engine*. Springer.

Harel, D., and A. Pnueli. 1985. On the development of reactive systems. In *Logics and Models of Concurrent Systems*, ed. K. Apt, Volume F-13 of *NATO ASI Series*, 477–498. Springer-Verlag.

Kam, N., D. Harel, H. Kugler, R. Marelly, A. Pnueli, E. Hubbard, and M. Stern. 2002. Formal modeling of C. elegans development: a scenario-based approach. In *Proceedings of the 1st International Workshop on Computational Methods in Systems Biology (ICMSB 2003)*, Number 2602 in LNCS, 4–20. Springer. (Revised version in *Modeling in Molecular Biology*, ed. G. Ciobanu and G. Rozenberg, 151-173. Berlin:Springer, 2004.).

Sadot, A., J. Fisher, D. Barak, Y. Admanit, M. Stern, E. Hubbard, and D. Harel. 2006. Towards verified biological models. Submitted.

Swerdlin, N., I. Cohen, and D. Harel. 2006. Towards an *in-silico* lymph node: a realistic approach to modeling dynamic behavior of lymphocytes. Submitted.

Turing, A. 1950. Computing machinery and intelligence. *Mind* 59:433–460.

AUTHOR BIOGRAPHY

DAVID HAREL has been at the Weizmann Institute of Science since 1980. He was a department head for six years and Dean of the Faculty of Mathematics and Computer Science for seven. He is also co-founder of I-Logix, Inc. He received his PhD from MIT in 1978. In the past he worked mainly in theoretical computer science, and now he works in several other areas, including software and systems engineering, biological modeling, and the synthesis and communication of smell. He is the inventor of statecharts and co-inventor of live sequence charts, and co-designed Statemate, Rhapsody and the Play-Engine. Among his awards are the ACM Karlstrom Outstanding Educator Award (1992), the Israel Prize (2004), the ACM SIGSOFT Outstanding Research Award (2006), and two honorary doctorates. He is a Fellow of the ACM and of the IEEE. . His e-mail address is <dharel@weizmann.ac.il>, and his web page is <http://www.wisdom.weizmann.ac.il/~dharel>.