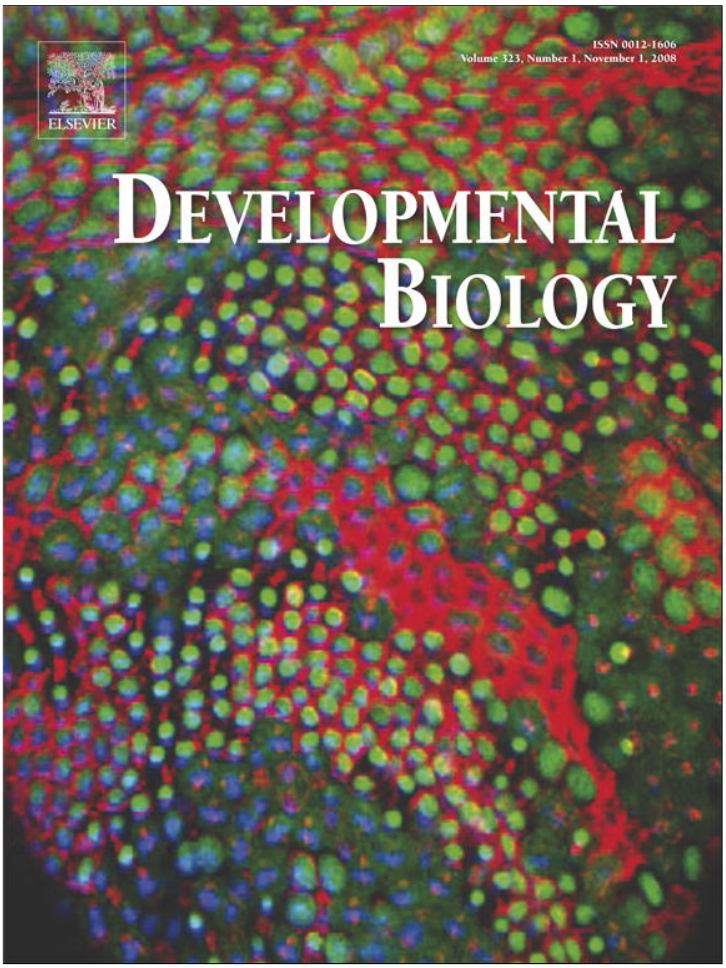


Developmental Biology 323 (2008) 1–5

Contents lists available at ScienceDirect

Developmental Biology

journal homepage: www.elsevier.com/developmentalbiology

A scenario-based approach to modeling development: A prototype model of *C. elegans* vulval fate specification

Na'aman Kam [a,1,2], Hillel Kugler [b,c,d,1], Rami Marelly [a], Lara Appleby [e], Jasmin Fisher [d], Amir Pnueli [a,c], David Harel [a,*], Michael J. Stern [e,*,3], E. Jane Albert Hubbard [b,f,*]

[a] *The Weizmann Institute of Science, Department of Computer Science and Applied Mathematics, Rehovot 76100, Israel*
[b] *New York University, Department of Biology, 100 Washington Square East, 1009 Silver Center, New York, NY 10003, USA*
[c] *New York University, Courant Institute of Mathematical Sciences, Computer Science Department, Warren Weaver Hall, Room 405, 251 Mercer Street, New York, NY 10012, USA*
[d] *Microsoft Research Cambridge, Roger Needham Building, 7 J J Thomson Avenue, Cambridge CB3 0FB, UK*
[e] *Yale University School of Medicine, Department of Genetics, SHM I-354, P.O. Box 208005, New Haven, CT 06520-8005, USA*
[f] *New York University School of Medicine, Department of Pathology, Developmental Genetics Program, Skirball Institute of Biomolecular Medicine, The Helen L. and Martin S. Kimmel Center for Stem Cell Biology, 540 First Avenue, New York, NY 10016, USA*

ARTICLE INFO

Article history:
Received for publication 21 July 2007
Revised 22 July 2008
Accepted 24 July 2008
Available online 31 July 2008

Keywords:
Vulval fate specification
VPC
Modeling
Scenario-based
Live Sequence Charts
C. elegans

ABSTRACT

Studies of developmental biology are often facilitated by diagram "models" that summarize the current understanding of underlying mechanisms. The increasing complexity of our understanding of development necessitates computational models that can extend these representations to include their dynamic behavior. Here we present a prototype model of *Caenorhabditis elegans* vulval precursor cell fate specification that represents many processes crucial for this developmental event but that are hard to integrate using other modeling methodologies. We demonstrate the integrative capabilities of our methodology by comprehensively incorporating the contents of three seminal papers, showing that this methodology can lead to comprehensive models of developmental biology. The prototype computational model was built and is run using a language (Live Sequence Charts) and tool (the Play-Engine) that facilitate the same conceptual processes biologists use to construct and probe diagram-type models. We demonstrate that this modeling approach permits rigorous tests of mutual consistency between experimental data and mechanistic hypotheses and can identify specific conflicting results, providing a useful approach to probe developmental systems.

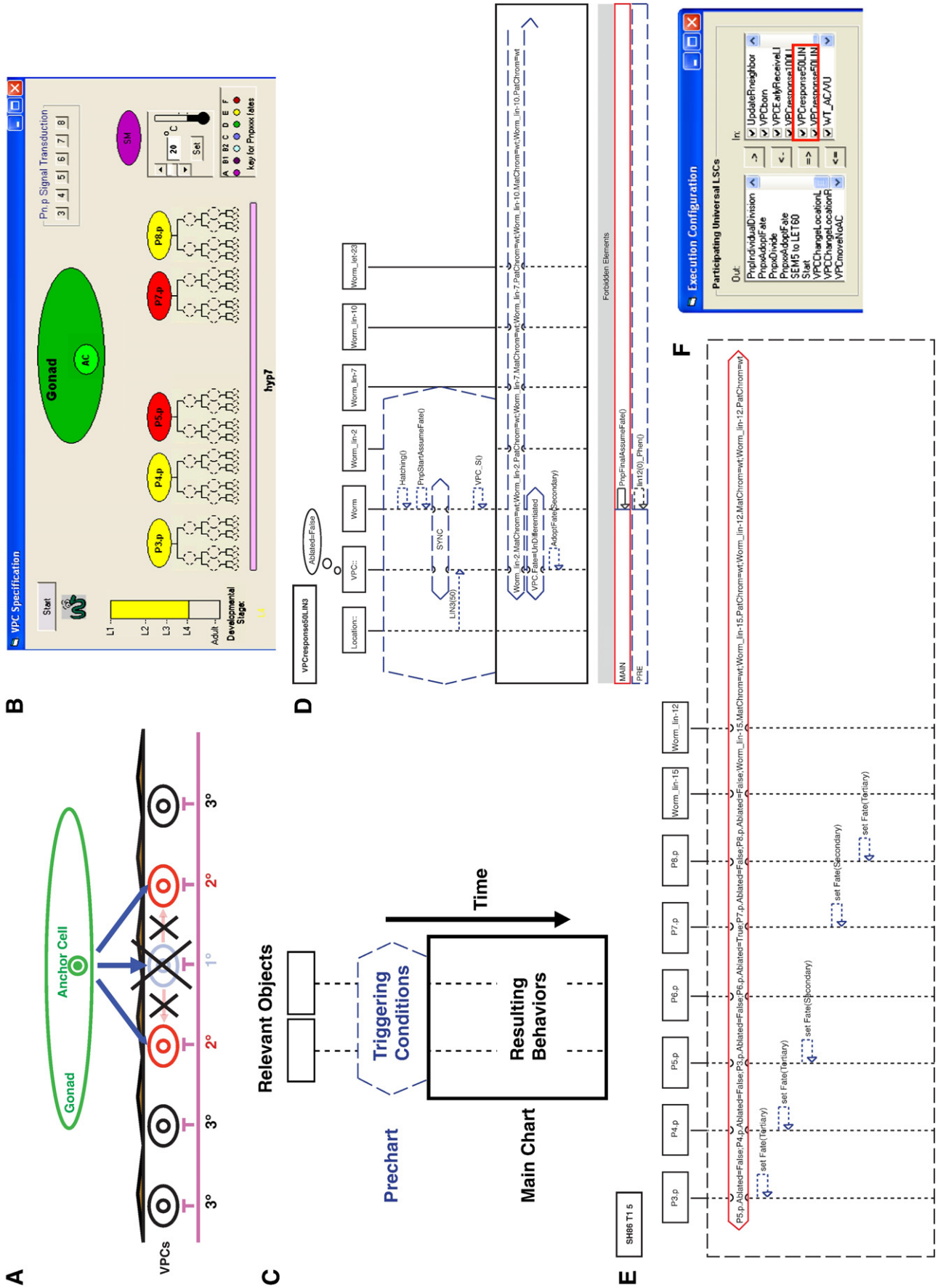
Introduction

Simple diagram "models" are used in experimental biology to summarize mechanisms inferred from detailed inter-related experimental results (e.g., see Fig. 1A). While executable computational models are becoming more prevalent, most models represent isolated aspects of what is known about a biological system, or they are geared to large scale data sets and limited in terms of the types of data they represent (for reviews, see de Jong, 2002; Ideker and Lauffenburger, 2003; Reeves et al., 2006). Moreover, the complexity of mathematical models makes them inaccessible to the average biologist to comprehend, use or extend further. Therefore, the vast majority of biological understanding is still represented using text and static diagrams, with dynamics and implications provided by human intuition. A methodology that can incorporate a system's dynamic behavior, expand the information-content of current diagrammatic models to include both its explicit and implicit contextual underpinnings, and formalize the semantics to make it computationally testable would tremendously enhance our current representations of biology.

Much information about biological systems derives from small-scale "reductionist" studies. This information is typically non-quantitative, compiled over time by multiple individuals using a variety of experimental approaches, and acquired and reported using non-systematic methods. This makes it relatively recalcitrant to conventional computational modeling approaches. Nevertheless, these data need to be represented in comprehensive models of biological systems. Here we present a computational modeling approach that facilitates the integration and analysis of diverse

* Corresponding authors. M.J. Stern is to be contacted at University of Central Florida, Department of Biology, 4000 Central Florida Boulevard, Orlando, FL 32816-2368, USA. Fax: +1 407 823 6442. E.J.A. Hubbard, New York University School of Medicine, Department of Pathology, Developmental Genetics Program, Skirball Institute of Biomolecular Medicine, The Helen L. and Martin S. Kimmel Center for Stem Cell Biology, 540 First Avenue, New York, NY 10016, USA. Fax: +1 212 263 0614. D. Harel, The William Sussman Professorial Chair, Department of Computer Science and Applied Mathematics, The Weizmann Institute of Science, Rehovot 76100, Israel. Fax: +972 8 934 6023.
E-mail addresses: Michael.Stern@aya.yale.edu (M.J. Stern), jane.hubbard@med.nyu.edu (E.J.A. Hubbard), dharel@weizmann.ac.il (D. Harel).
[1] These authors contributed equally.
[2] Current affiliation: The Weizmann Institute of Science, Department of Biological Chemistry, Ulman Building, Room 47, Rehovot 76100, Israel.
[3] Current affiliation: University of Central Florida, Department of Biology, 4000 Central Florida Boulevard, Orlando, FL 32816-2368, USA.

0012-1606/$ – see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.ydbio.2008.07.030



types of standard biological information. The graphical nature of both the interface (the GUI, see Fig. 1B) and the computational language itself (Figs. 1C–E; LSCs; Damm and Harel, 2001; Harel and Marelly, 2003) make this approach intuitive and user-friendly to biologists. To illustrate the approach, we have represented a portion of vulval precursor cell (VPC) fate specification in the nematode *Caenorhabditis elegans* (for review, see Sternberg, 2005).

Results

System design for software and system engineering seeks to represent all aspects of a modeled system. Similarly, our ultimate modeling goal is to represent all known aspects of a biological system (Harel, 2003). The model presented here is a comprehensive representation of virtually all of the information and experiments reported in three seminal papers (Sternberg, 1988; Sternberg and Horvitz, 1986, 1989) that helped establish the field of VPC fate specification in *C. elegans*. It represents and integrates different kinds of experimental results, including anatomical and genetic perturbations, and is a working prototype for an updatable model.

Our model is available to readers (see Supplementary materials), who are encouraged to download the model and use it to investigate its contents and run simulations. The Supplementary material accompanying this report contains a "User Guide" that describes how to manipulate the model. It also contains a set of movie files of recorded runs of simulations, showcasing the model without necessitating its download. Additional detailed explanations of the model and its testing can also be found in the Supplementary material.

A key attractive feature of our methodology is that it does not impose a computational way to re-think the biology. Instead, it uses the same conceptual process as the building and reason-based testing of static model diagrams that biologists are accustomed to. The building blocks of the model are "scenarios": "if–then" logic statements about the behavior or mechanistic basis of a limited "piece" of the system. The statements are time-constrained and have a precise syntax (that is, they are formal). This approach is particularly amenable to representing the understanding gained from reductionist analyses of biological systems. Each statement is captured in a Live Sequence Chart (LSC), which is a representation of conditions known to trigger a resulting behavior (Figs. 1C, D). The triggering conditions are represented in the "prechart;" resulting behaviors are represented in the "main chart." These modular descriptions of behavioral mechanism are linked by events and objects shared between LSCs.

A graphical user interface (GUI; Fig. 1B) serves as a dynamic visualization of the biology, and is used both in the construction of the LSCs (Play-In) and in simulations (Play-Out). LSC scenarios are not written by programming, but rather by actually performing the desired behavior using the GUI and menu-driven components (see User Guide in Supplementary material). Thus, the model can be modified and expanded by users with virtually no training in computer programming. Simulated perturbations/experiments are reproduced by the manipulation of objects (relevant cells and genes). The Play-Engine tool (Harel and Marelly, 2003) runs all aspects of our model.

Developmental time underlies the dynamics of the model (Kam et al., 2004). LSCs refer to a clock function correlated with developmental time, thus allowing developmental time to drive the progress of a simulation. The Play-Engine monitors all LSCs based on the state of the system, assessing which LSCs should be active and implementing events in their main charts when the requisite conditions are fulfilled. Events implemented by a main chart may, in turn, affect other precharts or main charts.

Mechanistic rule-based behavior and predictive power

The behavior of our model is controlled by a set of 86 *universal LSCs* (uLSCs) that specify the mechanistic rules inferred by the preponderance of existing data. Some uLSCs contain probabilistic events that generate the large number of possible outcomes seen *in vivo*. For example, ablation of certain VPCs allows other VPCs to occupy the vacated positions. Consistent with biological observations, the model produces alternative outcomes for specific ablations ("non-determinism"). Thus, simulations are not based on rote reproduction of experimental observations. Rather, they are based on mechanistic rules, explicitly stated as uLSCs. The model's predictive power comes from the fact that this general mechanistic rule can be used to execute and display the consequences of system perturbations (*in silico* "experiments") using a set of rules that are hypothesized to control the behaviors of the system. The uLSC "VPCresponse50LIN3" provides an illustration. uLSCs can define behaviors at different levels of detail, offering important flexibility (see the legend to Fig. 1D). Mechanisms that are well understood can be described in great detail, while those that are not as well understood – but are nonetheless important to drive simulations – can still be included. Additional mechanistic details can be added later without altering unrelated aspects of the model.

Fig. 1. The basic structure of Play-Engine-based LSC models. (A) Diagrammatic model representation of a specific experiment, the observed results, and the presumed perturbations of the mechanisms controlling vulval fate specification. The *C. elegans* hermaphrodite vulva forms from a set of six ventral epidermal/hypodermal blast cells known as VPCs (Sternberg, 2005). Under normal conditions, only three of these cells form vulval tissue: the cell that lies closest to the anchor cell in the overlying gonad acquires a primary (1°, blue) vulval fate, while the adjacent cells acquire a secondary (2°, red) vulval fate. The remaining three VPCs acquire a non-vulval, tertiary (3°) fate. LIN-3/EGF inductive signaling is depicted by blue arrows (strong signaling, thick blue arrow; medium signaling, thin blue arrows), LIN-12/Notch-mediated lateral signaling by red arrows. Purple "T" bars represent the LIN-15/SynMuv-controlled effects on VPC fate specification. In this experiment performed in a wild-type background, laser ablation of the presumptive primary cell did not result in its replacement by either of the adjacent cells. A simple interpretation of this experiment is that the effects of the lateral signaling of normal primary cells would be abolished. (B) An image of the Graphical User Interface (GUI) obtained at the end of one simulation of the same experiment described above, the one that matches the result shown in panel A. This matches the result described in Table 1, lines 5,6,8 of SH86. As in the diagram model of panel A, primary VPC fates are depicted in blue, secondary fates in red. In the GUI, tertiary fates are depicted in yellow. (C) The general structure of a universal *Live Sequence Chart* (uLSC). The behaviors described in uLSCs drive simulated behavior. The execution of the behaviors described in the main chart of a uLSC is compelled if the system satisfies a set of pre-existing conditions and events contained in its "prechart." If and when the behavior specified by the prechart has been completed – that is, all specified events have occurred in the correct order and all conditions have been evaluated to true – the resulting behaviors described in the main chart must be successfully executed. Objects relevant to the scenario described by the LSC are portrayed in boxes at the top and are associated with independent timelines that run from top to bottom (arrow depicts the time axis). These objects refer to a set of model objects that are either displayed in the GUI (e.g. anatomical structures), represented as internal objects (such as genes and anatomical locations), or come from the Play-Engine itself (such as the timeclock and the user/experimenter). (D) An example of a uLSC ("VPCresponse50LIN3") that corresponds to the thin blue arrows depicted in panel A. This uLSC assigns a 2° fate to VPCs that are exposed to an intermediate level (50) of the LIN-3 inducing signal. The objects represented in this uLSC include the genes from the core papers that mediate this mechanism, a generic VPC object, and a "Location" that experiences an intermediate level of the secreted LIN-3 inductive signal. Various elements of this uLSC describe the developmental time window and genetic background that allow this level of LIN-3 signal to induce a 2° VPC fate. (E) An example of an existential LSC (eLSC). This eLSC portrays the experiment reported in Table 1, lines 5,6,8 of SH86, in which P6.p is the only VPC that is ablated in a wild-type genetic background, and the remaining VPCs acquire the fates depicted in panel B. The dashed line surrounding the main chart (formally indicating its "existential" nature) indicates that the specified behavior is not universally binding for all runs of the system for which the conditions hold. Objects and timelines are similar to those found in uLSCs [see panel D]. The wild-type genotype for both *lin-15* and *lin-12* are explicitly stated in the experimental condition to eliminate the possibility that this chart could be satisfied by other mechanisms included in the model that are inappropriate for this specific experiment. (F) A portion of one of the *Execution Configurations* included in our model. The highlighted uLSCs, that are among those that are active in this Execution Configuration ("in"), describe the behavioral response of VPCs to a medium level of LIN-3 inductive signal [the thin blue arrows in panel A; one of these two uLSCs is shown in panel D]. Moving these two uLSCs to the inactive set ("out") creates a model that represents the Sequential Signaling Hypothesis. Moving uLSCs "in" and "out" of these sets is accomplished simply by clicking on the arrows between the lists of sets. Only a small portion of the sets of uLSCs is visible.

The model represents behaviors that influence VPC fate specification, either directly or indirectly. Direct influences include the establishment of the gradient of the LIN-3 inducing signal, a set of rules governing the movements of the VPCs following cell ablation experiments, and inductive and lateral signaling mechanisms. Details can be found in the Supplementary materials.

Model testing

A good working model can account for all the experimental observations from which its mechanistic rules were inferred. Working hypotheses represented by static diagrammatic models are typically tested using thought-based analyses. Computational models can be tested more systematically, matching the actual biological outcomes that result from specific experimental conditions to the outcomes of simulations that start with the same set of specific conditions.

In our methodology, a method called "play-out" allows simulations of system behavior under a set of *in silico* "experimental" conditions. Manual play-out allows the user to manipulate the system for a single run and directly observe the simulation. Batch-run play-out allows automated system runs for high-throughput testing, generating a number of different files that store the results of the simulations at various levels of detail (see Supplementary materials).

The Play-Engine is ideally suited for testing experimental outcomes against mechanistic hypotheses. During a simulation run, the Play-Engine tracks the states of all objects and traces all events. In performing this function, it activates the relevant LSCs and traces the progress of the events described in each LSC as they occur. Thus it can match the events driven by uLSCs during a simulation to a specific experimental result when the latter is described as an LSC. Experimental results are described using a second type of LSC: *existential LSCs* (eLSCs) (Fig. 1E). eLSCs differ from uLSCs in that they do not drive system behavior, but are monitored to determine whether a given simulation run of the system satisfies the statements they contain. Therefore, eLSCs do not have separate "condition" and "result" portions (Fig. 1E). We used a set of 260 eLSCs to represent essentially all of the actual experiments and results (table by table, line by line) reported in the core papers. Using the systematic testing capabilities of the Play-Engine, we have shown that our model can reproduce essentially all of the results observed for each experiment that was conducted in the core papers. An analysis of exceptions can be found in the Supplementary materials.

Representing alternative hypotheses

Biologists are often faced with more than one mechanistic hypothesis that appears to be compatible with the experimental data. Our modeling methodology easily represents alternative hypotheses. Each "Execution Configuration" in the Play-Engine's setup stores a specific subset of uLSCs that the Play-Engine will use during execution. Thus, different Execution Configurations can be used to include and exclude the specific uLSCs that make up the key mechanistic differences between alternative hypotheses, while leaving common elements of the model intact.

For example, determining the relative roles played by the inductive and lateral signaling mechanisms has been a long-standing issue in the study of VPC fate specification (see reviews by Sternberg, 2005; Sundaram, 2004). Fig. 1F highlights the two uLSCs that allow graded inductive signaling to influence the fates of the VPCs. Removal of these two uLSCs eliminates the differential response to inductive signal (the thin blue arrows in Fig. 1A). A similar small number of uLSCs allows the lateral signaling mechanism to promote sequential signaling. In addition to testing the complete model that incorporates all mechanisms, we similarly tested the model's behavior under only the "Graded" or only the "Sequential" signaling hypotheses (Sternberg and Horvitz, 1986) by defining two additional execution configurations. Of the experimental observations that can be reproduced by the combined model, our testing identified additional experimental outcomes that fail to be reproduced by these restricted "Graded" or the "Sequential" Execution Configurations (see Supplementary materials).

Discussion

The prototype model we present here was built to determine the extent to which our methodology can be applied to typical studies of developmental biology. The most important advantages this approach offers over the current reason-driven static models are: (1) visualization of explicit dynamic behavior based on a set of mechanistic rules; (2) the capability to follow multiple simultaneous events throughout a simulation; (3) systematic testing of all experimental results; and (4) incorporation of multiple data-types. Although the set of core papers represented is small and historically distant, subsequent progress within the field can now be modeled within the context of the early data, rather than being represented in isolation.

The ability to extend an existing computational model as new data become available is critical to the development of comprehensive models. The extendibility of this model is both its greatest strength and its future challenge. The challenge lies in the dramatic increase in complexity and scale as additional genes, alleles, processes and interactions are incorporated. Because of their modular nature, "scenario"-based descriptions of behavior are simpler to modify than non-scenario-based approaches. The addition of new data, or even paradigmatic shifts in our understanding, requires modification of only the affected scenario modules, and not a reconstruction of the entire model.

The generic nature of the Play-Engine tool will allow the translation of our modeling efforts to many other biological systems. New system-specific GUIs will allow similar representations of other systems, while the solutions we have found to represent the processes and behaviors of vulval fate specification should be applicable to similar aspects of other systems. Our current model can be extended and deepened to represent a growing proportion of this specific system, while also providing adaptable tools to represent other biological systems.

Acknowledgments

We gratefully acknowledge the contributions of D. Barak, and M. Yano to various aspects of this work. In addition, we thank R. Posner, L. Cooley, and K. Birnbaum for critical reading of early versions of the manuscript. This work was supported by collaborative NIH grant R24-GM066969, the Yale-Weizmann Exchange Program, and the John von Neumann Minerva Center at the Weizmann Institute of Science.

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.ydbio.2008.07.030.

References

Damm, W., Harel, D., 2001. LSCs: breathing life into message sequence charts. Form. Methods Syst. Des. 19, 45–80.

de Jong, H., 2002. Modeling and simulation of genetic regulatory systems: a literature review. J. Comput. Biol. J. Comput. Biol. 9, 67–103.

Harel, D., 2003. A grand challenge for computing: towards full reactive modeling of a multi-cellular animal. Bulletin of the EATCS 81, 226–235.

Harel, D., Marelly, R., 2003. Come, Let's Play. Scenario-Based Programming Using LSCs and the Play-Engine. Springer-Verlag, Berlin Heidelberg.

Ideker, T., Lauffenburger, D., 2003. Building with a scaffold: emerging strategies for high- to low-level cellular modeling. Trends Biotechnol. 21, 255–262.

Kam, N., et al., 2004. Formal modeling of *C. elegans* development: a scenario based approach. In: Ciobanu, G., Rozenberg, G. (Eds.), Modeling in Molecular Biology, Natural Computing Series. Springer-Verlag, Berlin Heidelberg.
Reeves, G.T., et al., 2006. Quantitative models of developmental pattern formation. Dev. Cell 11, 289–300.
Sternberg, P.W., 1988. Lateral inhibition during vulval induction in *Caenorhabditis elegans*. Nature 335, 551–554.
Sternberg, P.W., 2005. Vulval development. WormBook 1–28.
Sternberg, P.W., Horvitz, H.R., 1986. Pattern formation during vulval development in *C. elegans*. Cell 44, 761–772.
Sternberg, P.W., Horvitz, H.R., 1989. The combined action of two intercellular signaling pathways specifies three cell fates during vulval induction in *C. elegans*. Cell 58, 679–693.
Sundaram, M.V., 2004. Vulval development: the battle between Ras and Notch. Curr. Biol. 14, R311–R313.