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

Na'aman Kam*[1,2], Hillel Kugler*[3,4,5], Rami Marelly[1], Lara Appleby[6], Jasmin Fisher[1,5], Amir Pnueli[1,4], David Harel[1], Michael J. Stern[6, 7, 8] and E. Jane Albert Hubbard[3,8,9]

*These authors contributed equally

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

2 Current affiliation/address:
The Weizmann Institute of Science,
Department of Biological Chemistry,
Ulman Building, Room 47,
Rehovot 76100 Israel

3 New York University
Department of Biology
100 Washington Square East
1009 Silver Center
New York, NY 10003, USA

4 New York University
Courant Institute of Mathematical Sciences
Computer Science Department
Warren Weaver Hall, Room 405
251 Mercer Street
New York, NY 10012, USA

5 Current affiliation/address:
Microsoft Research Cambridge
Roger Needham Building
7 J J Thomson Ave
Cambridge CB3 0FB, UK

6 Yale University School of Medicine
Department of Genetics, SHM I-354
P.O. Box 208005
New Haven, CT 06520-8005, USA

7 Current affiliation/address:
University of Central Florida
Department of Biology
4000 Central Florida Blvd.

Orlando, FL 32816-2368, USA

8 Co-Corresponding authors:

Michael J. Stern

Tel: 407-823-2766

Fax: 407-823-6442

E-mail: Michael.Stern@aya.yale.edu

E. Jane Albert Hubbard

Tel: (212) 263-7154

Fax: (212) 263-0614

E-mail: jhubbard@saturn.med.nyu.edu

9Current affiliation/address:

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

Abstract

Studies of developmental biology are often facilitated by diagram “models” that summarize the current understanding of underlying mechanisms and their relationships. Given the increasing complexity of our understanding of development, there is a need for computational models that can systematically and rigorously test experimental data against prevailing mechanistic models. Here we present a prototype model of *C. 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 in 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. Model building, testing and analysis of simulations highlighted under-studied aspects of dynamic behavior of this system and gaps in our understanding. Experimental follow-up confirmed a previously untested hypothesis and led to the discovery that 1° vulval precursor cell fate acquisition occurs early in *lin-15* mutants. These studies indicate that this modeling approach could provide a useful means to probe developmental systems.

Keywords: vulval fate specification, VPC, modeling, scenario-based, Live Sequence Charts, C. elegans

INTRODUCTION

Simple diagram “models” are a mainstay of experimental biology and are used to summarize mechanisms inferred from detailed inter-related experimental results (e.g., Fig. 1A,B). Executable computational models are becoming more prevalent in biology. However, they usually either represent isolated aspects of what is known about a biological system, or they are geared to the representation of large scale data sets that are limited in terms of the types of data that are represented (for reviews, see de Jong, 2002; Ideker and Lauffenburger, 2003; Reeves et al., 2006). Moreover, the mathematical complexity of many models makes them inaccessible to the average biologist to comprehend, use or extend further. Thus, the vast majority of biological understanding is still represented using text and static diagrams, in which dynamics and implications are propelled by human intuition. While diagram models usually express useful information flow and logical relationships, they are “informal,” since the circles, arrows and bars typical of these models do not represent strictly unambiguous objects, relationships and events. These same diagrams are also poor representations of complex dynamic biology since they are limited in scope and are inherently static representations of dynamic events. Nonetheless, these static informal models continue to be a highly accessible and useful way for biologists to generate hypotheses and formulate experiments to test them. Therefore, a large gap exists between the way “reductionist” experimental biologists think about and “model” their results (diagrams) and the ways that modeling methodologies are being developed for genome-scale data representation and analysis (see also Sun and Zhao, 2004). Thus a methodology that can expand the information-content of current diagram-type modeling approach, formalize their semantics, and reflect a system's dynamic behavior would go a long way towards filling this gap.

Why do most biologists cling to static diagram representations of biological processes in an age of increasing computational sophistication? One deterrent is that much information from small-scale "reductionist" developmental studies is relatively recalcitrant to computational modeling approaches, since the results are typically non-quantitative, are compiled over time by multiple individuals using a variety of experimental approaches, and are acquired using non-systematic methods of data collection and reporting. Nevertheless, a vast amount of extremely valuable biological information of this type has informed our understanding of biological systems and needs to be represented in comprehensive models to obtain a holistic view of biological systems. Another deterrent is that traditional experimental biologists do not, by and large, have extensive computer programming skills. Therefore, a computational modeling approach that takes advantage of the information storage and retrieval capacity of computers in an intuitive and user-friendly way and simultaneously facilitates the integration and analysis of diverse types of standard biological information would constitute a major advance. Ideally, such approaches would be "formal" in the sense that they use mathematically or logically unambiguous statements that make them amenable to powerful testing approaches. In addition, such approaches could eventually incorporate parameter-based models (continuous and/or discrete), linking all relevant data about a particular system under investigation.

C. elegans vulval precursor cell (VPC) fate specification is a well-described process by which six cells of equal developmental potential interact with neighboring cells and with each other to establish a reproducible pattern of cell fates. The study of VPC fate specification proceeded through several overlapping stages, beginning with the analysis of cell lineage, subsequent cell ablation studies, and the isolation and molecular analysis of mutations that affect this process. In the last 30 years, this rich system has provided several key paradigms for the effects of cell-cell interaction on cell fate specification during development. In addition, components of highly-conserved signaling pathways and their interactions have been defined and elaborated using this system. Components of these signaling pathways are conserved in all animals, and their dysfunction has been linked to many human diseases, especially cancer. To date, over 80 genes are known to participate in vulval development and their interactions are still the subject of intense study (see Sternberg, 2005, and references therein).

The field of software and systems engineering routinely builds and analyzes exquisitely detailed dynamic models of working real-world systems in diverse industries such as aviation, telecommunications, and software development. Models for these applications are built and tested prior to implementation using specification languages that define “admissible” and “inadmissible” behaviors of the system. Among the conceptual distinctions made in specification languages is the distinction between “state-based” methods and “scenario-based” methods. The former represents system behavior by defining each object of the system and all of its possible behaviors/states whereas the latter represents the system’s behavior by defining “scenarios” or behavioral vignettes in which each of the objects participates. One advantage of “scenario-based” modeling over “state-based” modeling for applications to biology is that the expansion of models is achieved in a more intuitive way by adding or altering the “scenarios” consistent with advances addressed by new biological data.

We previously proposed that methods from system design can be used to render diagrammatic models of biology in a dynamic way (Efroni et al., 2003; Fisher et al., 2005; Kam et al., 2004), facilitating the integration and analysis of diverse types of biological information. Here, we used the scenario-based language of Live Sequence Charts (LSCs; Damm and Harel, 2001; Harel and Marelly, 2003) to build and test a prototype model of VPC fate specification in *C. elegans*. This work contrasts with a smaller-scale previous work that used state-based methodologies (Statecharts in Fisher et al., 2005; Reactive Modules in Fisher et al., 2007). Finally, the scenario-based approach also contrasts with more traditional mathematical approaches such as that used by (Giurumescu et al., 2006) who examined the influence of the coupling of two signaling pathways on the perception of inductive signaling and the ultimate phenotypic output of VPC fate using differential equations.

The scenario-based LSC language specifies the known or hypothesized behaviors of the objects of a system in a time-constrained manner (Damm and Harel, 2001; Harel and Marelly, 2003). For example, the language can succinctly and rigorously state that cell A must divide to form daughter cells A' and A'' within a specified time interval during development, contingent upon other conditions such as cell A assuming its proper prior identity. Information is coded in logic-based "scenarios", that are essentially descriptions of the behavior of limited parts of the system being represented. In the biological context, these could constitute small snippets of known "if-then" aspects of the system. For example, the statement above can be rendered as: *if* developmental time is within the specified window and cell A has assumed its correct identity, *then* cell A will divide to give rise to cells A' and A''. Each of these scenarios is depicted as an individual LSC, which is a formal “if-then” statement (that is, it can be converted to a formal logic statement) connecting triggering conditions to resulting behaviors (Fig. 2A). The trigger may be a condition or a behavior, i.e., an event or sequence of events. These “trigger-result” statements are encoded using a “prechart,” that specifies the triggering condition or event, and a “main chart,” that describes the compelled resulting behaviors (Fig. 2A,B). LSC-encoded scenarios are modular and are linked within the model by events that are shared between LSCs.

The scenario-based LSC model presented here is a comprehensive representation of virtually all of the information and experiments reported in three seminal papers that helped establish the field of VPC fate specification, along with additional selected data from other papers. It represents and integrates several different kinds of experimental results including both anatomical and genetic perturbations, and is a working prototype for an expandable and updatable model. This version of the model does not delve into the more recent molecular-genetic aspects of the field, though this is a natural future extension. The current model is driven by the relevant mechanistic rules of behavior that have been inferred from experimental observations and perturbations, and it can be run and analyzed dynamically (simulations) under varying “experimental” conditions. Although it represents only a small portion of the current state of knowledge in the field, this model is rigorously tested, executable, and expandable to include more recent developments. Analysis of this model led to several novel insights and to experimental validation of an underlying assumption. The model is publicly available for use, and the reader is strongly encouraged to manipulate the model itself to obtain a fuller understanding of its capabilities.

MATERIALS AND METHODS

The model was constructed using the object-oriented language of *Live Sequence Charts* (LSCs; Damm and Harel, 2001) supported by the Play-Engine tool (Harel and Marelly, 2003). The fundamental structure of this formal, graphical modeling language is scenario-based: descriptions of small pieces of time-constrained system behavior ("scenarios") are described in the "main chart" of an LSC (Fig. 2A). The language is sufficiently rich that these overlapping statements of time-constrained behaviors need not be mutually exclusive, nor need they be strictly deterministic. Several charts can remain open until the events required to complete them have occurred somewhere within the run, and for each event, a probability and/or other conditional factors can be added to either the "if" or the "then" statements that determine events. As a result, depending on the particular simulation run, the same experimental set-up can yield different outcomes, as is observed in the laboratory experiments. These features are more easily understood in the context of a simulation run (see below).

Extensive Supplementary Material accompanies this report, each of which are referred to in the relevant sections within RESULTS AND DISCUSSION below. Briefly, supplementary material includes access to the model itself (downloadable) together with a "Documentation Manual" that details the structure of the model (including each Use Case and the LSCs within it) and the "User Guide" that indicates how to manipulate and expand the model. Supplementary movie files contain recorded runs of simulations. These materials provide valuable "hands-on" experience with the model. A Supplementary Materials and Methods file includes details on implementation of VPC fate specification in the model, model testing, and experimental procedures relevant to Figure 8. Supplementary Table 1A-C relevant to model testing is also provided.

RESULTS AND DISCUSSION

A key attractive feature of the methodology is that it does not impose a computational way to re-think the biology, but rather uses the same conceptual process as the building and reason-based testing of static model diagrams. We will draw attention to these parallels as we describe features of this modeling approach and our model. We begin with (1) the rationale for the modeling approach and (2) its specific aspects that make it particularly well-suited for representing biological systems. We then present the model structure: the language and the tool in which it is built and run, and then its parts (objects and their behaviors). We follow with descriptions of a sample simulation run, mechanisms we used to implement VPC fate specification, how events are integrated in the model, and testing features of the model. In the context of testing, we describe several key uses of the model: testing experimental results against a specific mechanistic hypothesis and against competing hypotheses. Finally, we provide examples of biological insights gained from the modeling and experimental follow-up to an observation made from model simulations. Readers who are not interested in the model details may prefer to skip to the sample simulation or to later sections detailing testing applications of the model that are of particular use to developmental biologists.

Rationale for the modeling approach

Having previously noted the parallels between system design and the analysis of biological processes (Efroni et al., 2003; Fisher et al., 2005; Kam et al., 2004), we sought to use computational tools from software and systems engineering to create a model of a biological system that has a number of goals that distinguish it from most existing computational modeling approaches to biology. As in system design, the ultimate goal is to represent all known aspects of a biological system. Thus, we required that our model be capable of representing the diverse processes involved and the varied types of data that pertain to the system. Second, we sought a modeling methodology in which both the inferred mechanistic rules of behavior as well as the underlying data could be represented formally (in the logical/mathematical sense of unambiguous), to allow the mechanistic model to be systematically and rigorously tested to assess whether it can reproduce experimentally observed behaviors. Third, we wished to represent systems whose behaviors are incompletely understood. If a model is useful despite the presence of "black boxes", it may be well-suited to modeling biology since a problem that often hinders modeling is the presence of significant gaps in our understanding of all aspects of a biological system. Fourth, it was critical for our modeling methodology to allow relatively easy expandability. Finally, we sought tools that would be accessible to and usable by biologists. This allows the biological experts most familiar with the system to help in the construction and analysis of the model.

The structure of the LSC language closely matches the way biologists probe and understand biological behaviors (Kam et al., 2004). Biological entities subjected to specific sets of conditions will respond in defined ways within discrete windows of time. Experimental perturbations alter the conditions in well-defined ways, resulting in changes in the resulting observed behaviors. The inter-object representation ("scenarios" that relate specific events and the relevant objects) also matches well our description of biological systems. Both experimental and mechanistic biological descriptions of behavior refer to the specific interactions of various biological components within a system. This allows considerable flexibility: only relevant behaviors are modeled. By contrast, an intra-object representation, such as used in the language of *Statecharts* (Harel, 1987), encourages describing a system in terms of all possible behaviors for each of its objects.

The LSC language and biological modeling

Several additional practical aspects of the LSC language (implemented using the Play-Engine tool (Harel and Marelly, 2003)) are well-suited for biological modeling. First, the model (including much of the coding of LSCs) is built, manipulated, and run by way of a graphical user interface (GUI) that resembles a pre-existing diagram model of the system (Fig. 1 A,B and E,F; see Sternberg and Horvitz, 1986). The GUI reflects the states of the objects it represents both during model construction and simulation, thereby serving as a dynamic visualization of the biology. This feature permits model construction and running by people without programming language skills. Second, the richness of the LSC language permits the representation of complex biological systems: the basic condition-result structure can contain, among other things, must ("hot") and may ("cold") stipulations, temporal constraints, and the specification of forbidden behaviors and probabilistic choices based on information contained in other LSCs (Fig. 2B; see also below). Because the language includes temporal statements, the progression of time can drive model simulations. Third, the scenario format parallels informal descriptions of biological information based on reductionist approaches, enabling a straightforward formalization of biological analyses.

This modeling paradigm has many additional advantages: (a) it is intuitive to biologists; (b) it can incorporate many different types of information, both quantitative and non-quantitative; (c) it can handle both experimental observations and mechanistic hypotheses; (d) it is not stymied by unknown parameters; (e) it is expandable; and (f) it can highlight novel testable hypotheses that were not easily gleaned by intuition alone.

*The "*Core*" model*

We wished to determine whether the "scenario-based" modeling paradigm could handle a variety of different data types that contribute to the building of a typical mechanistic model in developmental biology. We chose to model information in three seminal papers from a now well-established field. Thus our prototype model can be seen as a test-case for this type of modeling paradigm for other emerging or established fields.

We constructed a model that describes and integrates the biology and experiments reported in three seminal papers, hereafter referred to as the "core" papers: "SH86", "S88", and "SH89" (Sternberg, 1988; Sternberg and Horvitz, 1986; Sternberg and Horvitz, 1989, respectively). These papers define the phenomenology of vulval fate specification that formed the basis of much of the subsequent work in the field. SH86 addresses cellular interactions and potential models of vulval fate specification, S88 provides evidence for lateral specification among the VPCs, and SH89 describes an analysis of genetic interactions between genes that were known at the time to affect VPC fate specification. Many genetic components pattern VPC fates, including a receptor tyrosine kinase (RTK)-mediated inductive signal (LIN-3) and a LIN-12/Notch-mediated lateral signal. Determining the relative roles played by these two pathways and their interactions at the level of specific pathway components remains at the forefront of this field (see reviews by Sternberg, 2005; Sundaram, 2004).

To represent the information in a way that simulates development and the accrual of knowledge and understanding in the field, the model also incorporates additional information, much of which was knowledge that existed at the time. Specifically, behaviors were incorporated from the description of the normal development of the vulva (Sulston and Horvitz, 1977), the initial data demonstrating regulative behavior between the VPCs (Sulston and White, 1980) and between the AC and the VPCs (Kimble, 1981), and the characterization of *lin-12* (Greenwald et al., 1983). These mechanisms include (a) inductive signal production by the anchor cell and gradient formation, (b) position-dependent response of the VPCs to the inductive signal, (c) VPC production of and response to a lateral signal, and (d) a *lin-15*-dependent hypodermal influence on vulval induction. In addition to these mechanisms that directly affect VPC fate specification, we also needed to represent a number of processes that have indirect effects: (e) the anchor cell/ventral uterine precursor cell (“AC/VU”) decision that determines the cellular source of the inductive signal; and (f) a non-deterministic choice of P3.p to participate in the vulva equivalence group (Chen and Han, 2001). Details of how these mechanisms are represented in LSCs are described in the Supplementary Materials and methods.

In addition, since the ultimate goal of our effort is to provide a core model from which to develop a comprehensive, contemporary dynamic model, we represented these critical early results and inferred mechanisms in a way that is largely consistent with our current mechanistic understanding of this system. Occasionally, this required us to make use of the post-1989 literature. Thus, for example, we included a mechanism reported in Ambros (1999) that uses a two-step cell-cycle gated response to *lin-12* to implement the influence of the lateral signal on vulval cell fates (Ambros, 1999). One major exception is that we did not model the more recent findings of Cui et al. (2006) regarding the role of SynMuv genes in preventing inappropriate expression of the inductive signal LIN-3 in the hypodermis. The intended expandability of this model will allow the vast repository of additional relevant data to be added to the model (step-by-step instructions for adding new LSCs are provided as part of the User Guide in the Supplementary Material).

Our model is comprehensive in the sense that it covers all types of behaviors (anatomical, cellular, genetic) described in the three papers. The model also allows the user to simulate 'in-silico' essentially all of the experiments described in the core papers, namely cell ablations and genetic perturbations. Despite its relatively small scale, we found this *Core* model is sufficiently complex to test specific desired features of the methodology, namely, the ability to (1) test the model against experimental results, and (2) easily generate alternate hypothetical models and re-test the data to “find” specific results that support or refute each hypothetical model. Furthermore, during the course of model building and testing, the model’s intuitive dynamic simulations highlighted areas of the model that are under-supported by the experimental data.

Model structure: the LSCs

The LSCs of this model represent the biological behaviors that govern how VPC fates are specified and experiments reported in the literature. The Play-Engine tool is the software that allows the model to work and implements the formal, mathematical (logical) relationships between the LSCs. Two types of LSCs, universal and existential, are used for different purposes in the model.

uLSCs

The behaviors described in one type of LSC, the *universal* LSCs (uLSC), can 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" (Fig. 2A). In this manner, each uLSC represents a small, formal statement of rules governing the behavior of the system. The net behavior of the modeled system is defined by the set of uLSCs that are interlocked by common events or objects. For example, one uLSC may state that if A happens, then B must happen. Event B could then force the occurrence of event C, if there exists another uLSC that states "if B happens, then C must happen". Figure 3 depicts a specific example of a sequence of events triggered by two interacting LSCs.

A Table of all uLSCs in the model and a brief description of their purpose is provided in Table 1; more detailed explanations of each LSC are provided in Supplementary "Model Documentation".

eLSCs

The other type of LSC is an *existential* LSC (eLSC). eLSCs differ from uLSCs in that they do not compel system behavior, but they can be monitored to determine whether a given simulation run of the system can satisfy the statements they contain. In an eLSC, both the "condition" and "result" statements are contained in the main chart (Fig 2C). In the biological context, eLSCs can be used to test whether a simulation driven by a given set of uLSCs can reproduce the outcome expected by experimental manipulations described in an eLSC. We have used eLSCs in the model to represent essentially all of the actual experiments and results (Table by Table, line by line) that are reported in the papers. A list of eLSCs in the model is provided in Supplementary "Existential Chart Documentation". See below (in "Testing") for an explanation of the use of eLSCs in model testing.

Within our model, LSCs are organized within Use Cases that group together descriptions of related behaviors. Figure 4A and 4B indicate Use Cases in which the LSCs are organized and depicts their higher-level relationship network, respectively. We will refer to LSCs by their "LSC Name(Use Case Name)". (See "Documentation Manual" in the Supplementary Material for each Use Case and the LSCs within it). During a simulation run, LSCs within and between Use Cases interact with one another to progress to the final output of VPC fate acquisition. LSCs in the Core Behaviors Use Case interact with or trigger LSCs in all other Use Cases, while LSCs related to the events of Pn.p fusion, for example, are affected by LSCs in the CoreBehaviors Use Case as well as by LSCs that relate to the Hyp7 Inhibitory signal. Thus the progression of a run is a complex web of interacting and interdependent LSCs, each of which describes a fragment of system behavior (see also below *Sample Simulation).*

The GUI

Both the graphical elements that make up the LSC language and the graphical user interface (GUI) contribute to making formal logical statements of behavior in this model understandable to biologists. The Play-Engine tool (Harel and Marelly, 2003) enables the construction of the GUI. The GUI reflects the states of the objects it represents during LSC construction (play-in) and during model simulation (play-out). It thereby serves as a dynamic representation of the biology that parallels the processes often represented in the current literature by static pictorial models (Fig. 1A,B and E,F; Supplementary Material). Starting with a visual representation of the system in the form of a GUI, LSC scenarios are not written by programming, but rather by actually performing the desired behavior via the GUI and menu-driven components. Thus, the model can be modified and expanded by a user with no previous training in a computer programming. A simple example of the construction of an LSC using the GUI is presented in Fig. 5A-L to illustrate this point. In this example, the GUI is used to write an LSC that specifies "if the cell P6.p adopts a 1° fate, it should send a lateral signal (LS) to P7.p that causes P7.p to adopt a 2° fate".

Model structure: Parts

The model is a set of objects and behaviors, both of which are described in turn below.

Objects

The perturbations used in the three papers to probe the mechanisms underlying vulval fate specification include laser ablation of specific cells and genetic mutations to compromise the function of specific genes. These perturbations are reproduced in this model by the manipulation of objects for each of the relevant cells and each of the relevant genes. A number of additional objects are used to represent the requisite temporal and spatial aspects of this development system.

GUI objects

The anatomy is depicted in the GUI (Fig.1 A,B), which includes objects representing the VPCs, the gonad, two AC/VU cells (corresponding to Z1.ppp and Z4.aaa) and the hypodermal syncytium, the site of action of *lin-15* (Herman and Hedgecock, 1990). Some additional objects have been included for future development, including one of the two sex myoblasts (SM), the complete vulval lineage, and a “thermometer” object that can be used to represent experiments performed at different temperatures.

Internal objects

By contrast to the anatomical objects represented in the GUI, genes are represented as internal objects (see Supplementary Material). Fifteen genes are in our model. Six of these are central to vulval fate specification and the behaviors of the core papers (*lin-3*, *lin-12*, *lin-2*, *lin-7*, *lin-10*, and *lin-15*). Two genes (*let-23* and *dig-1*) are not explicitly referred to in the three papers, but are included in the model for their central importance in structuring and analyzing the model. *let-23* encodes the receptor for the LIN-3 EGF-like (epidermal growth factor) inducing signal (Aroian et al., 1990); the papers refer directly to this receptor, whose existence is inferred by the observed behaviors of the system. It is also the substrate that is localized by the LIN-2, -7, -10 complex (Kaech et al., 1998). Although these components were not known to function as a complex at the time of the core papers, these are key genes in the three papers and are modeled in a way that is consistent with building the core of a current model. Mutations in *dig-1* cause the displacement of the gonad (Thomas et al., 1990), which provides a crucial test for vulval fate specification behaviors during model analysis throughout construction. Two additional genes, *lon-1* and *unc-84*, were used in the three papers for special analytical purposes. *lon-1* is represented in the model in its capacity to change the relationship between the positions of the VPCs and the AC. *unc-84* was used as a genetic means to ablate VPCs. The *unc-84* gene and its potential for mutation are included in the model as an internal object, but its behavior is not specified in the model in the form of any LSC. The User Guide (Supplementary Material) contains instructions on how to add an LSC to specify the *unc-84* mutant behavior. Instead, in the current model, in places where the *unc-84* mutation was used to genetically ablate VPCs, the cognate experiments were represented in the model in the same way as other cell ablation experiments (that are accomplished experimentally using a laser). Finally, five additional genes (*sem-5, let-60, mek-2, mpk-1* and *lin-45*) are included as internal objects. These genes encode components of the signal transduction pathway that acts downstream of the LET-23 EGFR (EGF receptor) as part of the induction pathway. Although these last five genes can not be manipulated in the current model, they are included for future expansion.

Genetic mutations can have a wide spectrum of effects on gene function. For many of the genes represented in this model, there are a large number of known alleles, each with their specific allele designation. In *C. elegans*, there are often alleles that are used as representative of particular types of perturbations of specific genes; these are referred to as “canonical” alleles. The canonical alleles referred to in the core papers are available in the model as genetic perturbations, as well as the wild-type (unperturbed) allele. These alleles are referred to by their official names, in order to be able to associate specific phenotypic effects with specific alleles. In certain instances, a large set of alleles behaves similarly (for example, the set of null alleles that eliminate gene function). To reduce the complexity of the model, we have included a procedure to map this set of alleles to their common phenotypic consequences (see Supplementary Material). An important area for future research is a computational method to map various allele behaviors (phenotypes) onto a scale or data-space relative to wild-type and bounded by the phenotypic behavior of null and hypermorphic alleles, together with the interaction of the gene of interest vis-à-vis other genes that impinge on the phenotype.

Behavior

Our model consists of 86 uLSCs (some of which contain probabilistic conditions that essentially multiply to the many additional outcomes seen in vivo) and 260 eLSCs that define condition-behavior statements of specific biological behaviors at different levels of detail. The flexibility regarding the level of behavior that is specified in the model is an important feature of the methodology. Mechanisms that are well understood can be described in great detail while those that are not as well understood – but that are nonetheless important to permit a working model and simulation of the system – can still be included. Because of the modular nature of scenario-based models, additional mechanistic details can be added later without altering unrelated aspects of the model. Examples from our model will serve to illustrate the ways in which various levels of behavior can be specified and the critical conceptual features of a flexible mechanistic rule-based model.

Developmental time and Model Progression

The progression of developmental time underlies the dynamic behavior of this model (Kam et al., 2004). The time period represented spans from hatching (L1 entry) to the completion of the divisions of the Pn.p cells by L4 entry. One uLSC ["Developmental Time 20c(Core Behaviors)"] describes the progression of time between these larval stages (Fig. 3). All other events are linked to this central uLSC either directly [for example, "PnDivide(Core Behaviors)", Fig. 3] or indirectly. The links between uLSCs are the events that are common between the main chart of one uLSC and the precharts of other uLSCs (for example, the event "L1entry" in Fig. 3). Thus, events compelled to occur in the main chart of one uLSC become the link to events specified in the other uLSC by being part of the triggering conditions. Although vulval fate specification occurs within a narrow window of time from the end of the second larval stage (L2) to the mid-L3 stage, earlier events are crucial for laying the groundwork for this process. Thus, for example, the events that determine which Pn.p cells participate in the vulval equivalence group (the set of VPCs) must be represented as well.

The Play-Engine incorporates the progression of time through the use of a clock function that can be referred to in the LSCs. Clock ticks can be set to advance automatically to mimic the normal progression of time. Within the LSCs, these clock ticks can be correlated to developmental time. In our model, with the exception of the first hour of the L3, clock ticks quantize time progression into 1-hour intervals. The first hour of the L3 is a crucial window of time for vulval fate specification, and fate prioritization requires a more refined quantization of time. Therefore, in the first hour of the L3, each clock tick represents 1 minute (see Supplementary Material). A "Developmental Stage" control has been included in the GUI to aid in the visualization of the progression of developmental time (see Fig. 1E,F).

Mechanistic rule-based behavior and predictive power

The behavior of the model is controlled by a set of 86 uLSCs that specify in a logically formal way the mechanistic rules supported by the preponderance of existing data. Thus, simulations that result from the running of the model are not based on rote reproduction of experimental observations themselves, but on the rules that have been inferred from the data by abstract reasoning. For example, the uLSC "VPCresponse100LIN3(VPC Fate Assumption)" describes the response of a VPC that receives a high level (a value we set arbitrarily to "100") of LIN-3 signal appropriate for the VPC immediately ventral to the AC, normally P6.p (Fig. 6D). This uLSC represents a mechanistic rule, since it does not refer to a specific result, but rather describes a more generic mechanism. Thus, the model's predictive power comes from the fact that it can be used to execute and display the behavioral consequences of system perturbations (*in silico* "experiments") using a set of rules that are hypothesized to control the behaviors of the system, rather than simply storing and retrieving experimental conditions and their direct cognate results (see also Testing, below).

The richness of the LSC language allows for representing complex biological situations. There are a wide range of types of behavior-constraining terms, including time constraints, genetic constraints and the occurrence of other events. For example, temporal conditions can be used to limit certain behaviors. This allows restricting events to certain stages of development or events to occur within a specified window of opportunity. Other conditions can restrict behaviors to appropriate genetic backgrounds. For example, the observation that VPCs do not adopt a 2º fate in the absence of *lin-12* is represented as a condition imposed on a particular set of uLSCs that describe behavior in a *lin-12(0)* background. Similar to the variety of levels of representation of the biology in mechanistic statements, these constraints may or may not be linked to a detailed mechanism, but permit many relevant behaviors to be modeled.

Major Components of Model Behavior: many different biological phenomena

The behaviors represented in the model are those that influence VPC fate specification, either directly or indirectly. Direct influences include the establishment of the LIN-3 gradient (LIN-3), a set of rules governing the movements of the VPCs following cell ablation experiments (Pn.p Movements, see below), inductive signaling (VPC Fate Assumption), lateral signaling (Lateral Signaling), and the *lin-15*-mediated "inhibitory signal" (Hyp7 Inhibitory Signal; the recent results demonstrating that the *lin-15*/SynMuv effect reflects ectopic LIN-3 expression in the hypodermis (Cui et al., 2006) can be incorporated into a later version of the model). Indirect influences include a simplified representation of the AC/VU decision in the gonad (AC/VU) and the fusion potential of P3.p (Pn.p Fusion). The AC plays an important role in *C. elegans* vulval fate specification, making its representation crucial to depict in this model. While much is known about AC development and function, only those aspects of its behavior that are relevant to vulval fate specification are included here (see Sadot et al., (in press)) for a model of AC fate acquisition). Thus, the presence or absence of an AC, the secretion of the LIN-3 inducing signal, and the relative positioning of the gonad/AC with respect to the VPCs are all represented in this model as early indirect influences on vulval fate specification. In addition to directly influencing VPC fate specification via lateral signaling, the role of *lin-12* in the AC/VU decision serves to showcase the ability of this methodology to aid in integrating distributed behaviors. The birth of both the VPCs [the division of P(3-8)] and the AC/VU precursors is also represented to aid the accurate representation of behavior via the GUI. Additional details of how these behaviors are represented are described in Supplementary Material. The mechanisms we used to implement VPC fate specification, the heart of this model, are overviewed in greater detail below.

VPC lineage representation

Though VPC fate specification can be represented at many different levels of description, for the majority of the experiments reported in the core papers, Pn.p fates are assigned based on the characteristics of the lineage pattern of each Pn.p cell (see Sternberg and Horvitz, 1986) and (Katz et al., 1995) for the rules used in our model). The model assigns fates within these categories, but does not associate them with specific lineages (note that the GUI is set up to be able to reflect the actual lineages in the future). Since no rules are known that govern the production of hybrid (mixed vulval/non-vulval) and intermediate (mixed 1º /2º) fates, these categories are not generated in the current version of this model. However, for purposes of completeness in model testing, eLSCs that represent hybrid and intermediate lineages were included (see below Testing; Supplementary Material).

Sample Simulation

How does a series of if-then statements or “scenarios” drive a model simulation (a “run” of the model) under specified starting conditions? As a simulation progresses, the Play-Engine monitors the Precharts of all participating LSCs. Once a given Prechart’s conditions are fulfilled, the Play-Engine generates the events contained in its main chart, which in turn may activate the conditions in Precharts of additional LSCs. This network of inter-related events captured in the charts forms a cascade of events that drives the simulation. A chart is “completed” once the events of its main chart are successfully executed. If one chart logically conflicts with another chart elsewhere in the set of participating LSCs, the user is alerted (either directly during a simulation or by virtue of the fact that the chart remains uncompleted at the end of a run). Many charts can be active at one time, in various states of completion, conceptually paralleling the concurrent dynamics of developmental processes in biology. Thus each simulation proceeds within the boundaries prescribed by the LSCs, including adherence to constraints on the behavior specified in the LSCs, such as forbidden scenarios. In the current implementation, in the absence of explicit specification of the order of events either within or between LSCs, play-out temporally interleaves events in a pre-set manner (according to the order of LSCs in the execution configuration and of the left-right order of instances within an LSC; see below for additional explanation). Mechanisms to control temporal prioritization within scenario based models is an area of current research. Simulations are visualized during a run either by tracking the progress of events in the charts or more globally via the GUI (see Supplementary Movies 1 and 2).

Overview of a simulation “run”

In this model, developmental time drives the progress of the simulation. Once the run begins, an LSC becomes "active" if conditions exist that are relevant to its prechart as monitored by the Play-Engine. For example, when sufficient time has passed, the worm hatches and enters the L1 (Fig. 3). This event triggers other LSCs to open, LSCs in which L1 entry is a relevant component of the LSC. An open or "active" LSC remains "active" until events specified by other LSCs (or by the progression of time) permit the completion of all events described in the LSC. Thus many LSCs can be active simultaneously. The Play-Engine monitors all active LSCs, trying to "complete" them – that is, monitoring whether the conditions and events specified in the precharts of open LSCs are satisfied and, if so, implementing the events in the corresponding main charts.

Example of a simulation "run"

Play-out can simulate the behavior of the system under any specific set of pre-set "experimental" conditions. This can be accomplished in either of two formats: Manual play-out that allows the user to manipulate the system for a single run or the Batch-Run play-out format that allows automated system runs for high-throughput testing. A sample manual run, in which we perturb the system by ablating P5.p and P6.p, will serve to illustrate the essence of how this model works.

A manual play-out simulation is initiated by the user, thereby allowing the user to pre-set the conditions of a simulated experiment prior to running the simulation. In this case, the ablation of P5.p and P6.p is set up by menu-driven manipulations of the cells as they are represented in the GUI (see User Guide in Supplementary Material). The GUI reflects the ablation of these cells by eliminating them from view (Fig. 7A, B). Genetic perturbations can be similarly performed by manipulating the internal object representations of genes. Once the desired starting conditions are set (P5.p and P6.p ablation), the user starts the simulation by clicking the START button on the GUI (see Supplementary User Guide). Time progresses via automatic clock ticks. All actions that are specified to occur within a specific developmental clock-tick time interval are completed before progressing to the next clock tick; that is, the CLOCK waits for the Play-Engine to complete its tasks before advancing the clock. This feature allows for developmentally “simultaneous” or temporally restricted events to occur properly even if the model simulation requires additional real time to complete the tasks.

Few events relevant to this model occur during the L1 stage. The major events are the birth of the Pn.p cells, visualized by their change from a dotted outline to white-filled ovals, and the birth of Z1.ppp and Z4.aaa, the AC/VU precursors. A module of uLSCs in the AC/VU Use Case specifies the outcome of the AC/VU decision (Fig. 6). Since the starting conditions of the particular simulation we are following (ablation of P5.p and P6.p) does not perturb any aspect of this decision, that process proceeds as in the wild type (via the sequence of uLSCs depicted in Fig. 6A-C).

During the second larval stage, the Pn.p cells migrate to fill the positions left available due to the absence of P5.p and P6.p. This movement is governed by a set of uLSCs within the Pn.p Movements Use Case. Both posterior and anterior Pn.p’s can move towards the center (the AC position), and moving cells leave spaces that can be filled by adjacent cells. The rules regulating movements have not been experimentally established in a systematic way, so the rules necessary for this model have been inferred from experimental ablation data (see below and

Supplementary Material). Consistent with the available data, the simulated movements are governed by non-deterministic rules (i.e., several outcomes can result from the same initial conditions). For example, P4.p will not always move to the P6.p position when both P5.p and P6.p are ablated, but instead will often acquire a final position somewhere between the normal positions of P4.p and P6.p, discretized into half-Pn.p-Pn.p units). In the example shown (Fig. 7B), P7.p moves to the P6.p position, P4.p only moves one half of a Pn.p-Pn.p unit distance (to the midpoint of P4.p-P5.p), and P3.p and P8.p do not move at all. Because of non-deterministic behavioral rules, the remaining VPCs will acquire different final positions in other runs that start with the same simulated experimental set-up.

In the L3, the fates of the VPCs are determined by a set of rules contained in a number of Use Cases including, Hyp7 Inhibitory Signal, Lateral Signaling, and VPC Fate Assumption. In the example shown (Fig. 7), P7.p acquired a 1° fate, since it was directly ventral to the AC. All other cells acquired a 3° fate, since they were neither adjacent to the 1° cell (P7.p) nor close enough to the AC to be induced to acquire a 2° fate. For this particular run, P7.p acquires a 1° fate upon entry into the L3 stage, since it "experiences" a high level (level = 100) of LIN-3 signal in its position directly ventral to the AC. The lateral signal mediated by LIN-12 does not influence the non-adjacent cells in this particular simulation. Cells that are not determined to acquire either 1° or 2° fates acquire a default 3° fate. The complete set of rules governing fate assignments are described in Supplementary Material.

Mechanisms used to implement VPC fate specification

The core papers represented in this model define two major pathways that influence VPC fate specification: an inductive signaling and a lateral signaling pathway. LIN-3 is the inductive signal, and the lateral signaling pathway is mediated by the Notch-like LIN-12 receptor (Sternberg, 2005). In the wild type situation, the AC is the major source of LIN-3, which triggers the 1^{o} and 2^{o} vulval fates. Our model includes a *Graded Signaling* mechanism by which high levels of LIN-3 stimulate 1^{o} fate specification while intermediate levels of LIN-3 stimulate 2^{o} fate specification. Our model also includes LIN-12-mediated lateral signaling between VPCs as part of a *Sequential Signaling* mechanism (Fig. 1B,D). Lateral signaling provides a mechanism by which VPCs specified as 1^{o} can promote 2^{o} fate specification in their neighbors. Thus, during the normal course of vulval induction, there is a sequential signaling process by which a VPC can acquire a 2^{o} fate specification: the initial LIN-3 signal induces a VPC to be specified as 1^{o}, and the 1^{o} cell subsequently specifies a 2^{o} fate in its neighbors by a lateral signaling mechanism. The combined action of these mechanisms provides two ways for a VPC to acquire a 2^{o} fate: (1) a medium level of LIN-3 inductive signal; and (2) a lateral signal from a 1^{o} neighbor. Further details on the implementation of inductive signaling, the influence of the lateral signal, and on fate acquisition in the *lin-15(0)* background are included in Supplementary Materials and Methods.

Integration of Events

A severe limitation of using abstract reasoning alone to evaluate an increasingly complex understanding of biology derives from the inability to keep track of a large number of concurrent events. One of the strengths of this modeling methodology is its ability to integrate concurrent events using a rigorous, formal methodology. Computational models can thereby assist the abstract reasoning that biologists use.

Integration of events first must take into account the dependency relationships between events. That is, which events depend on previous events or conditions. Some processes in uLSCs are linked by temporal causality, and the Play-Engine naturally integrates these processes. For example, the ability of P3.p to fuse with the hypodermal syncytium occurs *prior to* the subsequent events concerning VPC fate acquisition, but nonetheless, influences the outcome. Even in the wild type, in the L2 stage P3.p displays variability with respect to its decision to fuse with the hypodermal syncytium or remain unfused and be part of the vulval equivalence group (Sulston and Horvitz, 1977), and subsequent events will only influence the fate of P3.p if it remains unfused in the L2. Many mutations that affect vulval fate specification also influence the rate of P3.p fusion (Chen and Han, 2001), and this can be integrated into the model as well, and has been done for *lin-15(n309)* (see the uLSCs in the Pn.p Fusion Use Case). Nonetheless, because of the temporally distinct nature of these two processes, the Play-Engine naturally integrates these variable behaviors. This model includes a number of similar sub-processes that influence vulval fate specification (e.g., gonad location, AC/VU decision, Pn.p fusion, Pn.p movement). The ability of the Play-Engine to keep track of the ramifications of the events that occur in each of these sub-processes gives rise to some of the resulting biological insights obtained during the building and analysis of the model (see below).

Other behaviors are controlled by more interdependent events, for example genetic states. To constrain behavior and help integrate the processes affected by various genetic states, the current model uses Forbidden Elements and conditional statements (e.g., Fig. 6). A complete description of the pathway integration rules and methodology used in this model is available in Supplementary Material. Conditional statements can be used to restrict behaviors to the proper set of specific conditions; these are most appropriately used when the restrictions can be assigned to specific events that occur within an LSC. For example, the uLSC “VPCresponse100LIN3 (VPC Fate Assumption)” integrates the response of a VPC to a high level (level = 100) of LIN-3 with the genotype of the genes of the LET-23 signaling complex (Fig. 6D). Since the response to LIN-3 is dependent upon the activity of the receptor localization complex (LIN-2, LIN-7, LIN-10), the conditional statement allows a full 1º response only in cases where the complex is genetically intact.

To represent genetic perturbations that alter behavior, the governing uLSCs must not only trigger the new mutant behavior, but must also disable the relevant wild-type behavior. Representation of the AC/VU decision in this model serves as a useful example to explain how this can be accomplished. Under normal conditions, AC/VU behavior is represented by a wild-type AC/VU decision uLSC [“AC/VUdecision” (AC/VU)] (Fig. 6B). This uLSC represents the stochastic process that controls which of the pair of AC/VU cells (Z1.ppp or Z4.aaa) will acquire an AC fate, as well as the effect of cell ablations on this fate decision. Various *lin-12* mutations will block this wild-type behavior and trigger new behaviors. The mutant behaviors are represented by the creation of new uLSCs (e.g., see Fig. 6C), triggered by the specific relevant genetic perturbation (for example, lin12(d)_Phenotype). Disabling wild-type behaviors under specific conditions is done effectively via the use of “Forbidden Elements.” Forbidden Elements can obviate an entire uLSC and all of its compelled behaviors or prevent an event from occurring (see Harel and Marelly, 2003); for examples, see the Documentation Manual in Supplementary Material). Thus, the events denoted as Forbidden Elements will block the effects of the uLSC if they occur within the denoted timeframe. Fig. 6A shows that the wild-type behavior triggered in the "WT_AC/VU" (AC/VU) uLSC is blocked by any non-wild-type *lin-12* genotype (linked elsewhere in the model to lin12()_Phenotype).

The use of conditional statements and Forbidden Elements is reasonable for this model in which there are a small number of genes, gene activity levels (alleles which are essentially "ON" or "OFF"), and interacting pathways. As the system becomes increasingly complex, however, these elements will become increasingly cumbersome to list and keep track of. We are currently developing alternative methods to integrate genetic interactions for the next phase of this modeling effort.

Testing

Manual play-out allows a user to test whether individual simulations match the expected behavior of the system. While this is important for model development, the model must also be tested more rigorously and systematically.

Recapitulation

The first part of rigorous testing of a mechanistic model should demonstrate that the model can recapitulate the experimental observations from which the relevant mechanisms were inferred. This type of systematic testing parallels the reason-based testing of pictorial models.

For complex computational models, this testing is critical to bolster confidence in the predictive capacity of the model.

Testing is accomplished by matching an experimental condition and the observed result to a potential simulation that starts with the same conditions and ends with the same outcome. A basic feature of the Play-Engine is the ability to trace the events in all of the participating LSCs of a model. Thus, the standard functions of the Play-Engine can be used for testing purposes by representing experimental conditions and their outcomes as LSCs. Taking this approach, experiments reported in the literature were represented as eLSCs. In contrast to the uLSCs, which represent the mechanistic rules of our model and describe behaviors that must hold true for all runs of the model, eLSCs do not compel behavior, but rather describe a scenario that should be matched by at least one run of the system. For example, although P6.p will acquire a 1º fate under wild-type conditions, this will not occur under a variety of mutant (*vul* mutants) and ablation conditions (e.g., ablation of the AC). Thus, this behavior (the acquisition of a 1º fate by P6.p) cannot be used as a general rule, but can be represented as an eLSC to describe the various experimental scenarios in which this outcome does occur. To enable systematic testing, each of the experimental results of the core papers was translated into an eLSC and stored in a set of Use Cases matched to each paper (e.g., the Use Case S88 EXP contains the set of eLSCs describing the experimental results contained in S88; Fig. 4A).

We have shown that our model can reproduce essentially all of the results observed for each experiment that was conducted in the core papers. Even for a model that represents a small number of papers, this is a large task that involves running simulations describing each independent experimental result that was reported (over 250 total results). Furthermore, the non-determinism of experimental biology (and its representation in the model) requires multiple iterations for each independent experimental set-up (over 450 total runs; each of these runs were carried out independently on three different variants of the current model, the *Core* model described here, and two variants – see below, Hypothesis testing – for a total of over 1350 runs). To help this testing procedure, a number of features of the Play-Engine were developed to facilitate automated simulations and their analyses. Jump starts provide the automated setting of the initial simulation conditions. Batch-Run play-out automates the running of multiple simulations. Various aspects of the results of these simulations were automatically recorded in a number of different formats. These automated features (described in detail in the Supplementary Materials and Methods) allowed systematic test runs conducted according to a comprehensive test plan, to test the match between the outcome of runs and the results of each of the reported experiments in the core papers. One example is depicted in Figure 7C. Initially, these test runs were performed to identify bugs in the model; later they were used to systematically test the model as indicated below. (Detailed test plans including each configuration, the number of runs, run recordings, the outcome excel files, and the log files are available upon request).

Testing experimental results against a mechanistic hypothesis

After proposing a mechanism that underlies a biological phenomenon of interest, the biologist then reconsiders the relevant experimental results to ensure that they are all compatible with the hypothesized mechanism and that none contradict it. The computational model we have built can perform this task rigorously and comprehensively. As one of its normal functions, the Play-Engine monitors the progress and state-of-completion of all active LSCs during a run. We used this function to test the *Core* model on all of the experimental results reported in the three papers, SH86, S88 and SH89.

We ran individual simulations that mimicked essentially all of the experiments reported in the papers (table by table, line by line; over 450 runs representing approximately 260 results in total) and compared the simulation results with the actual results reported in the papers. To facilitate these tests, we formalized each of the experiments in a separate set of LSCs of a different type: existential LSCs (eLSCs). Each eLSC contains only a main chart (no prechart) that describes both the specific experimental set-up and the observed experimental result (Fig. 2C). eLSCs do not propel the model or affect its behavior, but allow monitoring of experiments formalized in this manner. Due to the non-determinism of the system (that is, one experimental set-up can yield non-identical results), we ran multiple iterations for each experimental set-up to cover the space of potential outcomes using semi-automated simulation and analysis features of the Play-Engine (see supplementary material section on testing within methods).

In general, model testing can detect three broad categories of inconsistencies between a model and the data: bugs (the model does not do what we wanted it to do), "acceptable failures" (model does not completely satisfy observations due to some approximations we made, or any other limitation we are willing to accept as a reasonable level of abstraction), and "interesting failures", which may lead to insights. Our iterative model development and testing aims to filter out the first category, assuring that we are aware of the second, and drawing our attention to the third. The latter are of particular interest since they point out areas in the mechanistic hypotheses that are not consistent with the data.

Tests showed that most but not all experimental results were reproduced by the *Core* model (examples of runs are given in Supplementary Movies). Results that were not reproduced included experiments in which Pn.p daughter cells were ablated (our model does not yet extend to the divisions of the Pn.p cells and so these data were not tested), and the twenty-three independent results listed in Supplementary Table 1A. These results are largely from experiments in which partial induction patterns were observed in some of the *vul* mutants (*lin-2, -3, -7, -10*) (Supplementary Table 1). The inability of the model to reproduce these results derives from several sources. First, the current model does not include the rules to generate hybrid or intermediate fates. In fact, this is still not understood close to twenty years later. Second, in the model, the inductive pathway mutants block the pathway completely, whereas in reality they block partially, a caveat that was appreciated by the field at the time. Third, experimental results are sometimes summarized in ways that are ambiguous or that do not strictly agree with observations (though they are consistent with the preponderance of the data they summarize). And, fourth, very rare results, such as a 2º fate in a *lin-12(0)* mutant, remain unexplained (Supplementary Table 1). The ability to detect inconsistencies between a model and the data it is based on is crucial to the model’s usefulness, and is a key feature of our modeling methodology.

Testing experimental results against competing mechanistic hypotheses

Biologists are often faced with more than one mechanistic hypothesis that appears compatible with the available experimental data. The next task is to identify what specific results are compatible with one or more hypotheses but not others. To replicate this aspect of the reasoning process, we used the same large set of experimental results to test a famous example of two competing mechanistic hypotheses that concern VPC fate specification: the *"Graded"* versus the *"Sequential"* signaling hypotheses (Sternberg and Horvitz, 1986). These hypotheses account for 2º fate specification by different mechanisms. The *Graded* signaling hypothesis postulates that a VPC that directly receives an intermediate level of LIN-3 signaling will adopt a 2º fate, while the *Sequential* hypothesis postulates that 2º fates are induced by lateral signaling from an adjacent VPC that receives a high level of LIN-3 and adopts a 1º fate (Fig. 1B). These hypotheses have been a subject of controversy ever since the publication of the papers covered in our model (see (Katz et al., 1995; Kenyon, 1995; Simske and Kim, 1995; Sundaram, 2004). Both mechanisms are operational in our *Core* model, each being represented by a small number of "universal" LSCs (uLSCs; see Fig. 2).

We tested whether our modeling approach could "find" the experiments that were inconsistent with each hypothesis by altering the settings in a feature of the Play-Engine called the *Execution Configuration* and then repeating the testing (over 900 additional runs). The *Execution Configuration* allows one to define the set of uLSCs that will be active in defining the behavior of the model. Leaving all other uLSCs from the *Core* model active, the *Sequential* model was created by inactivating the two uLSCs that enable the mechanism by which intermediate levels of LIN-3 can specify a 2º fate (Fig. 1C,D). The *Graded* model was similarly created from the *Core* model by removing two different uLSCs that enable the LIN-12-mediated lateral influence on VPC fate specification (see Supplementary Materials and methods). Systematic testing of each model independently revealed a set of experimental results that were not reproduced according to the testing procedure that had been successful for the *Core* model.

The un-reproduced results under the *Sequential* model correspond to cases in which VPCs adopt 2º fates in the absence of adjacent 1º VPCs (7 independent results; Supplementary Table 1B; see also Supplementary Movies 3 and 4). Testing under the *Graded* model showed that non-wild-type alternating 1º/2º fate patterns are difficult to reproduce (21 independent results; Supplementary Table 1C). One example is the result described in Table 2 C, line 1 of SH89. The pattern of fates for this *lin-15(n309)* mutant is 1º/2º/2º/1º/2º/1º from P3.p to P8.p, respectively. The graded signaling mechanism alone cannot generate a 2º fate for P4.p, since it is too far from the anchor cell, and 2º cell fates in *lin-15(n309)* animals derive predominantly from lateral signaling (Sternberg, 1988). Data collected since the publication of these papers suggest that both mechanisms contribute to the pattern, as modeled in the *Core*. Nonetheless, these systematic tests demonstrate the potential of this methodology for testing competing hypotheses against an established set of experimental results, and for supplying lists of results that are/are not reproduced under similar testing protocols running under the alternative hypotheses.

The study by Fisher et al., 2007 uses a verification-based method (model-checking) rather than a testing method to compare their model to experimental results. A main distinguishing feature and advantage of verification-based methods over testing is that verification can, in principle, prove that a certain behavior is impossible in a given model, whereas the fact that a certain behavior was never observed during testing does not mean it cannot occur. Although our modeling tool, the Play-Engine, supports model-checking (Harel et al., 2002), here we focused on a testing-based approach (Kugler et al., 2007) for comparing our model with the experimental results due to the large size and complexity of our model. To capitalize on the advantages of model-checking approaches for complex biological systems, future work is necessary to improve scaling of the verification-based methods to handle models such as our *Core* model and larger models.

Biological insights from the modeling

In addition to providing a basic prototype for a more detailed model of VPC fate specification, the construction and simulations of the *Core* model yielded several novel insights into our understanding of this extensively-studied biological process. Model construction primarily highlighted data inconsistencies and gaps in our understanding, while model simulations contributed insights that predominantly derived from our actual visualization of the dynamic behavior of the executable model. We found this visualization very instructive compared to the normal process of mentally following the implied dynamic behaviors of static pictorial diagram models representing the same events. We discuss several examples of these insights to emphasize the breadth of the biology they cover.

VPC movements

Our modeling effort has highlighted a key mechanistic gap in our understanding of VPC movements after cell ablations. This process can serve as a general model for cellular mechanisms that underlie re-occupation of newly opened space in response to normal processes such as programmed cell death and abnormal processes such as injury. The fates of the cells in the VPC equivalence group are dependent on their position, underscoring the importance of VPC movement. In fact, even the names of the cell fates (1º, 2º, 3º) reflect a hierarchy of fate replacement that requires cell movement to enable replacement regulation among the VPCs (Sulston and White, 1980).

In order to build an executable model that could recapitulate experimental results, we had to include the potential for these movements and a set of hypothesized rules that govern them. Though it has been documented that VPCs can move (usually in the direction of the AC) and that movement alters fate acquisition (Sternberg and Horvitz, 1986), the literature does not yet contain a systematic study of these movements from which a set of rules could be derived. Examples of the outstanding questions pertaining to this key process include establishing the rules to represent the non-deterministic positioning of isolated VPCs, the rules that govern movement in the absence and presence (or duplication) of the AC, and the differences in these rules for VPCs that have neighbors versus those that do not. We intuited some of these rules from the existing data, but clearly a systematic experimental analysis of this process would be of interest.

Analysis of simulations revealed further gaps in our understanding of the relative timing of VPC movements. Without time constraints or links to other processes, the Play-Engine (as expected in the absence of specified dependency relationships) treated VPC movement and other events independently. Thus, for example, in an early version of the model, one simulation showed P3.p moving prior to its decision to undergo an early fusion with the hypodermal syncytium. Presumably, an early decision to undergo fusion – a capability unique to P3.p in the second larval stage – would not occur after movement toward the AC since movement toward the AC would lead to a VPC fate for this cell, a fate that would preclude early fusion. This intuitively aberrant simulation (P3.p movement before fusion decision) occurred prior to our implementation of additional constraints, and drew our attention to the gap in our knowledge of this aspect of VPC behavior. Therefore, it will be important to establish experimentally not only the rules for VPC movement, but also the relative timing of these movements.

Fate asymmetries between P5.p and P7.p

Several insights derive from our renewed focus on published data, the formalization of these data necessary to create the model, and the relatively systematic representation of data. At different times during the elucidation of biological processes, different levels of abstraction are necessary and appropriate to summarize the data. The formal, systematic, mechanism-based description of biology brings a new perspective to what can be learned from an old set of data, demonstrating an additional usefulness of this approach. Moreover, data that provide exceptions to a generalized mechanistic understanding often provide motivation for further experimental inquiry. These data, however, can be "lost" within a field of study over time. Thus the implications of the detailed results can be lost in the higher-level data summaries that were necessary and useful for the field at the time the data were collected. Some of the summarizing abstractions in the field of VPC fate specification have not been updated since the original publication of the data we modeled. Thus our modeling effort provided the opportunity to re-evaluate these summaries. We encountered many examples of insights obtained in this manner.

One illustrative example of this type concerns details of fate asymmetries between P5.p and P7.p (Table 1, SH89). Specifically, Table 4 in SH89 provides a summary of the experimental results. In this table the authors interpret the outcome of the *lin-12(0)* experiments as if P(5-7).p all behave in a similar way, i.e., execute a 1° fate. The actual data reported in Table 1 of the paper, however, reveal a more complex situation. While P5.p and P6.p indeed behave similarly (the key result captured in the summary), P7.p becomes 1° in only 7 out of 11 animals (in 3 other animals P7.p executes what seems to be a 1°/3° hybrid fate, and in one animal P7.p executes a non-vulval 3° fate). Furthermore, P4.p and P.8 do not always acquire 3° fates, as summarized in Table 4. P4.p seems to be at least partially induced in 8 out of 11 animals (in one case it even executes what seems to be a 1° fate), and P8.p executes what seems to be a 1°/3° hybrid fate in 2 out of 11 animals. P3.p is actually the only VPC that remains completely uninduced.

Since we aimed at creating a model that could reproduce the actual data and not only a summary or idealized representation thereof, we had to suggest a mechanism to account for these discrepancies. Two possibilities are: (1) either the LIN-3 gradient is asymmetric with respect to the VPCs; or (2) the response of VPCs to a given LIN-3 concentration is not identical. The relative responsiveness of the VPCs to inductive signaling effect was investigated more thoroughly in studies conducted after the three core papers were published, and can be understood in terms of a pre-pattern set up by differential combinatorial expression of the *lin-39* and *mab-5* transcription factors in the VPCs (Clandinin et al., 1997). The incorporation of these studies and results is beyond the scope of this version of our model. Nevertheless, this example demonstrates that this modeling methodology permits us to store for future testing original data that are not necessarily easily integrated into a current mechanistic framework or abstraction.

LIN-12 activity

Despite the absence of a representation of the detailed molecular pathways constituting the interplay between the inductive and lateral signals mediated by the LET-23 and LIN-12 receptors, respectively, our modeling highlighted several interesting aspects of LIN-12 activity that affect VPC fate specification. One of these underscores our lack of understanding of the specific mechanism by which multiple VPCs adopt 1° fates in a lin-12(0) mutant.
In *lin-12(0)* mutants, there are multiple ACs, which could lead to a wider LIN-3 inductive gradient. A combination of the release of a larger amount of the LIN-3 inductive signal and the greater physical space spanned by multiple ACs might contribute to such a wider LIN-3 gradient. An alternative mechanism is based on the effect of the *lin-12(0)* mutation on the response of the VPCs rather than on a wider distribution of inductive signal. Since LIN-12 is required for the 2° fate (Greenwald et al., 1983), without LIN-12, there is no mechanism to specify 2° fate in response to signaling from a 1° neighbor. Without a pathway promoting a 2° fate, the threshold for signaling to become 1° might be significantly reduced.

Our modeling approach can represent any of these mechanisms, and test runs can simulate the resulting outcome. In our current model, a spatially-widened LIN-3 mechanism is employed. In a simulated experiment, we ablated the extra ACs generated as a consequence of reduced *lin-12* activity, leaving only one AC. Under these conditions, only P6.p adopts a 1° fate, while P5.p and P7.p adopt 3° fates. If the alternate mechanism (blocking induction of the 2° fate) is responsible for the pattern observed in a *lin-12(0)* mutant, a similar ablation would have no effect on the fates of P5.p and P7.p, and both should adopt a 1° fate. The model-building thereby highlighted our lack of knowledge concerning the relative contributions of these two potential mechanisms. Using the model, however, we can easily identify and visualize the result of an experiment that depends on the mechanism we implemented.

Experimental follow-up on temporal prioritization of VPC 1º fate acquisition

In addition to the normal competition between the LIN-12 and LIN-3 signaling pathways as VPCs choose between the 1º and 2º fates, the activity of the SynMuv genes normally prevents ectopic expression of LIN-3 from the hypodermis (Cui et al., 2006). Two SynMuv proteins are encoded by the *lin-15* locus. In a *lin-15* mutant that eliminates both activities, all VPCs acquire one of the two vulval fates, 1º or 2º. Sternberg (1988) established that isolated VPCs in a *lin-15* background predominantly acquire a 1º fate, and that lateral signaling between the VPCs accounts for the observed alternation between 1º and 2º fates observed in the *lin-15* mutant.

We represented this level of understanding in the absence of the recently-discovered mechanistic explanation for the effects of *lin-15(0)* (Cui et al., 2006)). In the *Core* model, two mechanisms were used to confer a 1º fate: (1) induction by the normal LIN-3/EGF signal deriving from the gonadal anchor cell (AC; see Fig. 1A); and (2) a default state particular to *lin-15(0)* whereby the ability to acquire a 1º fate occurs as a result of a "race" between the VPCs for the 1º fate (Fisher et al., 2005; Sternberg, 2005). The experimental data in S88, however, also show that P6.p always acquires a 1º fate in the presence of an AC. To enable the *Core* model to recapitulate this result, temporal priority was given to the LIN-3 mechanism (see Supplementary Movie 5).

The requirement for temporal prioritization for P6.p fate specification in the *lin-15* mutant in this model, as well as a similar requirement in a previous model (Fisher et al., 2005), led us to investigate this possibility further. We experimentally tested the hypothesis using the *ayIs4[$P_{egl\text{-}17}$::GFP]* transgene, a GFP reporter of 1° fate specification (Burdine et al., 1998). The data (Fig. 8) indicate that P6.p is biased to be specified as 1° prior to the other VPCs that ultimately acquire a 1° fate, confirming the temporal priority hypothesis that was necessary to implement for the model to reproduce the reported results. Similar experimental results were obtained by Fisher et al. (2007) as a validation for comparable findings in their modeling effort.

Surprisingly, we also found that 1° fate specification for all VPCs, as monitored by this marker, occurs significantly earlier in *lin-15(n309)* than in wild type (Fig. 8). Thus, an experiment prompted by our modeling efforts not only validated a mechanism that was required in the model to replicate experimental data but also generated new information concerning the dynamics of VPC fate specification. The finding that 1° fates are specified very early in *lin-15(n309)* compared to the wild type suggests that inappropriate expression of LIN-3 from the hypodermis (Cui et al., 2006) occurs in the L2, very early in development and/or that in the absence of *lin-15*, there is an overall elevation of *egl-17* expression in all VPCs. These results are consistent with recent results suggesting that VPCs can respond to ectopic or temporally altered expression of *lin-3* either from the hypodermis, other VPCs, or the germ line (Cui et al., 2006); (Thompson et al., 2006; Walser et al., 2006).

Conclusions

Our model is a prototype that was built and tested to determine the extent to which the methodology could be applied to a biological problem and serve as a means for biologist to test hypotheses in a manner consistent with the informal “model-building” common to developmental studies. Despite the relatively small scope of this proof-of-principle modeling study relative to the surfeit of published results on vulval development in *C. elegans*, we achieved our aim of testing a novel *process* and *methodology* by which results of classical developmental biology could be gainfully modeled. We asked if essentially all the findings and inferences from a particular set of papers could be incorporated into a single modeling approach. Second, we chose to limit our analysis to a set of papers that are historically relatively distant. We tried as much as possible to remain true to the ideas of the time so as to recapitulate the thought processes that may go into modeling emerging fields of study. As a result, though the current model focuses on a stage of the progression of the field that precedes the more current molecular analyses, this later progress within the field can now be modeled within the context of the early data, rather than being modeled in isolation.

The need for modeling methods that can handle several approaches simultaneously to cover the varied needs of the field of developmental genetics has been well articulated (Reeves et al., 2006). Several aspects of this modeling paradigm offer promise for enhancing the understanding of biological systems. Our model represents not only biological mechanisms in a rule-based conceptual framework, but also actual experimental results, allowing one to be tested against the other. The formal structure of this model allows for rigorous testing that is not feasible for static pictorial models. Despite the underlying rigorous formal logic, the graphical representations make it accessible to biologists. The modular nature of the description of behavior well suits the field of biology; the phenomenal successes of reductionist approaches in biology are most easily captured in LSCs that represent snippets of behavior (“scenarios”). Furthermore, these “scenario”-based descriptions of behavior are simpler to modify than in non-scenario-based approaches. The addition of new data or even paradigmatic shifts in our understanding should require relatively modest modifications. Finally, 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; facile methods to build system-specific GUIs is an area of ongoing research efforts. Furthermore, the solutions we have found to represent the processes and behaviors of vulval fate specification should be applicable to similar aspects of other systems, since they represent some of the fundamental underpinnings of many biological systems.

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. Improvements in scalability for the language and tool are ongoing research areas. Nonetheless, the inherent structure of this model aids its extendibility. The model contains a number of elements already designed for future development including the representation of core genes within the inductive signal transduction pathway, VPC lineages in the GUI, and a temperature control for representing behaviors at different temperatures. The ability to store non-participating LSCs within the model aids in the construction of new modules within an existing model. Thus, during model extension, an incompletely functional set of uLSCs can be kept from participating in model simulations until ready to be tested. For example, a number of uLSCs in the Use Case “Post PoP” are present in our model as part of extension modules that are not yet fully functional; these are kept “out” of the set of participating uLSCs in the execution configuration. An obvious extension of this model will be to incorporate the contents of more recent molecular work on vulval fate specification. This will require representing additional genetic perturbations that are common to the field, such as the use of transgenic constructs and RNAi. Related sub-systems, such as a more complete representation of the somatic gonad, as well as the more quantitative models within this modeling paradigm to create hybrid models can be modeled as semi-independent projects and linked together using a simulation engine coordinator, for which a prototype already exists (InterPlay; Barak et al., 2004; Barak et al., 2006; Sadot et al., (in press)). This is an exciting area for future work. The hope is that our current model will form a core that will be deepened and expanded, and provide adaptable tools to represent other biological systems.

Our results suggest that the hypothesis-experiment-hypothesis cycle often aided by static diagrammatic models can be supplemented by a hypothesis-modeling-testing-experiment-hypothesis cycle aided by executable dynamic models. The most important advantages this approach offers over the current reason-based approaches are twofold: the visual dynamic aspect of the modeling, and the increased volume of information that can be taken into consideration. We propose that similar benefits could be obtained with the application of this modeling paradigm to other areas of biology that incorporate many levels of related information and thereby face similar challenges. Our approach is poised to bridge the gap between classical “reductionist” and “systems” approaches to biology and could facilitate the building of comprehensive models of very complex biological systems.

Acknowledgments

We gratefully acknowledge the contributions of D. Barak, and M. Yano to various aspects of this work. In addition, we thank and R. Posner, L. Cooley, and K. Birnbaum for critical reading of early versions of the manuscript. This work was supported by collaborative NIH grant GM0066969, the Yale-Weizmann Exchange Program, the John von Neumann Minerva Center at the Weizmann Institute, and a grant from the Kahn Fund for Systems Biology at the Weizmann Institute of Science.

References

Ambros, V., 1999. Cell cycle-dependent sequencing of cell fate decisions in Caenorhabditis elegans vulva precursor cells. Development. 126, 1947-56.

Aroian, R. V., et al., 1990. The let-23 gene necessary for Caenorhabditis elegans vulval induction encodes a tyrosine kinase of the EGF receptor subfamily. Nature. 348, 693-9.

Barak, D., et al., 2004. InterPlay: Horizontal scale-up and transition to design in scenario-based programming. In: J. Desel, et al., Eds.), Lectures on concurrency and petri nets, Vol. 3098. Springer, 2004, pp. 849.

Barak, D., et al., 2006. InterPlay: Horizontal Scale-Up and Transition to Design in Scenario-Based Programming. IEEE Trans. Soft. Eng. 32, 467-485.

Burdine, R. D., et al., 1998. EGL-17(FGF) expression coordinates the attraction of the migrating sex myoblasts with vulval induction in C. elegans. Development. 125, 1083-93.

Chen, Z., Han, M., 2001. C. elegans Rb, NuRD, and Ras regulate lin-39-mediated cell fusion during vulval fate specification. Curr Biol. 11, 1874-9.

Clandinin, T. R., et al., 1997. Caenorhabditis elegans HOM-C genes regulate the response of vulval precursor cells to inductive signal. Dev Biol. 182, 150-61.

Cui, M., et al., 2006. SynMuv genes redundantly inhibit lin-3/EGF expression to prevent inappropriate vulval induction in C. elegans. Dev Cell. 10, 667-72.

Damm, W., Harel, D., 2001. LSCs: Breathing life into message sequence charts. Formal methods in system design. 19, 45-80.

de Jong, H., 2002. Modeling and Simulation of Genetic Regulatory Systems: A Literature Review. Journal of Computational Biology Journal of Computational Biology. 9**,** 67-103.

Efroni, S., et al., 2003. Toward rigorous comprehension of biological complexity: modeling, execution, and visualization of thymic T-cell maturation. Genome Res. 13**,** 2485-97.

Fisher, J., et al., 2007. Predictive modeling of signaling crosstalk during C. elegans vulval development. PLoS Comput Biol. 3**,** e92.

Fisher, J., et al., 2005. Computational insights into Caenorhabditis elegans vulval development. Proc Natl Acad Sci U S A. 102**,** 1951-6.

Giurumescu, C. A., et al., 2006. Intercellular coupling amplifies fate segregation during Caenorhabditis elegans vulval development. Proc Natl Acad Sci U S A. 103**,** 1331-6.

Greenwald, I., et al., 1983. The lin-12 locus specifies cell fates in Caenorhabditis elegans. Cell. 34**,** 435-44.

Harel, D., 1987. Statecharts: a visual formalism for complex systems. Science of computer programming. 8**,** 231-274.

Harel, D., et al., Smart Play-Out of Behavioral Requirements., Conference on Formal Methods in Computer-Aided Design (FMCAD'02), Vol. 2517 of Lect. Notes in Comp. Sci, Portland, Oregon, 2002, pp. 378-398.

Harel, D., Marelly, R., 2003. Come, Let's Play. Scenario-based programming using LSCs and the Play-Engine. Springer-Verlag.

Herman, R. K., Hedgecock, E. M., 1990. Limitation of the size of the vulval primordium of Caenorhabditis elegans by lin-15 expression in surrounding hypodermis. Nature. 348**,** 169-71.

Ideker, T., Lauffenburger, D., 2003. Building with a scaffold: emerging strategies for high- to low-level cellular modeling. Trends Biotechnol. 21, 255-62.

Kaech, S. M., et al., 1998. The LIN-2/LIN-7/LIN-10 complex mediates basolateral membrane localization of the C. elegans EGF receptor LET-23 in vulval epithelial cells. Cell. 94, 761-71.

Kam, N., et al., 2004. Formal Modeling of C. elegans Development: A Scenario Based Approach. In: G. Ciobanu, (Ed.), Modeling in Molecular Biology, Natural Computing Series. Springer, 2004.

Katz, W. S., et al., 1995. Different levels of the C. elegans growth factor LIN-3 promote distinct vulval precursor fates. Cell. 82, 297-307.

Kenyon, C., 1995. A perfect vulva every time: gradients and signaling cascades in C. elegans. Cell. 82, 171-4.

Kimble, J., 1981. Alterations in cell lineage following laser ablation of cells in the somatic gonad of Caenorhabditis elegans. Dev Biol. 87, 286-300.

Kugler, H., et al., Testing Scenario-Based Models. International Conference on Fundamental Approaches to Software Engineering (FASE). Springer, Braga, Portugal March 24-April 1, 2007, 2007.

Reeves, G. T., et al., 2006. Quantitative models of developmental pattern formation. Dev Cell. 11, 289-300.

Sadot, A., et al., (in press). Towards Verified Biological Models. IEEE IEEE/ACM Transactions on Computational Biology and Bioinformatics.

Simske, J. S., Kim, S. K., 1995. Sequential signalling during Caenorhabditis elegans vulval induction. Nature. 375, 142-6.

Sternberg, P., Vulval development (June, 25 2005). In: The C. elegans Research Community, (Ed.), WormBook, doi/10.1895/wormbook.1.6.1, http://www.wormbook.org, 2005.

Sternberg, P. W., 1988. Lateral inhibition during vulval induction in Caenorhabditis elegans. Nature. 335**,** 551-4.

Sternberg, P. W., Horvitz, H. R., 1986. Pattern formation during vulval development in C. elegans. Cell. 44**,** 761-72.

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**,** 679-93.

Sulston, J. E., Horvitz, H. R., 1977. Post-embryonic cell lineages of the nematode, Caenorhabditis elegans. Dev Biol. 56**,** 110-56.

Sulston, J. E., White, J. G., 1980. Regulation and cell autonomy during postembryonic development of Caenorhabditis elegans. Dev Biol. 78**,** 577-97.

Sun, N., Zhao, H., 2004. Genomic approaches in dissecting complex biological pathways. Pharmacogenomics. 5**,** 163-79.

Sundaram, M. V., 2004. Vulval development: the battle between Ras and Notch. Curr Biol. 14**,** R311-3.

Thomas, J. H., et al., 1990. Cell interactions coordinate the development of the C. elegans egg-laying system. Cell. 62**,** 1041-52.

Thompson, B. E., et al., 2006. Germ-line induction of the Caenorhabditis elegans vulva. Proc Natl Acad Sci U S A. 103**,** 620-5.

Walser, C. B., et al., 2006. Distinct roles of the Pumilio and FBF translational repressors during C. elegans vulval development. Development. 133**,** 3461-71.

Figure Legends

Figure 1. Diagram models, the GUI and hypothesis testing using different *Execution Configurations*. (A,B) Pictorial models of the mechanisms affecting vulval fate specification relevant to the core set of papers. The *C. elegans* hermaphrodite vulva forms from a set of six ventral epidermal/hypodermal blast cells (Pn.p cells) (Sternberg, 2005). Under normal conditions, only three of these cells (P(5-7).p) form vulval tissue, while the remaining three cells (P(3,4,8).p) acquire a non-vulval, tertiary (3º) fate. A cell in the overlying gonad, the anchor cell (AC), lies immediately dorsal to P6.p and induces it to form the central vulval cell fate, known as the primary (1º) fate. The adjacent VPCs, P5.p and P7.p, acquire a secondary (2º) vulval fate. As in the GUI, primary VPC fates are depicted in blue, secondary fates in red. In the GUI, tertiary fates are depicted in yellow. LIN-12/Notch-mediated lateral signaling is depicted by red arrows, LIN-3/EGF inductive signaling by blue arrows (strong signaling, thick blue arrow; medium signaling, thin blue arrows). The old representation of the LIN-15/SynMuv regulation of VPC fate specification is shown in purple, as is the line representing the surrounding hypodermal syncytium. (A) All mechanisms are operational in the *Core* model. (B) Medium levels of inductive signaling that induce secondary VPC fates are omitted in the *Sequential* signaling model. (C,D) The relevant portion of the *Execution Configuration* showing the set of participating uLSCs for the *Core* model (C) and *Sequential* signaling model (D). The *Sequential* signaling model was generated from the *Core* model by moving the two LSCs (boxed in red) “out” from the set of participating uLSCs. Moving uLSCs “in” and “out” of these sets is accomplished by clicking on the arrows between the lists of sets. These two uLSCs describe the behavioral response of VPCs to a medium level of inductive signal (the thin blue arrows in (A)).

When these uLSCs are "out" of the set of participating uLSCs, VPCs do not acquire a secondary fate in response to medium levels of inductive signal during simulations. Only a small portion of the sets of uLSCs is visible. (E,F) The final fates of the VPCs depicted in the GUI for a run which matches the result of an experiment in which P6.p was ablated in a wild-type background and neither of the adjacent VPCs moved to replace it (Table 1, lines 5,6,8 of SH86). Simulations using the *Execution Configuration* for the *Core* model (E) or the *Sequential* signaling model (F). Only the *Core* model reproduces the result obtained in this experiment.

Figure 2. Live sequence charts. (A) The general structure of a universal *Live Sequence Chart* (uLSC). 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 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). (B) An example of a uLSC ("AdoptTertiaryFate") that illustrates some of the complexity that can be built into an LSC specification. This uLSC assigns a 3º fate to all VPCs that remain unspecified ("UnDifferentiated") at the time of final fate determination. A global event in the prechart, PnpFinalAssumeFate, is the necessary condition that triggers the events portrayed in the main

chart. This event indicates the final timepoint at which the fate of all non-determined VPCs becomes determined and fixed. The simple method called to assign a 3º fate to unspecified VPCs (AdoptFate(Tertiary)) is constrained by a bounding pair of time conditions (to occur one hour after the PnpFinalAssumeFate event) and a state condition that specifies that only "UnDifferentiated" VPCs will be affected. Many other constraining constructs are available for specifying precise behaviors using LSCs (Harel and Marelly, 2003), and are used throughout the model. (C) An example of an existential LSC (eLSC). This eLSC portrays an experiment reported in Table 4, line 15 of SH86, in which P8.p is the only unablated VPC, moves to the position normally occupied by P7.p (ExactLoc=P7.p), and acquires a 2º fate. (This is a crucial piece of evidence for the *Graded* signaling hypothesis.) 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 Fig. 1A). 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.

Figure 3. The events in the main chart of one LSC triggers an event in the prechart of another LSC. The Developmental Time uLSC that lies at the heart of the VPC model project. The arrow indicates the direct link between the Developmental Time uLSC and the PnDivide uLSC. Note the time constraints within the red hexagon conditions. The numbers refer to hours post-hatching, except for after L3 entry (which include 60 minutes for the first hour of L3). The

"PnDivide(Core Behaviors)" LSC that responds to passage to the L1 that occurs within the "Developmental Time 20C(Core Behaviors)" LSC.

Figure 4. Use Cases. (A) Use Cases in the model into which LSCs are organized. (B) Schematic representation of interactions among LSCs within Use Cases. An arrow indicates that the source Use Case affects the target use case, either by direct LSC activation or by affecting relevant variables. Two-headed arrows indicate mutual effects. LSCs within Use Cases typically affect each other; these interactions are not depicted here.

Figure 5. An example of Play-in. Suppose the user wants to specify that "if P6.p adopts the primary fate, it should send a lateral signal to P7.p, and P7.p should become secondary". To "play-in" this scenario and simultaneously create the LSC that specifies it, the user selects "new LSC" from the approprate Use Case; an empty LSC appears on the screen. The user then selects "Play-In" from the menu. A purple dotted line is in the prechart. (A) To implement the "if" condition in the prechart, the user right-clicks on P6.p in the Graphical User Interface (GUI). (B) A menu appears including P6.p's "Fate" property. Selection of this property opens a list of possible fates (C), including "Primary". After the user has selected "Primary", both the LSC and GUI reflect the selection (D). To move to the main chart, the user goes to the LSC and clicks and drags the dotted purple line from the prechart into the main chart (E). The user then right-clicks on P6.p on the GUI, revealing additional options (F). The user selects "call other object" (G). The user right clicks on P7.p and selects LS, for lateral signal (H). The user selects "OK". The LSC reflects the changes (I). The user right-clicks on P7.p, selecting "Fate" (J). The user selects "Secondary (K). The GUI and LSC reflect the changes (L). Note: This Sample LSC has been

created for illustration purposes only and does not drive behavior of the current model; in the context of the actual *Core* model, this statement is broken down into several LSCs.

Figure 6. Integrating the effects of perturbations. (A) Normal ("WT"), time-constrained AC/VU behavior is triggered by the standard events of normal development: hatching and L2 entry (in prechart). Under conditions in which a non-WT *lin-12* phenotype is operative, forbidden elements at the bottom of this LSC prohibit the execution of the events in this uLSC. This occurs in the event of any of the three classes of genetic perturbation of *lin-12*. The effects of individual *lin-12* genotypes on behavior are grouped together into classes of similar effects, referred to as "lin12(x)_Phen()", where x is 0, d or d/0. (B) The normal AC/VU decision reflects the ability of either Z1.ppp or Z4.aaa to acquire the AC fate. The effect of ablations on this behavior is incorporated via the use of conditional statements. Nested "If…then…else" statements provide conditions for all relevant outcomes. (C) A *lin-12(d)* genotype triggers a different AC/VU decision behavior. (D) Integration via conditions. See text for details.

Figure 7. The graphical user interface (GUI) and Excel datasheet data storage. (A) The structure of the GUI at the beginning of a simulation. (B) The final GUI structure at the end of the sample run indicated in the text. (C) Entries into the Excel datasheet from the sample run indicated in the text. Columns A-F list the fates of P3.p-P8.p, respectively. Columns G-L list the positions of P3.p-P8.p, respectively. Columns M and N list the fates of Z1.ppp and Z4.aaa, and columns O and P list the positions of these cells. The results of each simulation are listed in subsequent rows.

Figure 8. 1º fate specification in *lin-15(n309)* occurs very early. The *ayIs4[P_{egl-17}::GFP]* reporter (Burdine et al., 1998) was used to score 1º fate specification (GFP+) in each of the Pn.p cells (numbered along the *x*-axis) a *lin-15(n309)* (A) and a *lin-15(+)* (B) background at various timepoints. P6.p expresses the marker of 1º fate specification more frequently and more strongly than other Pn.p cells in *lin-15(n309)*. 1º fate specification is also shifted dramatically earlier in *lin-15(n309)* than in *lin-15(+)*. These results are consistent with an alternative, hypodermal source of the LIN-3 inductive signal rather than an AC source in *lin-15(n309)* animals (Burdine et al., 1998).

Table

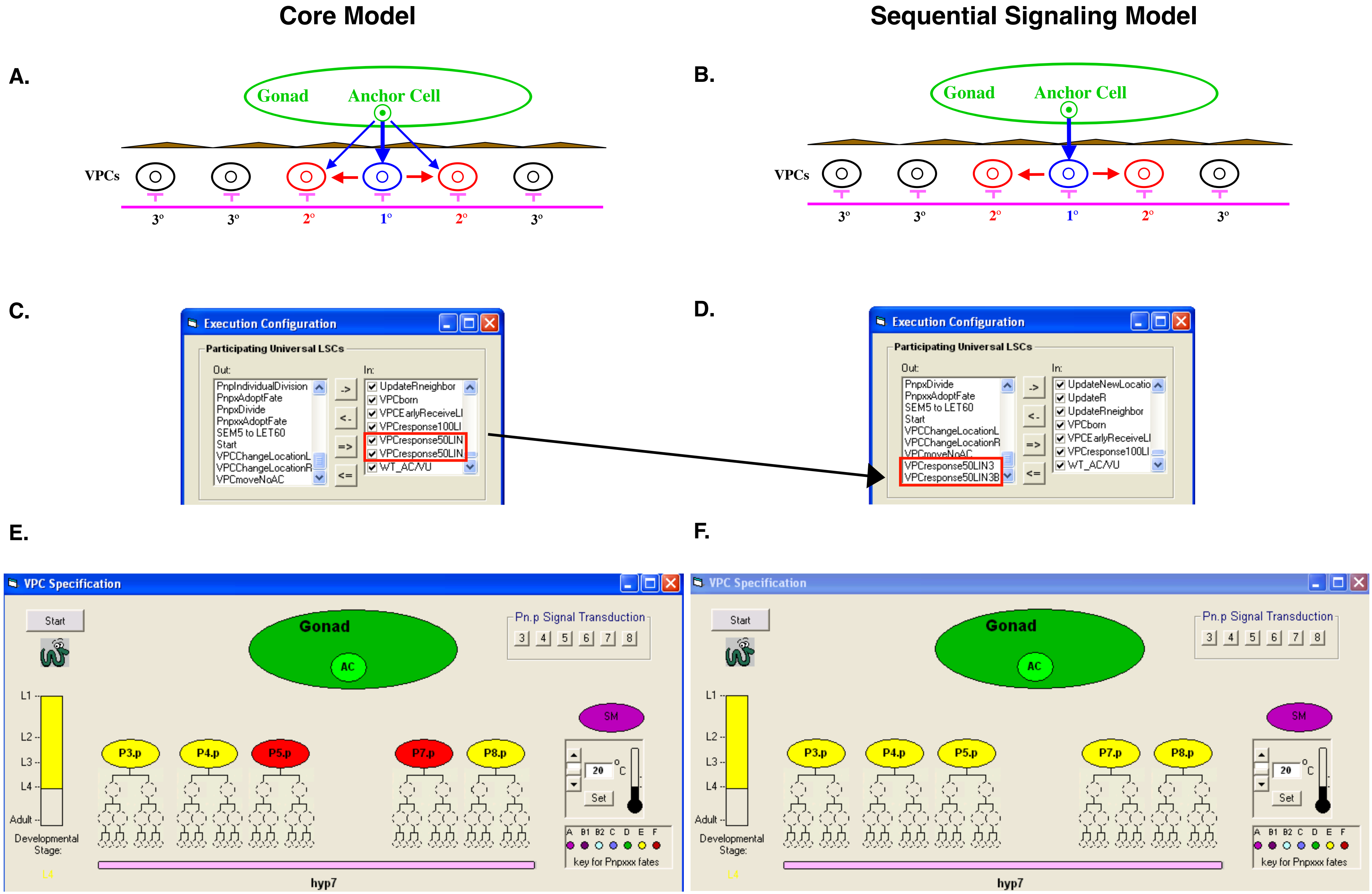
Table 1. List of uLSCs organized by Use Case.

Use Case	LSC Name	Abbreviated Statement of Purpose	Page # in Supplementary “Model Documentation”
Core Behaviors	DevelopmentalTime20c	Timing of the larval stages in the context of model time progression or ticks	2
	PnDivide	Ordering and timing of the division of the Pn cells	3
	VPCborn	Birth of P3.p - P8.p	3
	PnpAssumeFate	Time constraints for VPC fate assumption, especially relative to S-phase	4
ACVU	ACVUborn	Z1.ppp/Z4.aaa birth	5
	ACAdoptFate	Setting of the AC/VU fate	6
	WT_AC/VU	Timing and triggering of the AC/VU decision in WT	7
	AC/VUdecision	Nondeterministic AC/VU fate decision in WT	8
	lin12(d)_AC/VU	Timing and outcome of the AC/VU decision in *lin-12(d)* homozygotes	9
	lin12(d/0)_AC/VU	Timing and outcome of the AC/VU decision in *lin-12(d/0)* worms	10
	lin-12_AC	Timing and outcome of the AC/VU decision in *lin-12(0)* homozygotes, including AC positioning	11
	AC_WTposition	AC anterio-posterior position in WT	12
	Gonad2AC	AC/VU anterio-posterior position resulting from Gonad movement	13
	GonadAblation	AC/VU ablation upon Gonad ablation	13
	GonadAblation2	Triggering of AC/VU ablation upon interactive Gonad ablation	14
	dig-1	Effect of *dig-1* mutation on Gonad position	15
	lon-1	Effect of *lon-1* mutation on Gonad position	16
	Xloc2boxes	AC/VUs' anterio-posterior positions relative to the Locations	17
	ACFormedlocUpdate	Triggering of adjustment of VPC's record of its distance from	18

		AC upon AC movement	
	UpdateAC_location	Adjustment of VPC's record of its distance from AC upon AC movement	19
LIN-3	LIN-3expression	Timing of AC-derived LIN-3 expression, given WT *lin-3*	20
	LIN-3InBoxes	AC depositing of LIN-3 into proximal Locations	21
	Lin-3 50 Diffusion Anterior	Anterior diffusion from a Location with 100 units (a high level) of LIN-3	22
	Lin-3 50 Diffusion Posterior	Posterior diffusion from a Location with 100 units (a high level) of LIN-3	22
	Lin-3 20 Diffusion Anterior	Anterior diffusion from a Location with 50 units (a medium level) of LIN-3	23
	Lin-3 20 Diffusion Posterior	Posterior diffusion from a Location with 50 units (a medium level) of LIN-3	23
	Medium2100	Summing of two doses of 50 units (a medium level) of LIN-3 in a Location, resulting in 100 units (a high level) of LIN-3	24
	ClearLIN-3	Immediate clearing of LIN-3 levels in all Locations, upon ablation of both AC/VUs	24
	ClearLIN-3(b)	Immediate clearing of LIN-3 levels in all Locations, upon ablation of the one and only AC	25
	Boxes2VPCs(@StartSpec)	Transfer of LIN-3 in Locations to the resident VPCs	25
Pn.p Movements	ClearAblatedLocation	If a VPC is interactively ablated, its former Location updates its record of its resident cell, and the cell clears its own record of its neighbors	27
	BatchClearAblatedLocation	If a VPC is ablated, its former Location updates its record of its resident cell, and the cell clears its own record of its neighbors	28
	Box2MovingVPC	Transfer of LIN-3 from a Location to its new resident VPC	28
	ClearNeighbors	If a cell is ablated, its neighbors update their own records of their neighbors	29
	FreedUpdatesHalfs	Triggering of the updating of the Locations previously containing fragments of a now ablated or moving cell	30
	MoveLeft	A cell whose left neighbor has been ablated or has moved	31

		Phase to its right neighbor	
	PrimSendLSSphase(R2L)	A second dose of Lateral Signal is sent by a 1^o cell after S-Phase to its left neighbor	49
VPC Fate Assumption	VPCresponse100LIN3	1^o fate assumption upon receipt of high levels of LIN-3	50
	VPCresponse50LIN3	2^o fate assumption of an Undifferentiated cell upon receipt of medium levels of LIN-3	50
	VPCresponse50LIN3B	2^o fate assumption of a non-1^o cell upon receipt of medium levels of LIN-3	51
	VPCEarlyReceiveLS	non-1^o fate assumption by a cell that is Undifferentiated upon receipt of Lateral Signal before S-phase	52
	Sphase_receiveLS	2^o fate assumption by a cell that received Lateral Signal after S-Phase	53
	GroundStateNonVulval	3^o fate assumption by a cell that remained Undifferentiated even after all signaling had taken place	54
	PnpAdoptFate	Setting of the VPC fates	55
	FateOnce	A VPC cannot acquire two different fates (excluding non-1^o fates)	56
Hyp7 Inhibitory Signal	lin15n309	An *n309* homozygote executes a *lin15(n309)* phenotype, which is a *lin-15AB* phenotype	57
	lin15_GS	1^o and 2^o fate assumption in a *lin-15* mutant	58
Pn.p Fusion	P3p_fusion	Probability of P3.p fusion in all cases except a *lin-15* mutant	60
	P3p nonVPC	Nondeterministic decision of P3.p fusion and adoption of a nonVPC fate	61
	P3p_lin15AB	Probability of P3.p fusion in a *lin-15* mutant	61
Mechanics	L1sync	Update of the graphical representation of developmental time	63

		upon L1 entry	
	L2sync	Update of the graphical representation of developmental time upon L2 entry	63
	L3sync	Update of the graphical representation of developmental time upon L3 entry	64
	L4sync	Update of the graphical representation of developmental time upon L4 entry	64
	Start	User clicking of "Start" triggers Worm hatching	64
	BatchStart	The first time tick during a Batch-run triggers Worm hatching and activates the model’s reporting capabilities	65
	Hours1	At the start of a model run, each time tick represents an hour	66
	Hours2	After S-phase, each time tick again represents an hour	66
	Minutes	Upon L3 entry, model time ticks represent minutes	66
	TimeSync	Graphical representation of developmental time progresses appropriately according to whether ticks represent hours or minutes	67
Reporter	OpenReport	Opening of the Excel document for storage of model run results	68
	GatherData	Store AC/VU and VPC fates and positions in the Excel document at the end of a run	69



Core Model

Sequential Signaling Model

C. Execution Configuration

Participating Universal LSCs

Out:
PnpIndividualDivision
PnpxAdoptFate
PnpxDivide
PnpxxAdoptFate
SEM5 to LET60
Start
VPCChangeLocationL
VPCChangeLocationR
VPCmoveNoAC

In:
☑ UpdateRneighbor
☑ VPCborn
☑ VPCEarlyReceiveLI
☑ VPCresponse100LI
☑ VPCresponse50LIN
☑ VPCresponse50LIN
☑ WT_AC/VU

D. Execution Configuration

Participating Universal LSCs

Out:
PnpxDivide
PnpxxAdoptFate
SEM5 to LET60
Start
VPCChangeLocationL
VPCChangeLocationR
VPCmoveNoAC
VPCresponse50LIN3
VPCresponse50LIN3B

In:
☑ UpdateNewLocatio
☑ UpdateR
☑ UpdateRneighbor
☑ VPCborn
☑ VPCEarlyReceiveLI
☑ VPCresponse100LI
☑ WT_AC/VU

E. VPC Specification

Developmental Stage: L4

20 °C

F. VPC Specification

Developmental Stage: L4

20 °C

FIGURE 1

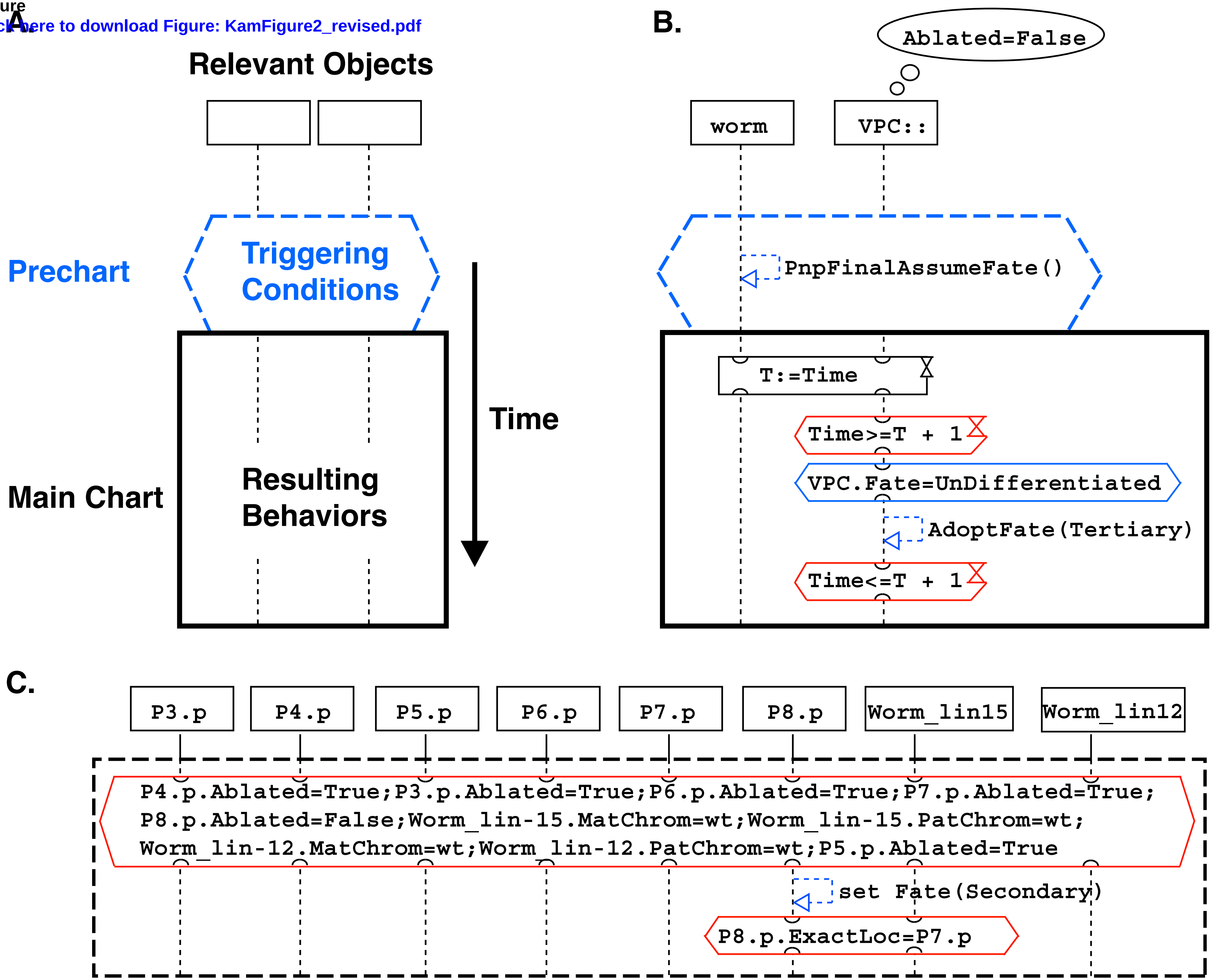

A.
Relevant Objects
Prechart
Triggering Conditions
Time
Main Chart
Resulting Behaviors
B.
Ablated=False
worm
VPC::
PnpFinalAssumeFate()
T:=Time
Time>=T + 1
VPC.Fate=UnDifferentiated
AdoptFate(Tertiary)
Time<=T + 1
C.
P3.p
P4.p
P5.p
P6.p
P7.p
P8.p
Worm_lin15
Worm_lin12
P4.p.Ablated=True;P3.p.Ablated=True;P6.p.Ablated=True;P7.p.Ablated=True;
P8.p.Ablated=False;Worm_lin-15.MatChrom=wt;Worm_lin-15.PatChrom=wt;
Worm_lin-12.MatChrom=wt;Worm_lin-12.PatChrom=wt;P5.p.Ablated=True
set Fate(Secondary)
P8.p.ExactLoc=P7.p

FIGURE 2

Figure 3

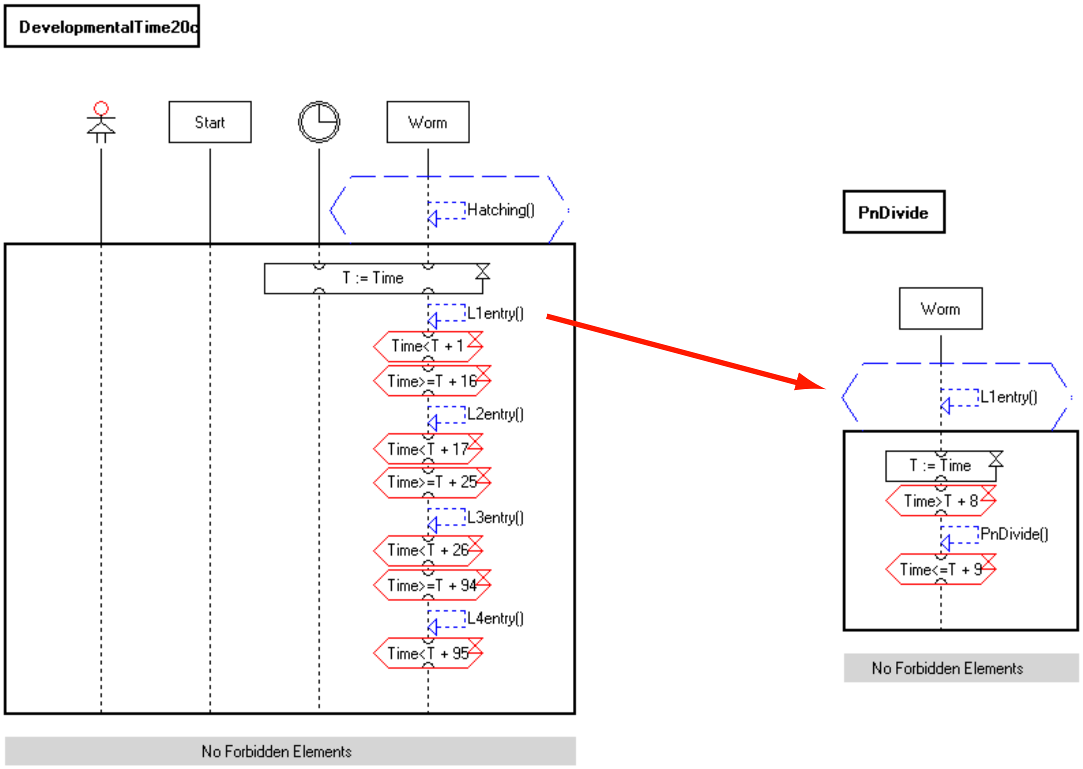
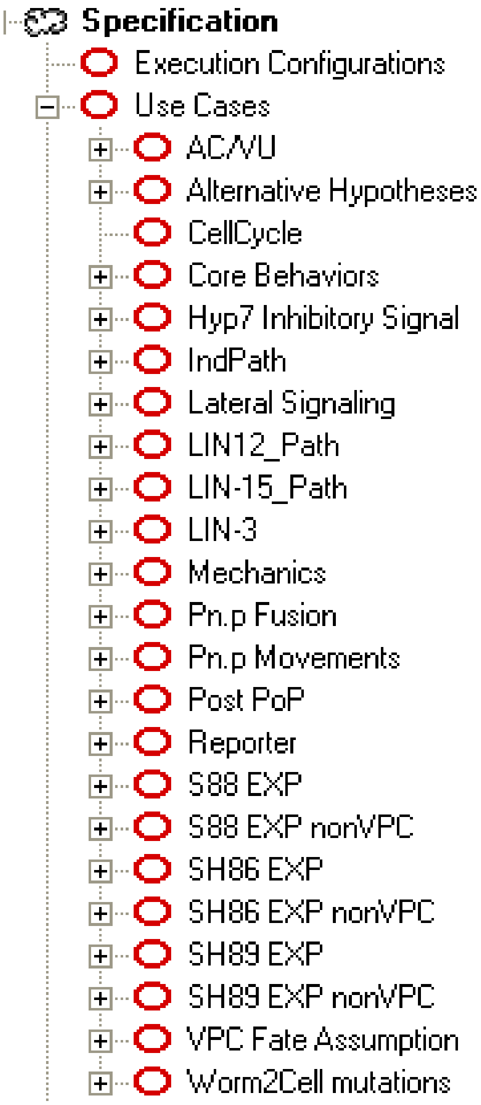

DevelopmentalTime20c
Start
Worm
Hatching()
T := Time
L1entry()
Time<T + 1
Time>=T + 16
L2entry()
Time<T + 17
Time>=T + 25
L3entry()
Time<T + 26
Time>=T + 94
L4entry()
Time<T + 95
No Forbidden Elements
PnDivide
Worm
L1entry()
T := Time
Time>T + 8
PnDivide()
Time<=T + 9
No Forbidden Elements

Figure 4A



- **Specification**
  - Execution Configurations
  - Use Cases
    - AC/VU
    - Alternative Hypotheses
    - CellCycle
    - Core Behaviors
    - Hyp7 Inhibitory Signal
    - IndPath
    - Lateral Signaling
    - LIN12_Path
    - LIN-15_Path
    - LIN-3
    - Mechanics
    - Pn.p Fusion
    - Pn.p Movements
    - Post PoP
    - Reporter
    - S88 EXP
    - S88 EXP nonVPC
    - SH86 EXP
    - SH86 EXP nonVPC
    - SH89 EXP
    - SH89 EXP nonVPC
    - VPC Fate Assumption
    - Worm2Cell mutations

Figure 4B

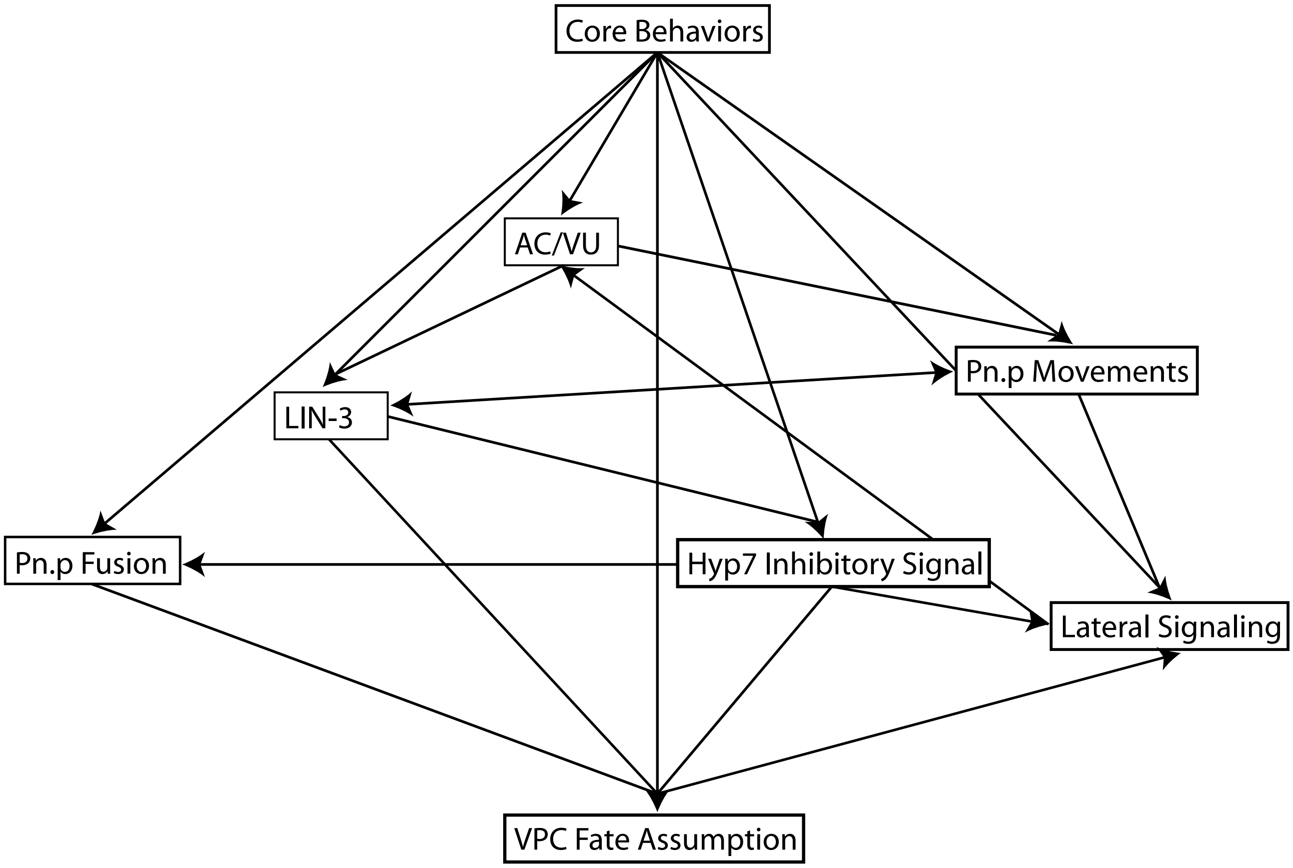

Core Behaviors
AC/VU
Pn.p Movements
LIN-3
Pn.p Fusion
Hyp7 Inhibitory Signal
Lateral Signaling
VPC Fate Assumption

Figure 5A

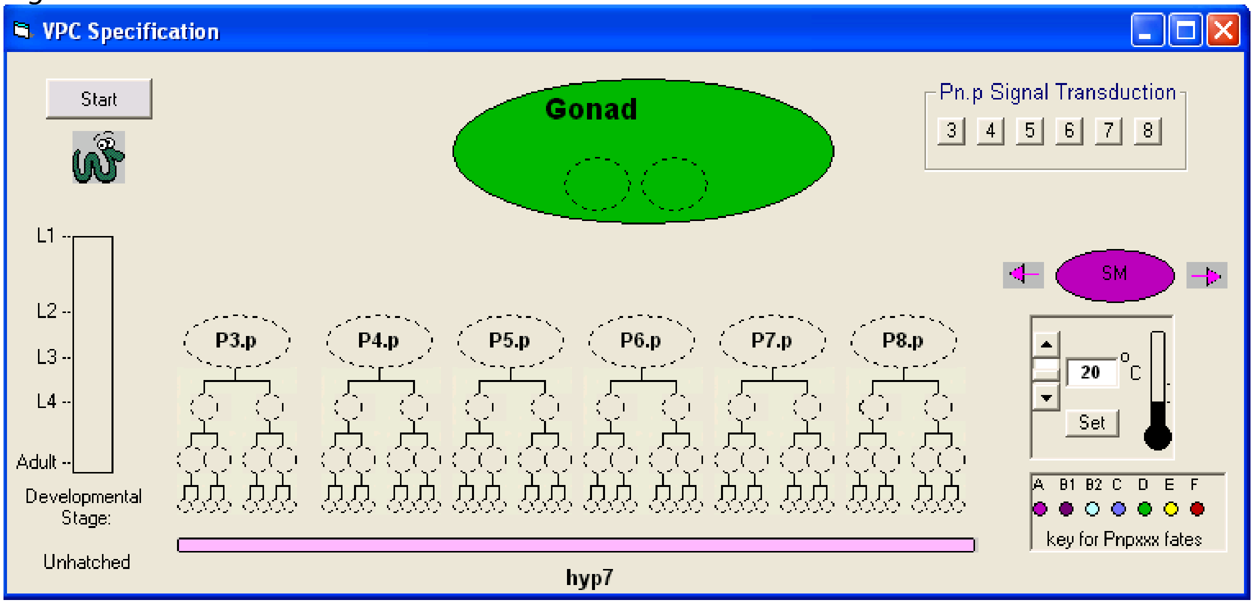


Figure 5B

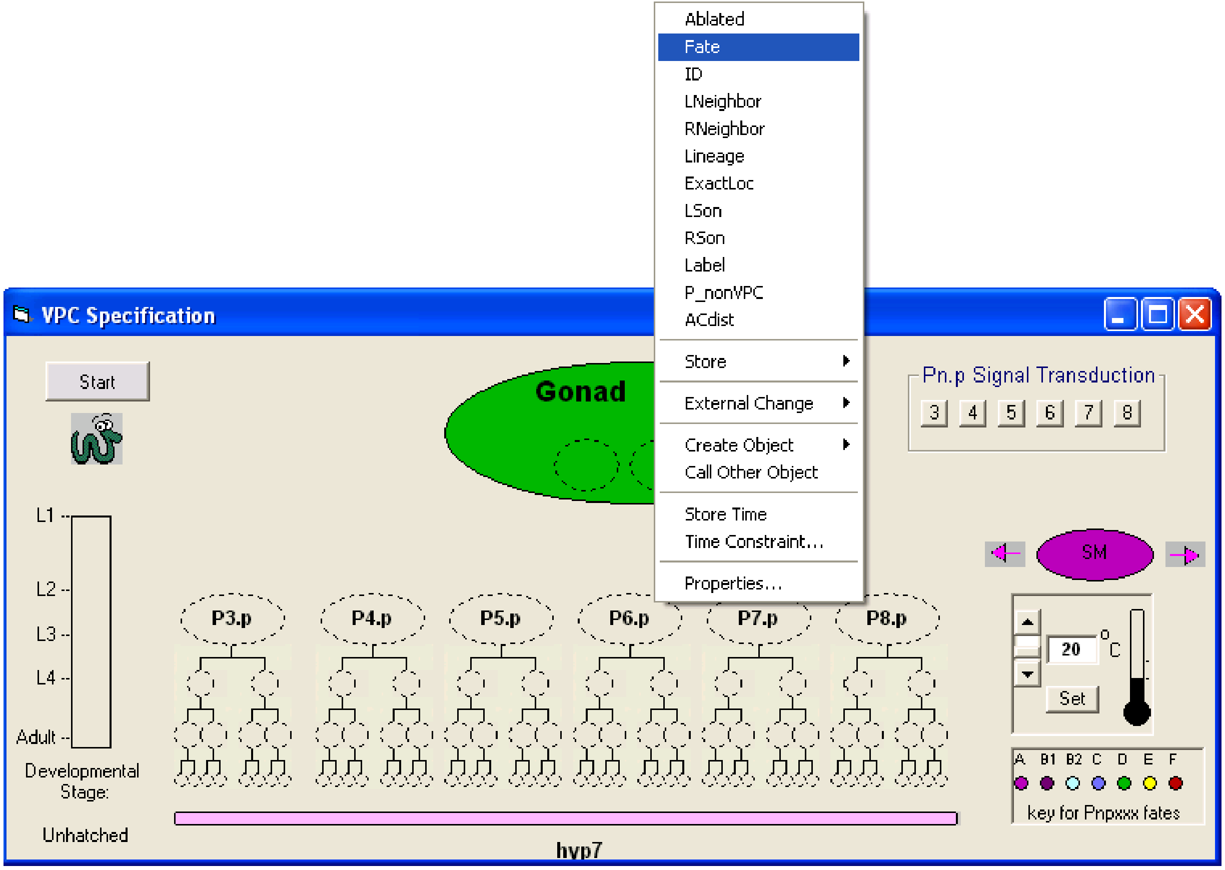
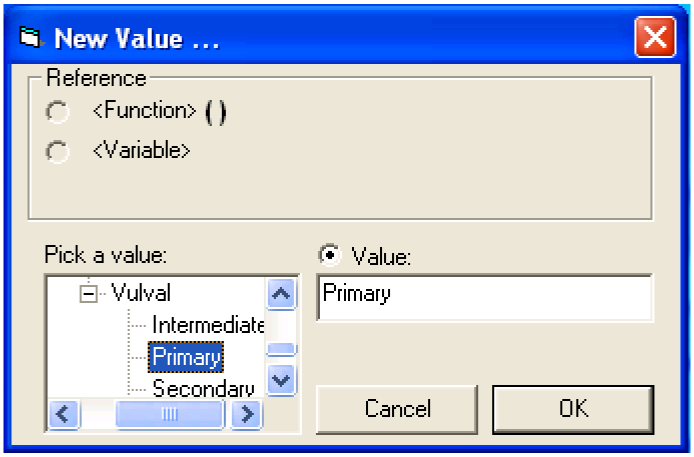


Figure 5C



New Value ...

Reference

○ <Function> ( )

○ <Variable>

Pick a value:

- Vulval
  - Intermediate
  - Primary
  - Secondary

● Value: Primary

Cancel

OK

Figure 5D

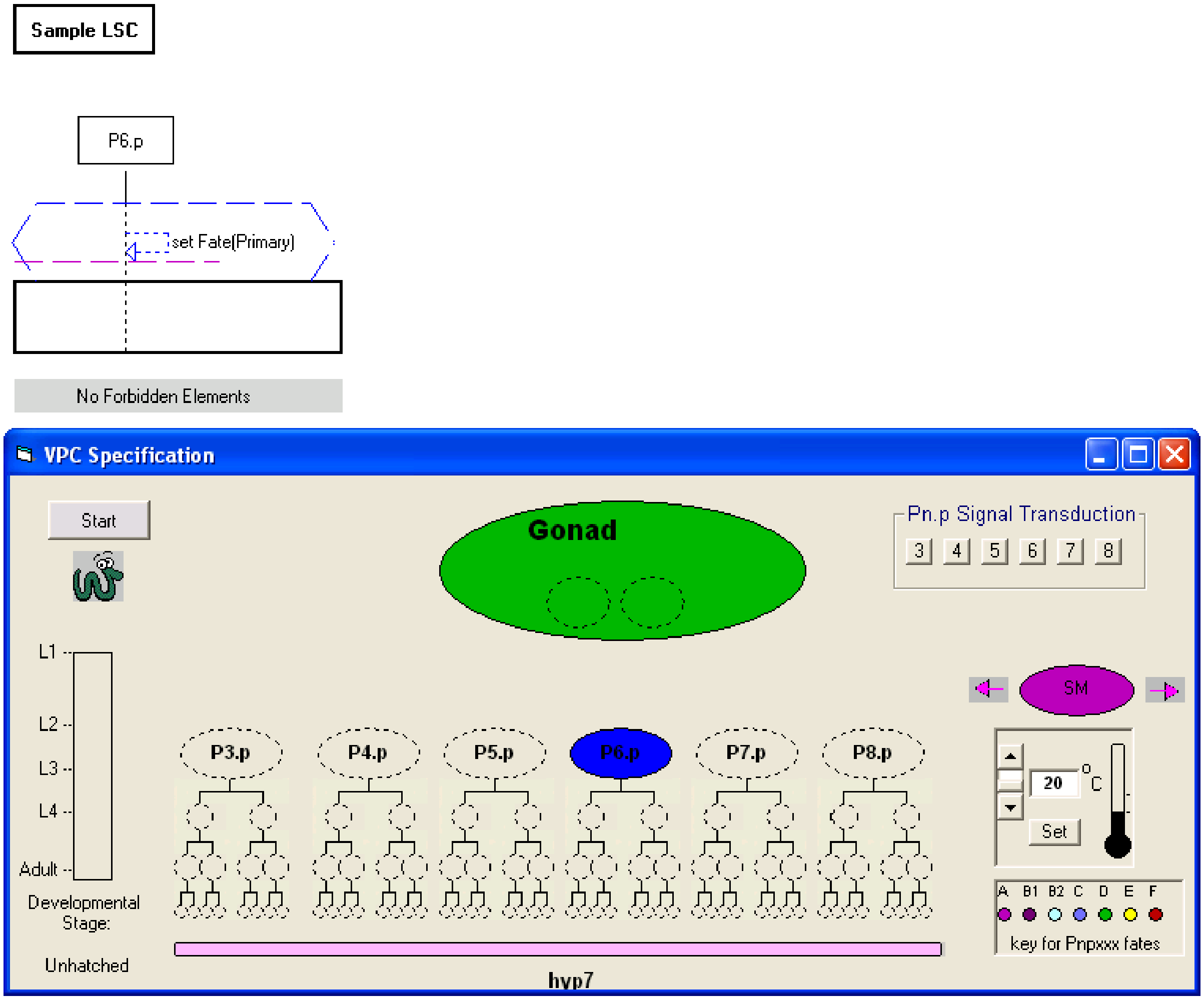


Figure 5E

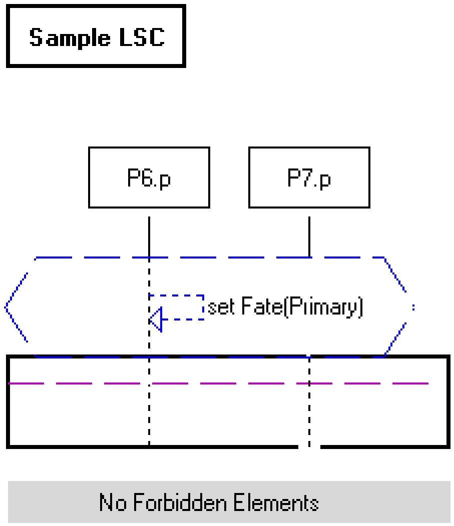


Figure 5F

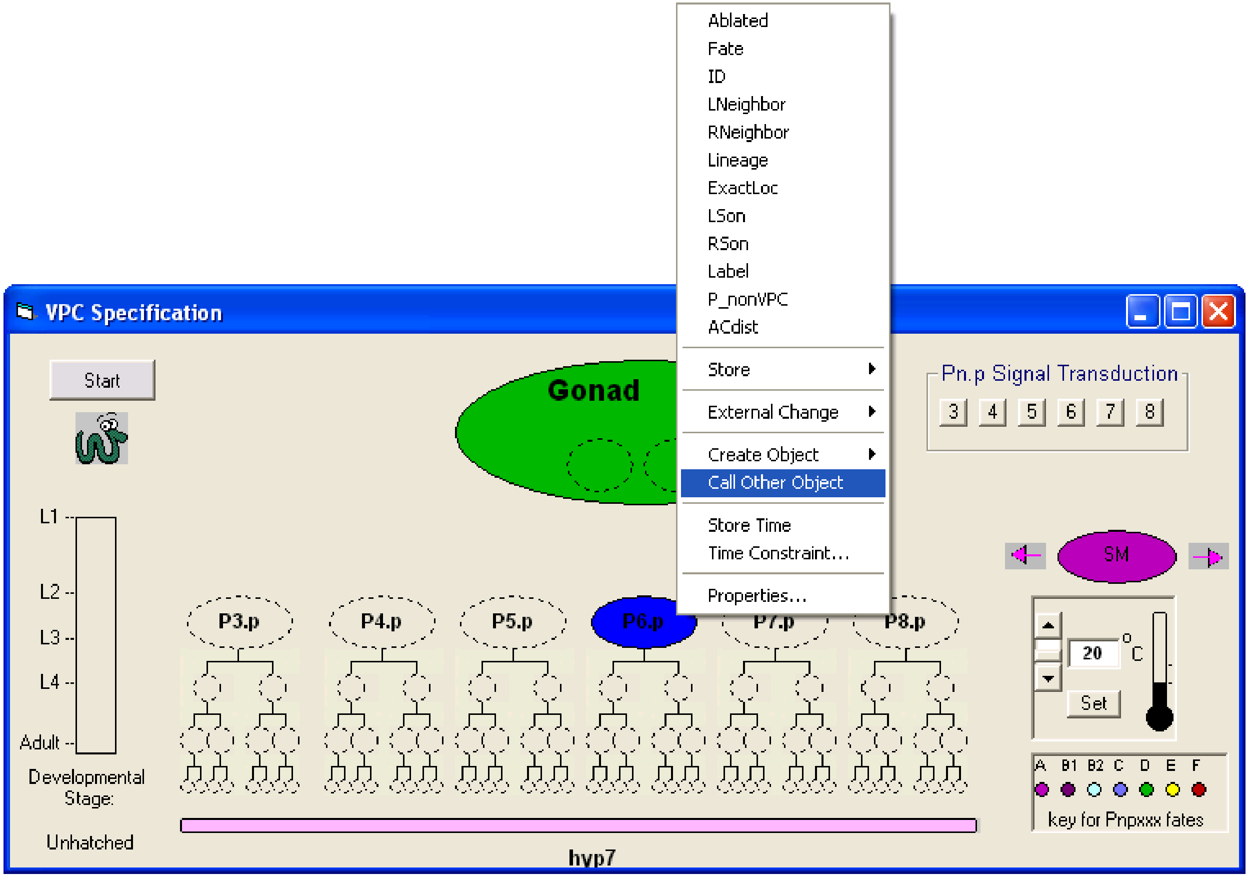


Figure 5G

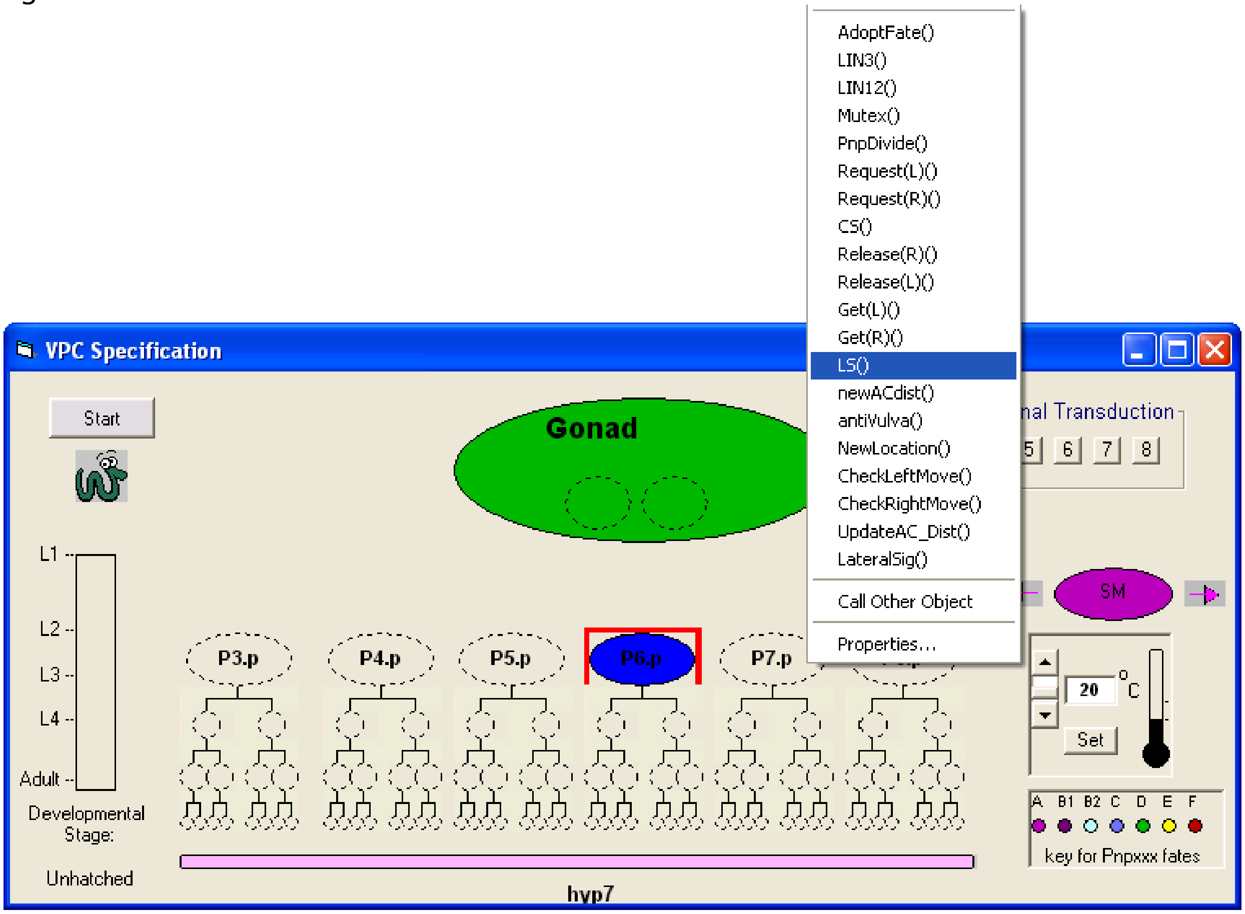
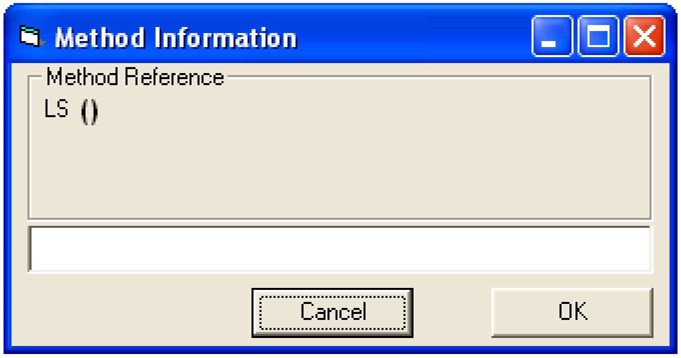


Figure 5H



Method Reference

LS ()

Figure 5I

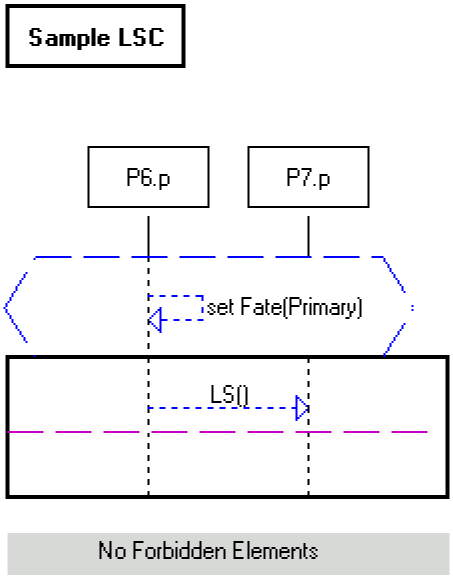


Figure 5J

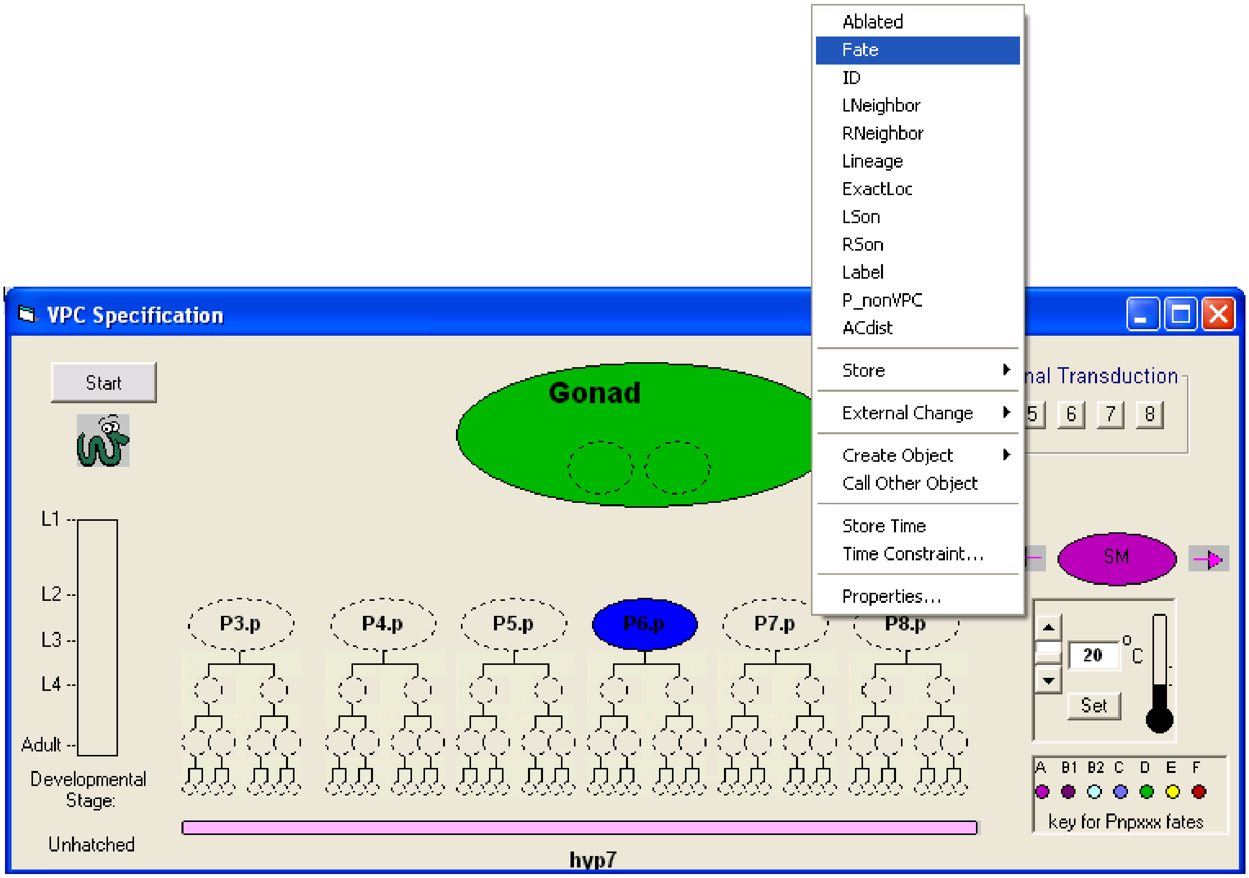
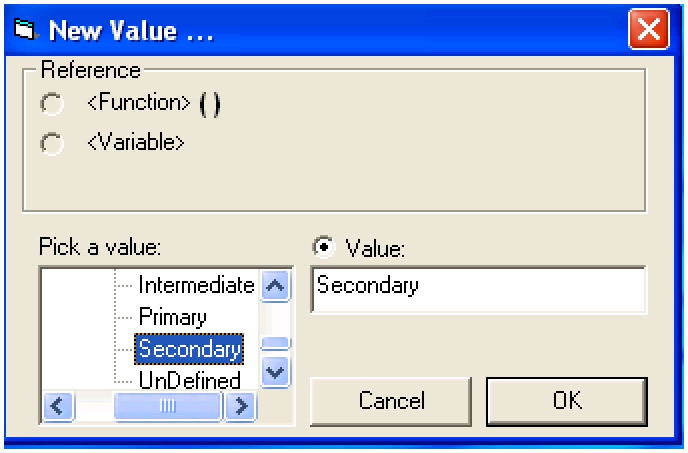


Figure 5K



New Value ...

Reference

○ <Function> ( )

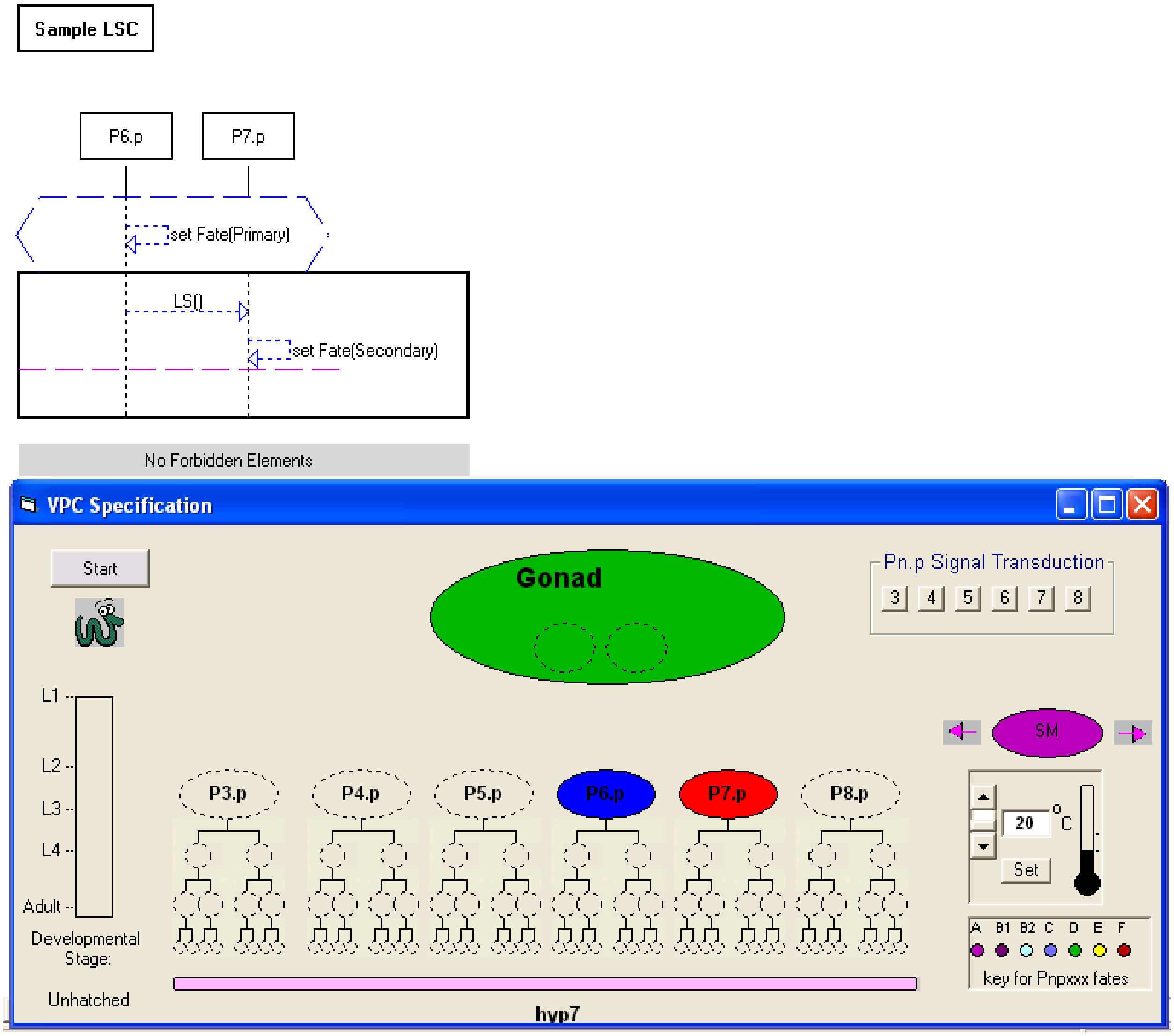
○ <Variable>

Pick a value:

- Intermediate
- Primary
- Secondary
- UnDefined

● Value: Secondary

Figure 5L



Sample LSC

No Forbidden Elements

VPC Specification

Figure 6 A, B, C

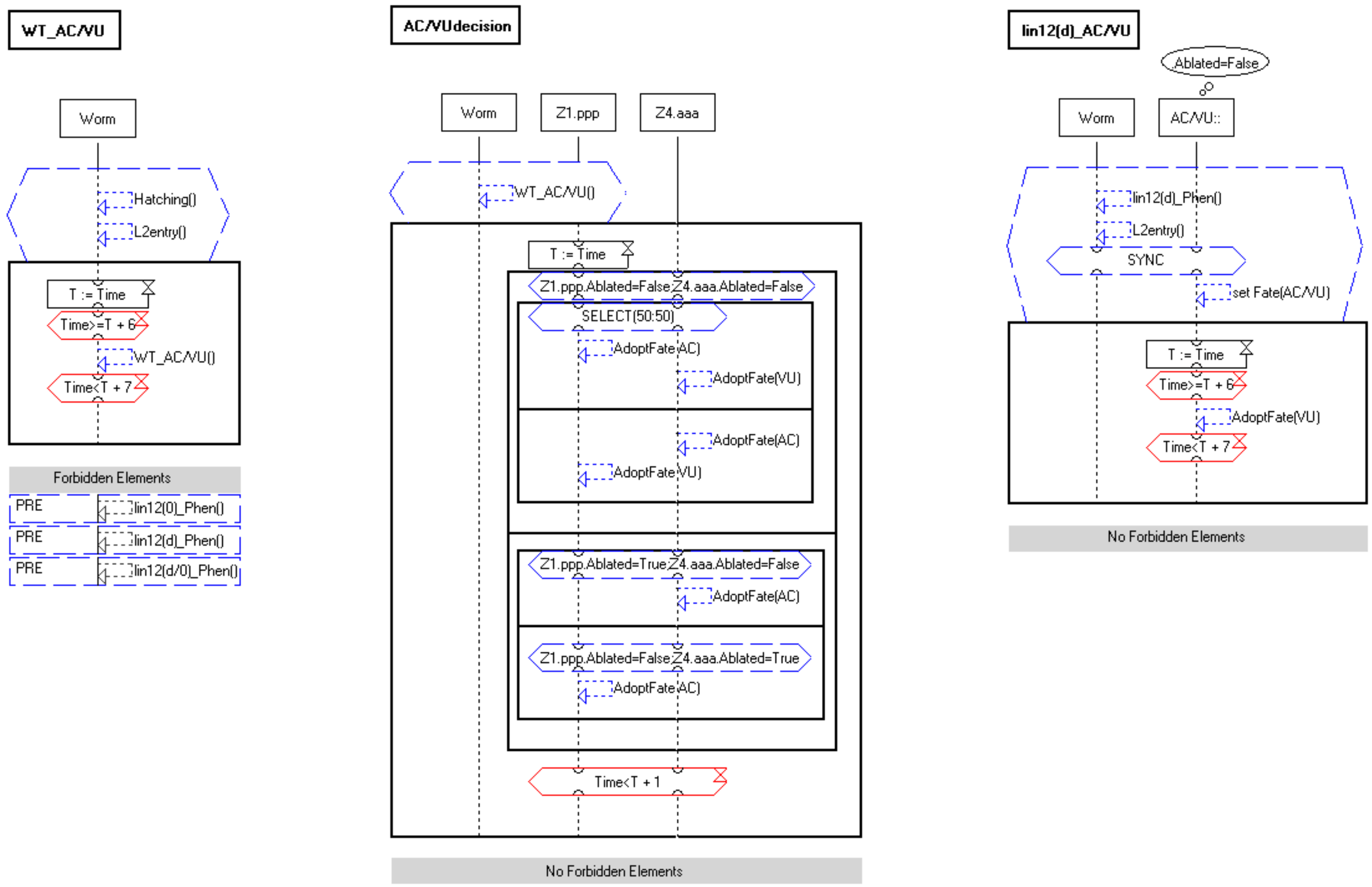
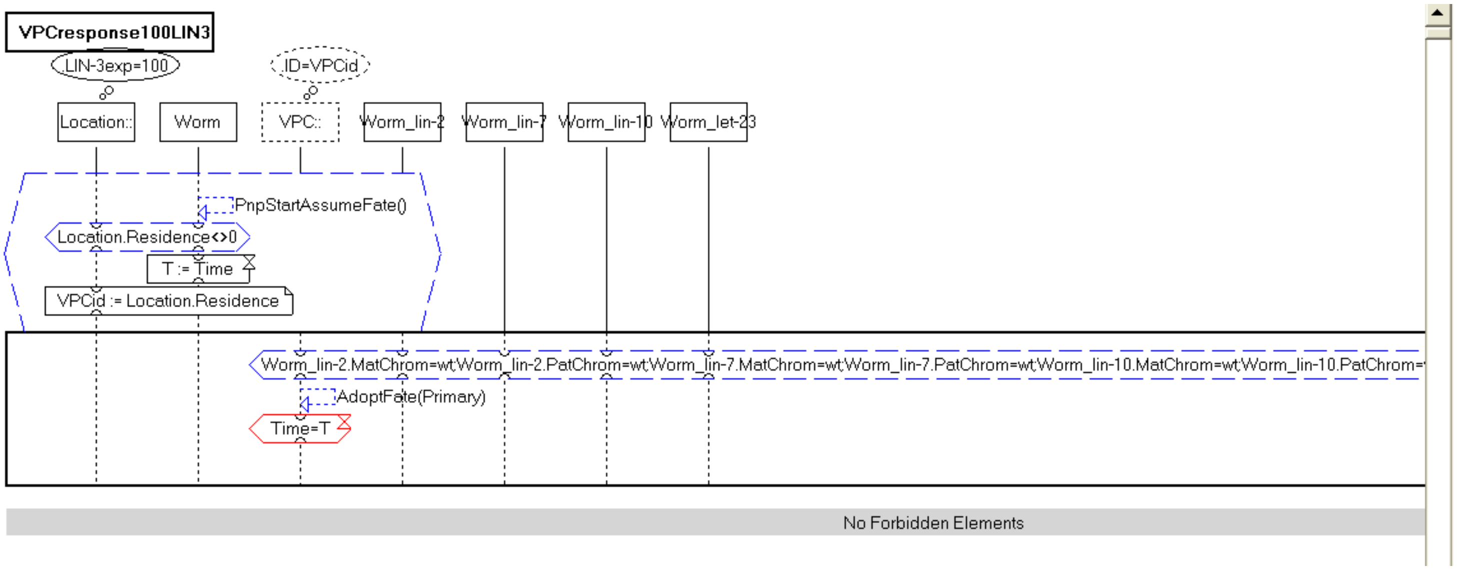


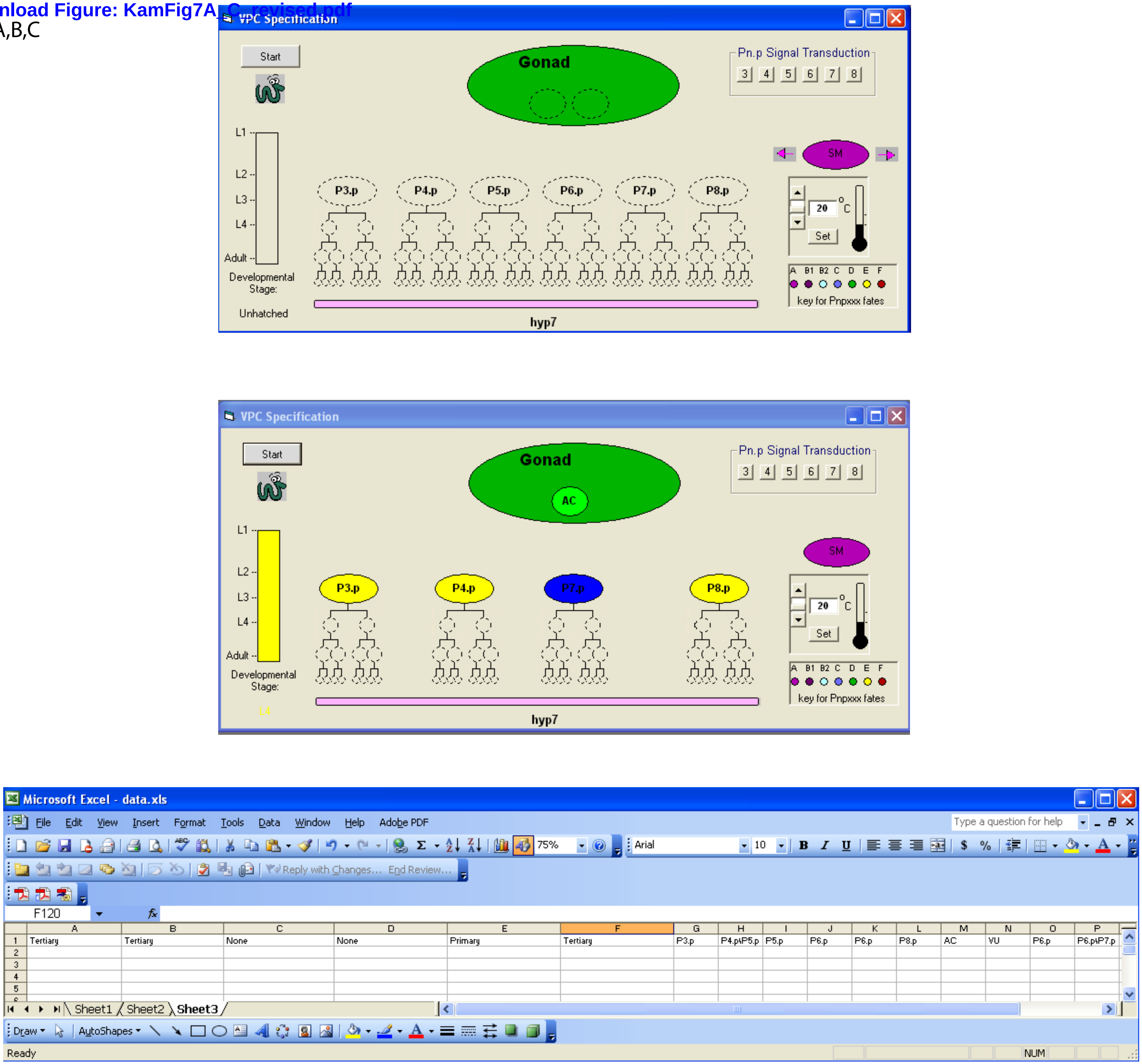
Figure 6D



Figure

Click here to download Figure: KamFig7A_C_revised.pdf

Figure 7A,B,C



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Tertiary	Tertiary	None	None	Primary	Tertiary	P3.p	P4.p\P5.p	P5.p	P6.p	P6.p	P8.p	AC	VU	P6.p	P6.p\P7.p
2																
3																
4																
5																

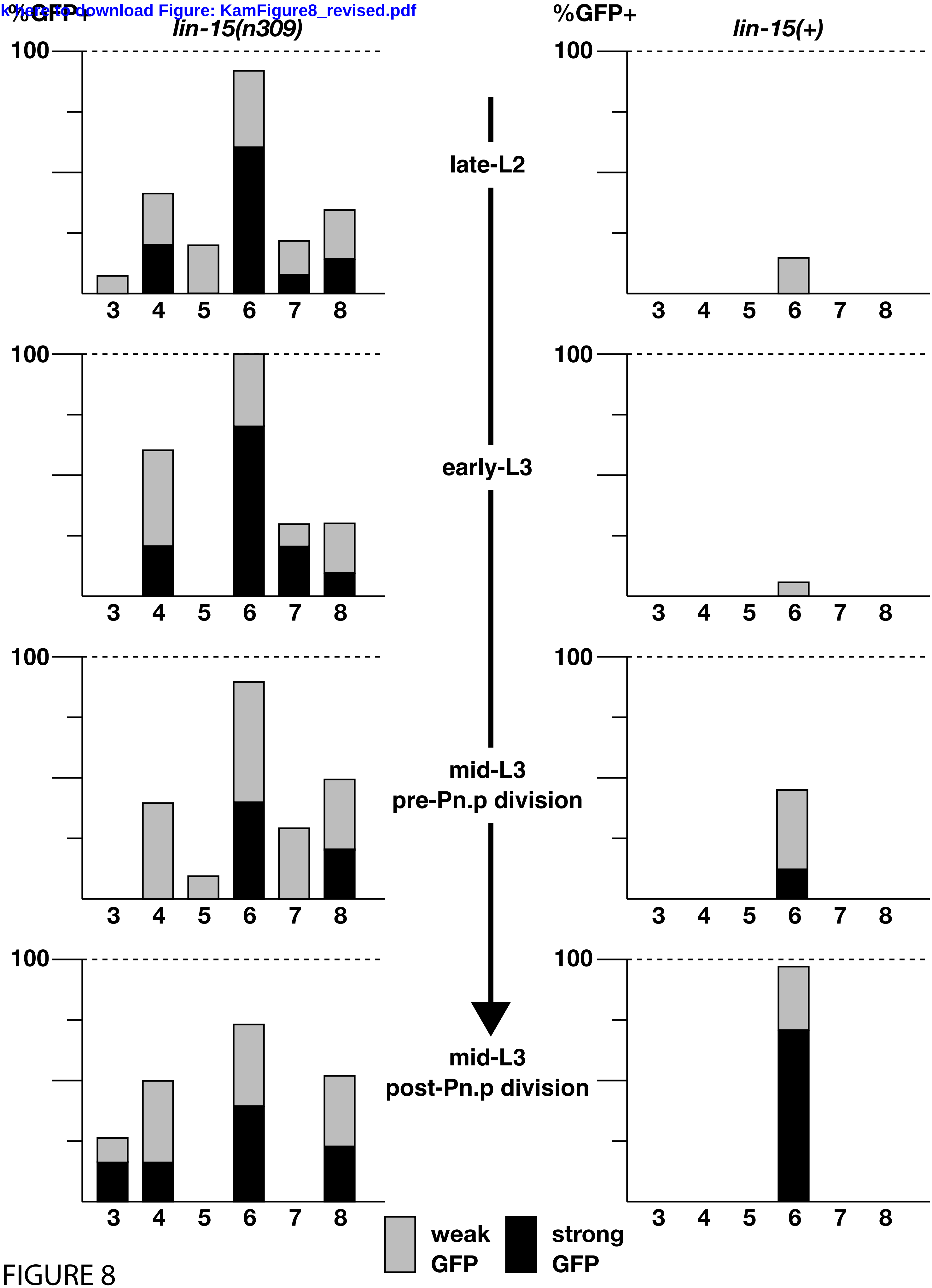

%GFP+
lin-15(n309)
%GFP+
lin-15(+)
100
3
4
5
6
7
8
late-L2
early-L3
mid-L3
pre-Pn.p division
mid-L3
post-Pn.p division
weak GFP
strong GFP

FIGURE 8

Supplementary Material

Click here to download Supplementary Material: kamsuppmovie1.mov

Supplementary Material

Click here to download Supplementary Material: kamsuppmovie3.mov

Supplementary Material

Click here to download Supplementary Material: kamsuppmovie4.mov

Supplementary Material

Click here to download Supplementary Material: KamSupplTABLE1_ABC.doc