A Grand Challenge for Computing: Towards Full Reactive Modeling of a Multi-Cellular Animal[1]

David Harel
The Weizmann Institute of Science

Acknowledgements

I would like to thank my students, Naaman Kam and Sol Efroni, and my colleagues Michael Stern, Jane Hubbard, Amir Pnueli and Irun Cohen, for their insights and comments. This challenge is really a direct outcome of our interaction, and many of the points made herein are outcomes of discussions with them. In particular, Naaman Kam deserves special thanks for initiating the group's joint work on C. elegans, and thus causing that particular organism to be the focus of the challenge proposed here.

Abstract

Biological systems can be modeled beneficially as reactive systems, using languages and tools developed for the construction of man-made systems. Our long-term aim is to model a full multi-cellular animal as a reactive system ---- specifically, the C. elegans nematode worm, which is complex, but very well-defined in terms of anatomy and genetics. The challenge is to construct a full, true-to-all-known-facts, 4-dimensional, fully animated model of the development and behavior of this worm (or of a comparable multi-cellular animal), which is easily extendable as new biological facts are discovered.

1. Introduction

One of the central issues in software and system engineering over the last couple of decades has been to develop languages, methods and tools for the reliable construction of *reactive systems*, those whose complexity stems not necessarily from complicated computation but from complicated reactivity over time; see [HP85]. Such systems are most often highly concurrent and time-intensive, and can exhibit hybrid behavior that is predominantly discrete in nature but has continuous aspects too. Their structure consists of many interacting, often distributed, components. Very often the structure itself is dynamic, with its components being repeatedly created and destroyed during the system's life span. The heart of the problem is the need for good approaches to modeling and analyzing the dynamic behavior of such systems.

Our proposal has three premises: (i) that satisfactory frameworks now exist for reactive system modeling and design; (ii) that biological research is ready for an extremely significant

[1] An early version appeared as a position paper in the *UK Workshop on Grand Challenges in Computing Research*, November 2002. Final version appeared in *Bulletin of the European Association for Theoretical Computer Science* (*Bull. EATCS*) **81** (2003), 226-235. 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. 559-568, 2004.

transition from analysis (reducing experimental observations to elementary building blocks) to synthesis (integrating the parts into a comprehensive whole), a transition that requires mathematics and computation; and (iii) that the true complexity of biological systems --- specifically multi-cellular living organisms --- stems from their reactivity.

As far as the third of these goes, biological systems exhibit the characteristics of reactive systems remarkably, and on many levels; from the molecular, via the cellular, and all the way up to organs, full organisms, and even entire populations. It doesn't take much to observe within such systems the heavy concurrency, the event-driven discrete nature of the behavior, the chain-reactions and cause-effect phenomena, the time-dependent patterns, etc. From these three premises there follows a thesis: that biological systems can be modeled beneficially as reactive systems, using languages and tools developed in computer science for the construction of man-made systems.

Our long-term aim is to model a full multi-cellular animal as a reactive system. 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 organism 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.) The grand challenge --- which we estimate to require at least 10 years of work by many groups with diverse backgrounds--- 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 (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.

For the next 2 or 3 years, we have several topics of specific research that we are planning, all aimed at advancing the ultimate long-term goal. They include finishing the project of modeling the cell fate acquisition process of the *vulval precursor cells* (VPCs) --- part of the vulval development in C. elegans --- which we are in the midst of (see [K+03]), and then extending this to the full egg-laying system, possibly using intra-object statecharts in Rhapsody. We also plan to launch into new territory in C. elegans modeling, particularly modeling nerve cells (there are 302 all in all, and only 2 of them in the relevant part of the vulval development system --- but with extremely complex behavior) and incorporating into our models also intra-cellular-level behavior, and not only inter-cell-level behavior. We will also make great efforts to achieve distributed modeling, using reactive system tools that talk to each other in order to merge parts of the behavior of the C. elegans modeled separately in different labs. We also want to improve a technique we have developed recently, called *reactive animation,* whereby the reactive systems modeling tool (e.g., Rhapsody with the Statecharts language, or the Play-Engine with the language of LSCs) is linked up with an animation tool (e.g., Flash or Director) to yield a realistic animation of a reactive system (see [EHCb]), and to apply it to the models we will have built, namely the full C. elegans egg-laying system.

The reasons for starting the long-term project with the egg-laying system, are (i) it is one of the most well studied developmental systems in C. elegans, and (ii) it is a micro-cosmos of biological development, involving such major challenges as cell differentiation, cell-cell interactions (both long range and direct), cell proliferation, cell migration, axonal guidance, separate tissues (each of which starts to the develop independently of the other) forming a single functional system, and synchronization and timing of events.

The proposed long term effort could possibly result in an unprecedented tool for the research community -- not only those working on the particular worm being modeled. It would make it possible to ask questions and perform cyber experiments before taking them to the lab bench, and to compare the bench results with what we think we know. It would help confront open questions, which, because of their complexity, cannot be addressed by standard laboratory techniques alone. A model of this depth and flexibility would also serve as a basis for comparisons not only within a species, but with evolutionarily related species that have different forms and behaviors.

Also, if this long-term modeling challenge can be met for a worm, for which we have lots of information, it could actually tell us a lot about what we would need to make reasonable models for other less-defined and far more complex organisms (much in the way that the genome project used the worm as a test case for sequencing the human genome). Obviously, this kind of grand-scale work will impact not only biology, but also computer science, by revealing limitations of the computational tools and motivating further development, in order to modify, extend and adapt them for such use. We feel that much research in *systems biology* will be going this way in the future: grand efforts at using the methods and languages of computerized reactive systems for understanding sophisticated biology.

2. Scientific and technological background

One of the most widely used frameworks for developing reactive systems feature *visual formalisms* [H88], which are both graphically intuitive and mathematically rigorous, supported by powerful tools that enable full model executability, linkable to GUIs and other structural descriptions of the system under development. This enables realistic simulation prior to actual implementation. 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 being linked to tools for dealing with the system's continuous aspects in a full hybrid fashion. Most of the tools are state-based, encouraging intra-object style specification, using the language of *statecharts* [H87]. The most powerful of these tools is *Rhapsody* [HG97,I].

More recently we have developed an inter-object, scenario-based approach to reactive system specification. It is based on the language of *live sequence charts*, LSCs [DH99], and features playing in the behavior directly from the GUI and being able to then play it out just as if it were an intra-object model; see [HM03a,HM03b]. The tool supporting the *play-in/play-out* approach is called the *Play-Engine*. Its relevance to modeling biological systems stems from the condition/result nature of many pieces of biological behavioral data, which is scenario-based and intra-object in nature. LSCs can specify scenarios of behavior that cut across object boundaries and exhibit a variety of modalities, such as scenarios that can occur, ones that must occur, ones that may not occur (called *anti-scenarios*), ones that must follow others, ones that overlap with others, and more. An LSC consists of a *prechart* and a *main* chart, the former triggering the execution of the latter, and it states that whenever the scenario in the prechart occurs (e.g., the user has flipped a switch), the scenario in the main chart must follow it (e.g., the light goes on). Thus, we get a dynamic condition/result rule.

Play-in enables people who are unfamiliar with LSCs to specify system behavior using a high level, intuitive and user-friendly mechanism. The process asks that the user first build the graphical user interface (GUI) of the system, with no behavior built into it. The user then 'plays' the GUI by clicking the graphical control elements (in electronic systems these might be buttons, knobs, and so on), in an intuitive manner, in this way giving the engine sequences of events and actions, and teaching it how the system should respond to them. As this is being done, the Play-Engine continuously constructs the corresponding LSCs automatically.

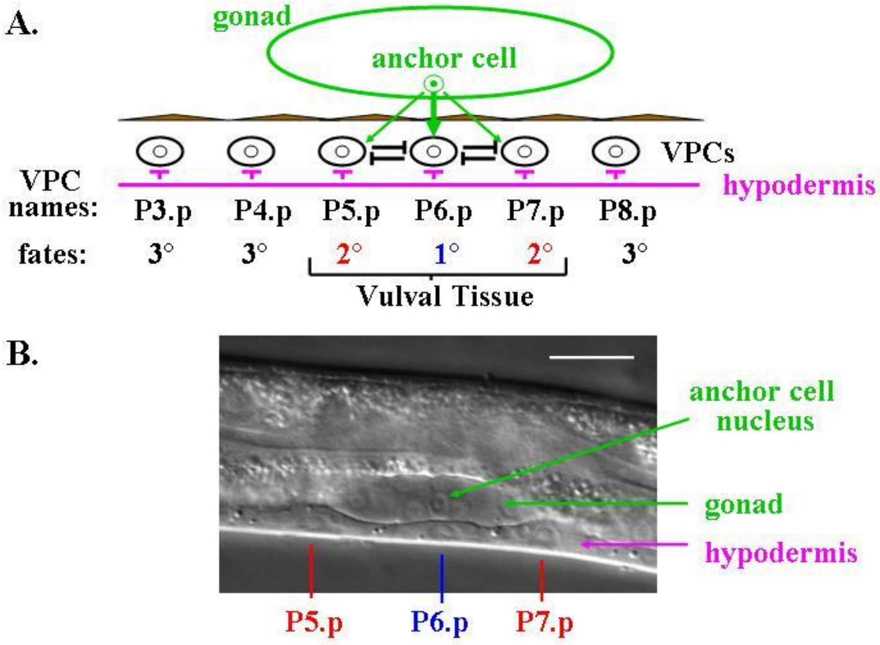
While play-in is the analogue of writing programs, play-out is the analogue of running them. Here the user simply plays the GUI as he/she would have done when executing the real system, also by clicking buttons and rotating knobs, and so on, but limiting him/herself to end-user and external environment actions. As this is going on, the Play-Engine interacts with the GUI and uses it to reflect the system state at any given moment. The scenarios played in using any number of LSCs are all taken into account during play-out, so that the user gets the full effect of the system with all its modeled behaviors operating correctly in tandem. Any violations of constraints or contradictions between scenarios will be detected if attempted. This kind of sweeping integration of the specified condition-result style behavior is most fitting for biological systems, where it can often be very difficult to connect the many pieces of behavioral data that are continuously discovered or refined.

In earlier work on T-cell reactivity (the cellular level; see [KHC01]), we have addressed the feasibility of modeling biological systems as reactive systems and the results were very encouraging. We used statecharts with Rhapsody. Since then, we have turned to two more complex systems. One is modeling T-cell behavior in the thymus (uniform multi-cellular level; see [EHCa]), and the other is on VPC fate acquisition in the egg-laying system of C. elegans (see [K+03]).

The thymus is an organ in which thymocytes develop to become mature T cells, which are an important part of the immune system. We have modeled the behavior of this system, and especially that of thymocytes in their journey within the thymus, using statecharts in Rhapsody. The Rhapsody simulation is very difficult for a biologist to understand --- it being but a diagrammatic simulation, showing the generation of instances, the switching between states, the events that have been consumed, the events that are about to be consumed, and so on. This representation is detailed and rigorous, and it does help in understanding what the thymus is doing, but not nearly enough. To alleviate this problem, we built a Flash front-end that visually resembles the way cells actually appear when viewed through a microscope, and which animates the course of events in the thymus. It shows the viewer how cells move, change their receptors, interact with other cells, proliferate, mature, die and secrete different substances. All the processes occur within the Rhapsody model, and can, in principle, be viewed in the Rhapsody simulation itself, but it is the link between Rhapsody and the Flash animation that is crucial here. Recorded demos for this system appear in [T].

Turning to the second piece of preliminary work, the C. elegans nematode is very well-defined in terms of anatomy and genetics. In fact, its anatomy is better defined than any other multi-cellular organism, and it is the only creature for which the entire somatic cell lineage is known from the one-cell embryo to the adult worm. This was possible since the worm is transparent, its cell lineage is invariant (i.e., does not change from worm to worm), and the total cell number is relatively small (959 for the hermaphrodite and 1031 for the male). Moreover the wiring diagram for its 302 neurons is known. Genetically, the genome is sequenced; the functions of many genes are known, and this data set is expanding rapidly. The project (carried out with two teams of expert C. elegans) is to model the vulval development of C. elegans, which is part of its egg-laying system. The vulva structure derives from three cells within a set of six vulval precursor cells (VPCs; named P3.p to P8.p). Vulval development was one of the first areas to which considerable effort was applied to achieve a molecular understanding of how cells acquire their particular fates that play the crucial roles in the development of the vulva. This system, though limited in cell number, is quite complex and ultimately integrates at least four different molecular signaling pathways in three different interacting tissue types. Cell fate acquisition in development also occurs in the context of cell cycle control and the general global controls on the growth and development of the organism. Hence, vulval development indeed represents a rather complex reactive system. The figure below shows some of the components of this system, diagrammatically as well as in an electron microscope image.

We have applied LSCs and the play-in/play-out methodology to this system, incorporating biological scenarios from several studies, including, for example, the 1986 data in[SH86], which reports on experiments where cells are destroyed using a laser microbeam (ablation). Some pre-recorded demos of the model in operation appear in [K]. Since signals among the VPCs and between the VPCs and adjacent tissues cooperate to determine the fates of these cells, these experiments test the nature of these interactions and the relative potential of the VPCs to adopt vulval (versus non-vulval) fates and, among those that adopt a vulval fate, the specific type of vulval fate. These data are used to infer specific properties of the events that occur in the wild-type situation and to generate a model for how the unperturbed system works based on the behavior of the perturbed system. In our modeling effort so far, we have addressed a number of issues that provide evidence for the flexibility and potential of this modeling approach. As mentioned earlier, much of the research discussed here will involve the continuation and enrichment of this model.



3. Methodology

The long term modeling of the entire C. elegans will require enormous amounts of work, both in setting up the computational framework --- with its reactive system modeling and execution engine, the analysis and verification tools, and the animation mechanisms --- and in gathering the biological data, translating it into reactive system terminology and incorporating it into the model. This would have to involve many teams of researchers, of multi-faceted backgrounds on both sides of bioinformatics. Our present, far more modest, proposal, also calls for a variety of methods and tools to be used.

The biological data of relevance to our vulval development modeling is to a large extent in the form of "condition-result" scenarios of behavior, that document the results of experiments conducted under specific conditions. These are then translated into rigorous LSCs by playing them into the Play-Engine through a GUI similar to the one shown in the figure above. Actually, we incorporate two levels of data into the model. One represents the "rules" of the behavior of the system, and are based on abstractions and inferences from various sets of primary data. The second level of data includes the "observations" that comprise the primary data itself. The set of observations utilized is a crucial component of the model, allowing both execution and verification, including consistency checks between the observations and the inferred rules. We will also use *smart play-out*, a more recent technique that employs powerful formal verification tools to drive the play-out and to compute ways to satisfy desired scenarios. Thus, we are able to ask true questions of the form "can this ever happen" and get the Play-Engine to show the way. The demo site [K] has some examples of this usage for the C. elegans model.

We also plan to incorporate the reactive animation ideas from our thymus work into the C. elegans model, so that we will not be limited to static GUI rendition of the behavior of the system, but will be able to exploit the full power of animation with the flexibility afforded by the link with a reactive modeling system. Thus we will build a *reactive animation* link from the Play-Engine to Flash, similar to the one we built from Rhapsody, as described in [ECHb] and shown in [T]. Following this, we will redo the front-end of the C. elegans model to be animated in 2-D or 3-D.

We also intend to build an alternative model of the egg-laying system using intra-object statecharts in Rhapsody, and use this to compare the two approaches. In addition, we plan to start modeling the neural activity in the relevant part of the egg-laying system of C. elegans. We shall be using similar methods, incorporating scenario-based information into the model via LSCs and the Play-Engine, and state-based data via statecharts and Rhapsody, but we will also investigate the relevance of neural-net modeling techniques to this.

Finally, we will attempt to achieve distributed modeling, using Play-Engines that talk to each other and Play-Engines that talk to Rhapsody. This will enable us to merge modeled behavior from many different laboratories, a must for attaining the goal of a full-scale model of the entire creature.

Bibliography

[DH01] W. Damm and D. Harel, "LSCs: Breathing Life into Message Sequence Charts", *Formal Methods in System Design* **19**:1 (2001). (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.)

[EHCa] S. Efroni, D. Harel and Irun. R. Cohen, "Towards Rigorous Comprehension of Biological Complexity: Modeling, Execution and Visualization of Thymic T Cell Maturation", *Genome Research,* special issue on Systems Biology, in press, 2003.

[EHCb] S. Efroni, D. Harel and Irun. R. Cohen, "Reactive Animation", ", *Proc. 1st Int. Symposium on Formal Methods for Components and Objects* (FMCO 2002), Lecture Notes in Computer Science, Springer-Verlag, 2003. to appear.

[I] http://www.ilogix.com

[H87] D. Harel, "Statecharts: A Visual Formalism for Complex Systems", *Sci. Comput. Programming* 8 (1987), 231-274. (Preliminary version: Technical Report CS84-05, The Weizmann Institute of Science, Rehovot, Israel, February 1984.)

[H88] D. Harel, "On Visual Formalisms", Comm. Assoc. Comput. Mach. 31:5 (1988), 514-530.

[HG97] D. Harel and E. Gery, "Executable Object Modeling with Statecharts", IEEE *Computer* 30:7 (July 1997), 31-42.

[HKMP02] D. Harel, H. Kugler, R. Marelly and A. Pnueli, "Smart Play-Out of Behavioral Requirements", *Proc. 4th Int. Conf. on Formal Methods in Computer-Aided Design* (FMCAD 2002), November 2002, pp. 378-398.

[HM03a] D. Harel and R. Marelly, "Playing with Time: On the Specification and Execution of Time-Enriched LSCs", *Proc. 10th IEEE/ACM Int. Symp. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems* (MASCOTS 2002), Fort Worth, Texas, 2002.

[HM03ab] D. Harel and R. Marelly*, Come, Let's Play: Scenario-Based Programming Using LSCs and the Play-Engine*, Springer-Verlag, 2003.

[HP85] D. Harel and A. Pnueli, "On the Development of Reactive Systems", in *Logics and Models of Concurrent Systems* (K. R. Apt, ed.), NATO ASI Series, Vol. F-13, Springer-Verlag, New York, 1985, pp. 477-498.

[K] http://www.wisdom.weizmann.ac.il/~dharel/Celegans/

[KCH01] N. Kam, I.R. Cohen and D. Harel, "The Immune System as a Reactive System: Modeling T Cell Activation with Statecharts", *Bull. Math. Bio.*, to appear. (Extended abstract in *Proc. Visual Languages and Formal Methods* (VLFM'01), part of *IEEE Symp. on Human-Centric Computing* (HCC'01), 2001, pp. 15-22.)

[K+03] N. Kam, D. Harel, H. Kugler, R. Marelly, A. Pnueli, E.J.A. Hubbard and M.J. Stern, "Formal Modeling of C. elegans Development: A Scenario-Based Approach", *Proc. Int. Workshop on Computational Methods in Systems Biology* (ICMSB 2003), February, 2003.

[R+97] D.L. Riddle, T. Blumenthal, B. J. Meyer and J.R. Priess (eds.): *C. elegans II* , Cold Spring Harbor Laboratory Press Plainview, NY (1997).

[S87] P.W. Sternberg, "Lateral inhibition during vulval induction in Caenorhabditis elegans", *Nature* **335** (1988), 551-554.

[SH86] P.W. Sternberg and H.R. Horvitz, "Pattern formation during vulval development in C. elegans", *Cell* **44** (1986), 761-772.

[T] http://www.wisdom.weizmann.ac.il/~dharel/ReactiveAnimation/demo.htm

[WP01] M. Wang and P. W. Sternberg, "Pattern formation during C. elegans vulval induction" *Curr. Top. Dev. Biol.* **51** (2001), 189-220