ORIGINAL ARTICLE

On comprehensive and realistic modeling: Some ruminations on the what, the how and the why

David Harel, PhD

The Weizmann Institute of Science,
Rehovot, Israel
dharel@weizmann.ac.il

Clin Invest Med 2005; 28 (6): 334–337.

Summary
This short paper is about comprehensive realistic modeling in general. I am no expert at all on health care or on modeling health-related systems. Rather, I am a computer scientist and, in recent years, have spent time applying some of my work on systems and software engineering to the modeling of biology. Indeed, the examples given in the talk are of two of our group's biological modeling projects. Nevertheless, I invited members of the audience to try to substitute "biology" for "health care" throughout the lecture. All I promised was that this experiment could yield interesting, perhaps thought-provoking, results. Towards the end I posed a "grand challenge" for the health-care modeling community.

The lecture emphasizes the two adjectives "comprehensive" and "realistic", as applied to modeling, and the questions it tries to deal with include:

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

One of the main points made is to highlight the notion of *comprehensive* modeling – where the goal is to model an entire organ, an entire organism, or even an entire population – and to distinguish it from 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 multi-fold. We really want to understand the system and to gain deep comprehension of how it works and of how it behaves over time, but we also want to predict its future behavior under varying circumstances, often ones that haven't yet been actually tried out in the laboratory.

It is obvious that comprehensive modeling, if carried out successfully, can yield very 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.

The notion of *realistic* modeling is a key issue, and it is addressed throughout the lecture. To be realistic, a model must capture not only the overall viable stochastic behaviour of the system as a whole, but also the behaviour of the individual entities and their inter-relationships, their cooperation and their influence on each other. In fact, it is best if the model is such that the overall emergent picture be the result of the combined behavior of the individually modeled entities. A realistic model must be fully executable, which is more than carrying through a probabilistic computation of projected average case behaviour, or doing queuing theory analysis of probable outcomes. Executing the system is not just producing the end results, say, in the form of the probability of some event at the end of computing a Markov chain. Rather, we want the ability to execute the "program" of the system, which, just like running any computer program, can be done on various inputs, in a one-step-at-a-time debugging fashion, in ways that highlight the behaviour of individual pieces, in ways that take into account the probability distribution of inputs and of certain decisions made in the process, in best and worst case fashion, and indeed in typical average cases too. Thus, model execution should be the true analogue of running a conventional computer program, and model analysis is the analogue of verification, validation and complexity analysis.

Another aspect of the realism of the 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 when modeling biology, and in the present case perhaps health care researchers, hospital officials, and decision makers) to be able to model themselves or, at the very least, to comprehend and modify existing models. Thus, heavy use of differential equations or operations research theories and techniques in the modeling has the added disadvantage of being unfitting for use, or even modification by these experts, and indeed it can easily alienate them.

In way of illustrating the "realistic" facet of modeling, the lecture describes the general approach to modeling taken by our group. It is based on viewing the biological artifacts to be modeled as *reactive systems*[1], and to use for their modeling and simulating *visual formalisms*.[2] 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*.[3] 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). 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 variants of our approach is state-based, encouraging an *intra-object* style of specification, and uses the language of *statecharts*[4] to describe the system's behaviour by objects. One powerful tool supporting this is *Rhapsody*,[5,6] but there are many statechart tools. (Matlab has also adopted statecharts for its discrete aspects, in its *StateFlow* tool.) Another, more recent variant is scenario-based, and *inter-object* in spirit. It uses the language of *live sequence charts* (LSCs),[7] and allows one to play in the behaviour directly from the system's GUI and to then play it out just as if it were an intra-object model.[8] 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 I proposed a few years ago to the computer science and systems biology community,[9] from which this paragraph and the next one are adapted. The challenge is to fully model an entire multi-cellular organism. We actually have a particular organism 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 behaviour 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 next part of the lecture describes briefly two modeling projects that we have been carrying out; one using the state-based intra-object approach and the other using (mainly) the scenario-based inter-object approach. The first project involves T-cell development in the thymus,[3,10] and shows thousands of cells entering the thymus, struggling and competing for the prize if becoming fully-fledged T-calls. This model was the motivation for developing reactive animation, and uses Flash linked with Rhapsody and its statecharts.

The second project involves vulval cell fate determination in the C. elegans nematode,[11,12] and its key players are six vulval precursor cells who have to decide which of them gets the honour of working with a special anchor cell to form the worm's vulva, which is its egg-laying venue. This model was built mainly from LSCs using the Play-Engine, but we have also done some verification work of cell mechanistic behavior against lab observations, using LSCs and statecharts.

At this point, I propose a Grand Challenge for this community. The challenge – in full analogy with the challenge for modeling biology[9] – is to model a complete health care system, fully and realistically. This could be "merely" an entire hospital, but my feeling is that it should be larger: perhaps the complete hospital system for a region or a state. It could, and possibly should, also include (or at least solidly interface with) other relevant entities, such as governmental health offices, medical schools, health insurance companies, etc. This kind of challenge – again, in full analogy with modeling a biological organism – is very long term and incredibly complex and might never be achieved. However, it also enjoys the same potential benefits, i.e., providing an unparalleled understanding of a vast system of relevance. If achieved, such a challenge will no doubt result in new ideas, predictions, and recommendations, that could help improve the overall quality of health care. Interestingly, truly grand challenges often yield significant advances even if they are not successful, simply by the massive amounts of work that come from the talent, energy, money and dedication concentrated around them.

The final part of the lecture addresses the particularly interesting question of how we know when we are done. Or, in other words, when is a comprehensive, realistic model deemed complete, or valid? Here I propose a sort of Turing test, 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.[13] There are many issues that have to be addressed for such a test to be even conceivable, such as the "buffer" that has to be set up to prevent the interrogating team from knowing the difference simply by peripheral things like sight and smell or the time difference between a computerized model answering a query and a lab experiment set up to do the same.

Of course, this test is perhaps too wild and farfetched, almost imaginary, but it deserves discussion because it does try, just like Turing's original test for computerized intelligence[14] to put an upper bound on what is needed for comprehensive modeling to be complete. 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.

Bibliography

1. Harel D, Pnueli A. On the Development of Reactive Systems. In: Apt KR, editor. Logics and Models of Concurrent Systems.New York: Springer-Verlag; 1985. p. 477-98.
2. Harel D. On Visual Formalisms. Communications of the ACM 1988;31:514-30.
3. Efroni S, Harel D, Cohen IR. Reactive animation: Realistic modeling of complex dynamic systems. Computer 2005 Jan;38:38-47.
4. Harel D. Statecharts: A visual formalism for complex systems. Sci Comput Program 1987;8:231-74. (Preliminary version: Technical Report CS84-05, The Weizmann Institute of Science, Rehovot, Israel, February 1984.)
5. Harel D, Gery E. Executable object modeling with statecharts. IEEE Computer 1997 ;30:31-42.
6. I-Logix web site. http://www.ilogix.com
7. Damm W, Harel D. LSCs: Breathing life into message sequence charts. Formal Methods in System Design 2001;19(1):45-80. (Preliminary version in Proc. 3rd IFIP Int. Conf. on Formal Methods for Open Object-Based Distributed Systems (FMOODS'99), (P. Ciancarini, A. Fantechi and R. Gorrieri, eds.), Kluwer Academic Publishers, 1999, pp. 293-312.)
8. Harel D, Marelly R. Come, Let's Play: Scenario-Based Programming Using LSCs and the Play-Engine. Springer-Verlag; 2003.
9. Harel D. A grand challenge for computing: Towards full reactive modeling of a multi-cellular animal. Bulletin of the EATCS 2003;81:226-35. (Reprinted in Current Trends in Theoretical Computer Science: The Challenge of the New Century, Algorithms and Complexity, Vol I, Paun, Rozenberg and Salomaa, eds., World Scientific, pp. 559-68, 2004.)
10. Efroni S, Harel D, Cohen LR. Toward rigorous comprehension of biological complexity: Modeling, execution, and visualization of thymic T-cell maturation. Genome Research 2003;13:2485-97.
11. Kam N, Harel D, Kugler H, Marelly R, Pnueli A, Hubbard E.J.A, Stern M.J. “Formal Modeling of C. elegans Development: A Scenario-Based Approach”. Proc. 1st. Int. Workshop on Computational Methods in Systems Biology (ICMSB 2003), Lecture Notes in

Computer Science, Vol. 2602, Springer-Verlag, pp. 4–20, Feb. 2003. (Revised version in Modeling in Molecular Biology (G. Ciobanu and G. Rozenberg, eds.), Springer, Berlin, 2004, pp. 151–173.)
12. Fisher J, Piterman N, Hubbard EJA, Stern MJ, Harel D. Computational insights into Caenorhabditis elegans vulval development. Proceedings of the National Academy of Sciences of the United States of America 2005 Feb 8;102(6):1951-6.
13.Harel D. A Turing-like test for biological modeling. Nature Biotechnology 2005;23:495-6.
14. Turing AM. Computing Machinery and Intelligence. Mind 1950;59:433-60.