Combining Analysis and Synthesis in a Model of a Biological Cell

Ken Webb
Department of Informatics
University of Sussex, UK
K.S.Webb@sussex.ac.uk

Tony White
School of Computer Science
Carleton University, Canada
arpwhite@scs.carleton.ca

ABSTRACT

We have previously described a top-down analytical approach, Cell Assembly Kit (CellAK), based on the object-oriented (OO) paradigm and the Unified Modeling Language (UML) and Real-Time Object-Oriented Methodology (ROOM) formalisms, for developing models and simulations of cells and other biological entities. In this approach, models consist of a hierarchy of containers (ex: cytosol), active objects with behavior (ex: enzymes, lipid bilayers, transport proteins), and passive small molecules (ex: glucose, pyruvate). In this paper we describe the Substrate Catalyst Link (SCL) bottom-up synthesis approach [17], the concept of autopoiesis on which it is based, and what we have learned in trying to integrate this approach into CellAK. The enhanced CellAK architecture consists of a network of active objects (polymers), each of which has behavior that causally depends partly on its own fine-grained structure (monomers), where this structure is constantly changing through interaction with other active objects.

Keywords

cell simulation; reactive systems; analysis; synthesis; autopoiesis.

1. INTRODUCTION

Biological cells and organisms are the most complex systems we know about. They develop naturally in a bottom-up manner without human purposeful intervention. Operating systems, telecommunication, air-traffic control, and other computer-based systems are rapidly becoming more complex, routinely involving tens of millions of lines of code [8]. These systems are generally developed in a top-down manner involving an ever-growing amount of purposeful error-prone intellectual effort. The resulting systems are typically fragile, difficult to manage and maintain. This observation has led to the Autonomic Computing initiative by IBM, where biological principles of organization are employed in system design.

SAC '04, March 14-17, 2004, Nicosia, Cyprus

Researchers have recently been looking to biology for inspiration, for ways to automate and simplify software development. Genetic algorithms, genetic programming, artificial neural networks, ant and other agent-based systems, are some of the more mature biologically-inspired technologies. Most of these approaches mine biology for ideas, and then abstract away from these ideas to produce algorithmic processes that can create problem solutions in a bottom-up manner.

However, biology offers more than just specific techniques. Living cells represent architectural plans that have been refined over millions of years of evolutionary history. They continuously adapt to evolving and hostile environments, can remain productive for 100 years or more with no external intervention, and basically run themselves, maintaining organization and function.

Biologists study various aspects of cells, typically analyzing them at greater and greater levels of detail. The three standard analytical aspects mentioned in biology textbooks [1] are (1) cellular structure, (2) cellular function through metabolic processes, and (3) genetic processing. The newer field of systems biology [12] offers a fourth more synthesis-based perspective.

We have been modeling and simulating cells with the ultimate goal of finding architectural principles that can be applied to the bottom-up production, management and maintenance of complex computer systems, thus automating and simplifying these processes. In a previous paper we have described a modeling approach, Cell Assembly Kit (CellAK), that integrates the traditionally studied cellular structure and cellular function aspects. CellAK is based on a small number of principles. It is constructed using the object-oriented (OO) paradigm, the Unified Modeling Language (UML) and Real-Time Object-Oriented Methodology (ROOM) formalisms, and the Rational Rose RealTime (RRT) tool, all commonly used for the top-down commercial development of complex real-time, embedded and other reactive systems. It is our hypothesis that tools and techniques that embody essential principles found in living organisms are required if we are ever to develop truly autonomic systems [8], and that such tools and techniques can be evolved from the current state of the art in Software Engineering. One such principle is that of autopoiesis [13] in which all parts of a system participate in the making of all other parts and of the organism as a whole.

In related work, a number of projects are attempting to simulate whole cells and organisms. The E-CELL project, with a biology focus, is working on the "grand challenge" of constructing whole virtual cells *in silica* [24]. Another "grand challenge", this time with a computer science focus, is that of "full reactive modeling of a multi-cellular animal" [7]. The Creatures computer game attempts to create a whole artificial organism by integrating a

variety of molecular and neural concepts [2]. Our motivation for this paper is the demonstration of the utility of standard software engineering tools and modified modeling process, coupled with the theory of autopoiesis, in modeling both intra- and inter-cellular processes, thereby providing empirical support for our hypothesis. Our long-term goal is the evolution of the software development process to integrate biological principles of organization, self-maintenance and information processing.

Ours is a gradual bi-directional process, in which state-of-the-art software development approaches are used to model and simulate more and more aspects of biology, in an effort to discover architectural principles that apply in a wide variety of biology and technology domains. These principles can offer inspiration for the evolution of new software development, management and maintenance approaches. Our work on CellAK independently supports the claim that "many kinds of biological systems exhibit characteristics that are remarkably similar to those of reactive systems" [11].

In this paper we describe enhancements to CellAK that are a first step towards incorporating a more systems level approach, based on the additional simple bottom-up principle of autopoiesis. Section 2 introduces the OO, UML, ROOM, Rose RealTime, and CellAK top-down approaches. Section 3 introduces autopoiesis theory and the SCL bottom-up model. Section 4 discusses the enhanced version of CellAK that incorporates the principles of autopoiesis, and provides a first step towards a general approach to integrating top-down and bottom-up development. Section 5 suggests future work to confirm the utility of this general approach.

2. TOP-DOWN ANALYSIS

Object-oriented (OO) approaches have become the dominant paradigm for software development, especially for commercial and other applied systems developed in a top-down manner. The Unified Modeling Language (UML) [21] has become the standard common visual notation for analysis, design and implementation of OO systems.

David Harel, originator of the hierarchical state diagram (statecharts) formalism used today in UML [4], and an early proponent of visual formalisms in software analysis and design [5], has argued that biological cells and multi-cellular organisms can be modeled as reactive systems using real-time software development tools [6], [11]. Two such tools are I-Logix Rhapsody [9] and Rational Rose RealTime [20]. Rational Rose RealTime (RRT) was used to implement the system described in this paper.

RRT is a visual design and implementation tool for the production of telecommunication systems, embedded software, and other highly concurrent reactive systems. It combines the features of UML with the real-time specific features and visual notation of the Real-time Object-Oriented Modeling (ROOM) [22]. A RRT application's main function is to react to events in the environment, and to internally-generated timeout events, in real-time.

The powerful combination of the OO paradigm as embodied in the UML and ROOM visual formalisms with the added flexibility of the C, C++ or Java programming languages, integrated in a development tool such as RRT, provide much that is appropriate for biological modeling. In fact, the advantages of “agent-based” modeling when compared to traditional modeling approaches for biological systems are only now being appreciated [10].

2.1 CellAK

CellAK provides an approach to modeling and simulating cells and other biological entities. It is based on the OO paradigm and the UML and ROOM visual formalisms, includes a multi-step process for development of a model, and has been implemented using RRT. With CellAK it is possible to start with a direct diagrammatic representation of a biological structure such as a cell, using terminology familiar to biologists, and by following a process of gradually adding more and more detail, arrive at a system with structure and behavior (the cellular structure and metabolic cellular function described in biology textbooks) of arbitrary complexity that can run and be observed on a computer [26]. Recent work on domain-specific modeling [19] emphasizes the importance of designing at an appropriately high level of abstraction, using concepts and terminology from the problem domain (e.g. biology) rather than from the solution domain (e.g. computer code).

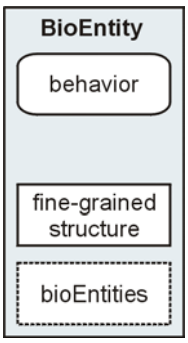

Figure 1: BioEntity

A CellAK BioEntity, shown in Figure 1, can be directly implemented as a capsule, a RRT primitive. It may contain behavior, fine-grained structure, and instances of other bioEntities. Behavior in the form of UML state diagrams and C++ state transition code is used to configure the system so that all entities that need to interact are in an adjacency relationship with each other. Once configuration is complete, a potentially large number of active bioEntity subclasses such as enzymes, lipid bilayers, and transport proteins are scheduled at regular intervals to act (using short segments of C++ code in their behavior) on collections of small molecules that make up the fine-grained structure of other bioEntities they are adjacent to. The resulting system models metabolic or other biochemical networks of a eukaryotic cell, the type of cell found in humans and other multi-cellular organisms. When modeling metabolism, it is able to produce quantitative results [26] very similar to those produced by Gepasi [18], an ordinary differential equation (ODE) based biochemistry modeling tool.

BioEntities may contain any combination of behavior, fine-grained structure, and contained bioEntities. Three distinct types can be distinguished. A *pure active object* contains only behavior. A *pure passive object* contains only fine-grained structure (small molecules) that is acted on by other bioEntities. A *pure container object* only contains other bioEntities (although it may have initial behavior to configure the contained bioEntities). Table 1 summarizes these types and provides examples.

Table 1: BioEntity types found in the CellAK Model.

	Behavior	Fine-grained Structure (FGS)	Contained bioEntities	Examples
Pure Active Object	√			Enzyme TransportProtein CellBilayer
Pure Passive Object		√		Cytosol ExtraCellularSolution
Pure Container			√	EukaryoticCell Cytoplasm
	√	√		
	√		√	Enhanced CellAK Cytosol
		√	√	
	√	√	√	Enhanced CellAK CellBilayer

3. BOTTOM-UP SYNTHESIS

3.1 Autopoiesis

Neuroscientists Humberto Maturana and Francisco Varela [13], in an attempt to define and characterize both life and cognition, developed the *autopoiesis* theory. The term *autopoiesis* means "self-making". All entities in an autopoietic system or network participate in the creation and continual transformation of other entities, and hence participate in the creation and renewal of the autopoietic system itself in part by creating a boundary around it. This boundary in turn is of critical importance in creating the right conditions for creating the active entities within the autopoietic system. Everything is created from within, except for the basic building blocks of matter and energy. Maturana, Varela, and others have proposed the biological cell as validation of autopoiesis theory.

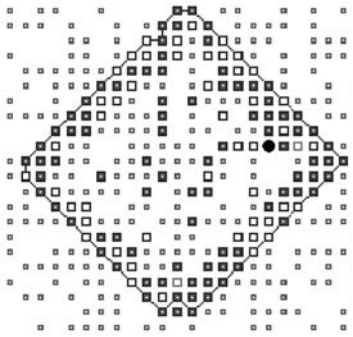

Figure 2: SCL-GRO

3.2 Varela & McMullin Models of Autopoiesis

Varela and colleagues produced a model of a cell as "a simple embodiment of the autopoietic organization" [25]. The model contained three types of entities randomly moving in a two-dimensional grid. A single type of catalyst (CellAK: enzyme) converts a small population of substrates (CellAK: small molecules) into links (CellAK: lipids). For the remainder of the paper these will be referred to using CellAK terminology. The lipids are capable of bonding, and may form a continuous membrane around the internal enzymes and small molecules, an example of bottom-up processing synthesizing new structure.

McMullin has more recently produced updated versions of this model. Substrate Catalyst Link (SCL) [14],[15],[16], is a re-implementation of the 1974 model using Swarm [23] and Objective C [3]. SCL-GRO [17] extended this to address various issues related to the take-up and maintenance of links in the membrane. In addition to small molecules, lipids, and enzymes, SCL also distributes holes (CellAK: water molecules) throughout the grid to fill otherwise vacant space.

Figure 2, [17], is a screenshot of SCL-GRO in action. The small squares are small molecules; the large unfilled and large filled squares are lipids without and with absorbed small molecules respectively; the filled circle in the middle right part of the figure is an enzyme; vacant spaces are water molecules. At this point in the simulation, a continuous membrane of lipids has formed, which appears as a large diamond in the figure. Three distinct concentric layers are visible, surrounded by an exterior extra cellular solution that only contains passive small molecules and water. The innermost layer consists of a single enzyme, small molecules, free lipids, and water. The middle layer consists of a higher concentration of free lipids and other small molecules. Because they are in the vicinity of the membrane, these lipids are ready to become part of that structure. The synthesized membrane layer consists of bonded lipids (shown connected by lines), and small molecules on their way between the inside and outside of the cell.

4. ENHANCED CellAK – INTEGRATION WITH SCL AUTOPOIESIS MODEL

The bioEntities of the original CellAK were previously shown in Figure 1 and described in Table 1. An enhanced version of CellAK adds the causal dependency line shown in Figure 3. In bioEntities that contain both behavior and fine-grained structure (FGS), the behavior may be at least partly dependent on details of that FGS.

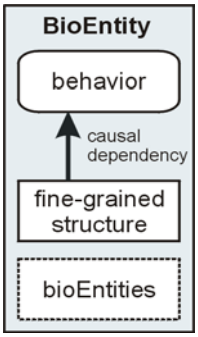

Figure 3: BioEntity with dependency

Figure 4 is a conceptual view showing how five bioEntities might interact and establish a dependency network. The container C_i has no behavior or FGS, but only contains other bioEntities. P_i is a passive object that only contains FGS, used by the behavior of active objects A_1, A_2, A_n as a blackboard. All three active objects continuously change values of attributes in the FGS within P_i, and in turn base their behavior partly on those values. The behavior of A_1 also depends on values in its own FGS and within the FGS of A_2. The FGS of A_1 is in turn modified by A_n. A_n also contains additional bioEntities at least some of which modify and are in turn causally dependent on the FGS of A_n. The behavior of A_2 modifies the FGS of some bioEntity outside the container C_i. The FGS of P_i is modified by some outside bioEntity.

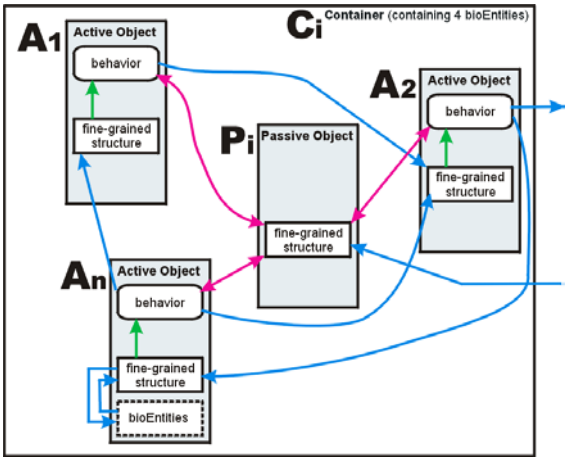

Figure 4: A BioEntity Network Diagram

Note the dependency chain between A_1, A_2 and A_n, by following the arrows: A_1 behavior → A_2 fgs → A_2 behavior → A_n fgs → A_n behavior → A_1 fgs → A_1 behavior, where the arrow signifies "has an influence on". The three active objects are also mutually dependent through their blackboard interactions with P_i. In short, there is a considerable amount of complexity in the configuration shown in Figure 4, of the type typically found in biological systems.

Figure 5 shows the bioEntity network diagram in the enhanced version of CellAK to implement the type of functionality found in the original model of autopoiesis [25] and the SCL autopoiesis model [17]. EukCellsInSolution, EukaryoticCell, CellMembrane, and Cytoplasm are container bioEntities. Cytoplasm contains an arbitrary number of Enzyme bioEntity active objects. Each Enzyme has a simple behavior that catalyses the transformation of substrate small molecules into products. The directed arrow labeled 0 represents many reactions that together make up the metabolism of the Cytosol, including the multi-step glycolytic pathway that converts glucose into pyruvate and creates lipid precursors.

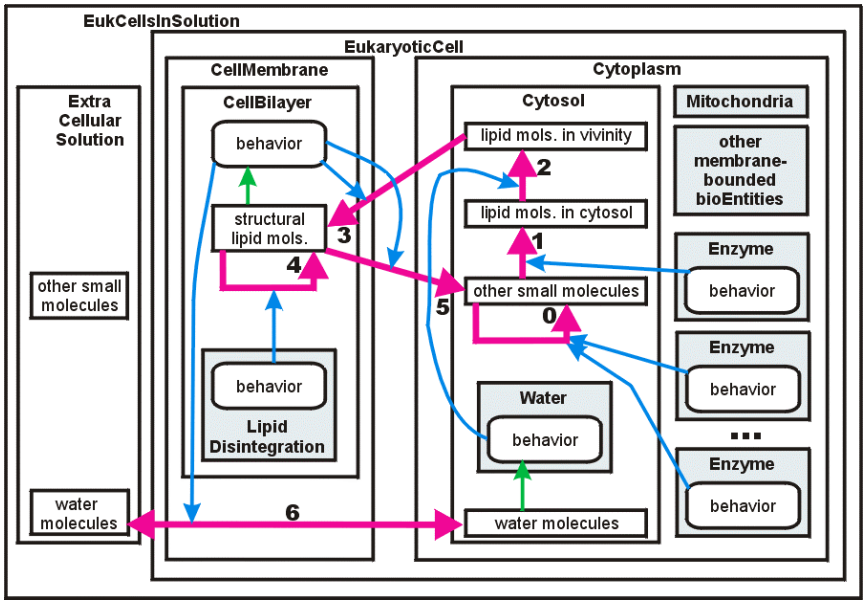

Figure 5: BioEntity Network Diagram

Cytoplasm also contains a single instance of Cytosol, a passive bioEntity. The FGS of Cytosol includes water molecules, lipids, and other small molecules. There are two types of lipids mirroring a distinction made in SCL. In this simplified model, when lipids are first created by an enzyme operating on some precursor molecule (the transformation represented by directed arrow 1), they are freely mixed within the Cytosol. They are subsequently pushed outward to the vicinity of the CellMembrane (arrow 2), effectively implementing the affinity feature found in SCL [17]. It should be noted that this simplistic way of creating and incorporating new lipids into a bilayer is by far the least common actually found in nature, although it does capture the required abstraction. The activity in a real cell involves the creation of new lipids directly into a bilayer with the subsequent budding off of spherical sections of bilayer for regulated incorporation elsewhere.

In addition to being a passive bioEntity, Cytosol is also modeled as a container that contains an instance of Water. Water is an active bioEntity whose behavior is dependent upon the changing quantity of water molecules found in the FGS of Cytosol. Water is the active object that gradually (over a number of time steps) pushes free lipids toward the CellMembrane. In the real world, lipids have a hydrophobic (water-hating) end that is repelled by water.

The CellMembrane container includes a CellBilayer bioEntity. CellBilayer is an active object with a behavior, and a FGS consisting of lipid molecules. The behavior selectively transports certain small molecules, including water (arrow 6) between the inside (Cytosol) and outside (ExtraCellularSolution) of the EukaryoticCell, incorporates lipids into its FGS (arrow 3), and pushes disintegrated lipids back to the Cytosol (arrow 5). The amount of activity in each time step is proportional to the quantity of lipids within its FGS. CellBilayer also contains LipidDisintegration, an active bioEntity whose behavior continuously breaks down (arrow 4) lipid molecules in the CellBilayer. If the proper balance of lipid creation and disintegration is not maintained in the model (or in the real world), then eventually the quantity of lipids in CellBilayer would reach 0, at which point it would cease to exist in any functional sense.

Cytoplasm also contains Mitochondria and other membrane-bounded bioEntities. Each of these has an internal structure that is quite similar to that of EukaryoticCell, complete with enzymes, a solution containing small molecules, and a bilayer that selectively moves small molecules into and out of the Cytosol.

The numbered arrows in Figure 5 represent three different molecular pathways. Arrow 0 represents the many metabolic pathways found in the Cytoplasm. Arrows 1, 2, 3, 4, 5 represent a cyclical sequence of steps that are found in both SCL [17] and CellAK. This is the cycle of (1) lipid production, (2) hydrophobic movement of lipids toward the vicinity of the CellBilayer, (3) bonding of the lipids with the CellBilayer, (4) the disintegration of lipids, and (5) bond decay. Arrow 6 represents the diffusion of water. The result is the constant creation and destruction of cellular components. This satisfies one of the requirements of an autopoietic system, that it outlive the lifetimes of its constituent components [17]. It is essential that, at the same time that lipids are being created and destroyed, the CellBilayer retain its organizational structure and its function of selectively moving small molecules in and out of the cell.

Note the interdependent activity. The ability of Water to transport lipid molecules depends on the quantity of water molecules, and the ability of CellBilayer to transport water molecules across the bilayer depends on the quantity of structural lipid molecules.

In summary, the original CellAK did not include the ability for an active object such as CellBilayer to include fine-grained structure (FGS) such as structural lipid molecules, and for the behavior to be causally dependent on this FGS. It did not include the Water and LipidDisintegration active objects, did not include the lipid molecules found in the FGS of Cytosol and CellBilayer, and did not include the transformations represented by arrows 1, 2, 3, 4, 5. The main innovations with the enhanced CellAK as described in this paper are (1) the optional addition of fine-grained structure to active bioEntities such as CellBilayer, (2) the causal dependency of the behavior of such active bioEntities on the values of attributes within this FGS, and (3) the ability for the behaviors of other active bioEntities to modify the contents of this FGS. The Varela and McMullin autopoiesis models suggested the need for these enhancements.

5. CONCLUSIONS AND FUTURE WORK

This paper has described a modeling approach and tool, CellAK, developed using principles from software engineering that is suitable for application to sophisticated cell modeling. The quantitative validation of the model against GEPASI has demonstrated the correctness of the implementation [26]. The visual nature of the tool is considerably simpler to understand when compared to conventional differential equation based models and, being container based, can more effectively support system level models proposed by Tomita. [24] We believe that this paper clearly confirms the value of agent-based modeling reported in [10].

Table 2: Four types of small molecule.

Polymer	Repeating Monomer	Number of repeating unit types
Proteins (ex: enzymes)	Amino acids	20
Nucleic acids (DNA, RNA)	Nucleotides	4 in DNA 4 in RNA
Lipid bilayers	Lipids	A few
Polysaccharides (ex: starch)	Monosaccharides (ex: glucose)	One or a few

Much has been learned since the initial version of CellAK was developed in 1999. Specifically, this paper has described an approach toward making the predominantly top-down CellAK more responsive to bottom-up processes, and illustrated this using the cell bilayer active object as an example.

Clearly other modeling work is possible. Other active objects in CellAK (polymers) are also composed of repeating units of small molecules (monomers). Becker [1] states that there are three major types of polymers in a cell. Table 2 adds lipid bilayers which are also made up of monomers.

This suggests a general principle. Active objects have an influence on other active objects in CellAK by having an effect on their constituent monomers. This enhancement should now be implemented for enzymes, transport proteins, and other proteins in CellAK. However, proteins are considerably more complex than lipid bilayers. The amino acids that constitute a protein are coded for in the DNA, the order of amino acids is of critical importance, and the string of amino acids folds into a three-dimensional shape. The behavior of a protein is therefore an extremely complex function of its fine-grained structure.

A more tractable problem is found in the interactions of proteins with each other, such as when one protein regulates (activates or inactivates) another protein through the process of phosphorylation [1], which involves a relatively simple reversible structural modification (a change in the fine-grained structure of another protein). The approach described in this paper could be applied to the modeling of networks of such interacting proteins.

Finally, our long-term vision is the exploration and understanding of the principles of self-organization in cell-based systems and how such principles can be integrated into an evolved software development process. We look forward to reporting on such progress in a future paper.

6. REFERENCES

[1] Becker, W., Reece, J., and Poenie, M. The World of the Cell, 3rd ed. Benjamin/Cummings, Menlo Park, CA, 1996.

[2] Cliff, D., Grand, S. The Creatures Global Digital Ecosystem. Artificial Life 5, 77-94, 1999.

[3] Cox, B. Object-Oriented Programming - An Evolutionary Approach, 2nd ed. Addison-Wesley, Reading, MA, 1986.

[4] Harel, D. Statecharts: A Visual Formalism for Complex Systems. Science of Computer Programming 8, 231-274, 1987.

[5] Harel, D. On Visual Formalisms. Communications of the ACM 31, 514-530, 1988.

[6] Harel, D. A Grand Challenge for Computing: Full Reactive Modeling of a Multi-Cellular Animal. In Workshop on Grand Challenges for Computing Research, Edinburgh, Scotland, November 2002. http://www.wisdom.weizmann.ac.il /~dharel/papers/GrandChallenge.doc

[7] Harel, D. A Grand Challenge: Full Reactive Modeling of a Multi-cellular Animal. In Proceedings of Hybrid Systems: Computation and Control: 6th International Workshop, HSCC 2003, Prague, Czech Republic, April 3-5, 2003. LNCS 2623, 2.

[8] IBM. Autonomic Computing: IBM's Perspective on the State of Information Technology, 2001. http://www.ibm.com/research/autonomic.

[9] I-Logix. I-Logix Rhapsody and Statemate, 2003. http://www.ilogix.com.

[10] Kahn, S., Makkena, R., McGeary, F., Decker, K., Gillis, W. and Schmidt, C. A Multi-agent system for the Quantitative Simulation of Biological Networks. In Proceedings of the 2nd Annual Conference on Autonomous Agents and Multi-agent Systems (AMMAS 03), Melbourne, Australia.

[11] Kam, N., Harel, D., Kugler, H., et al. Formal Modeling of C. elegans Development: A Scenario-Based Approach. In Proceedings of Computational Methods in Systems Biology: First International Workshop, CMSB 2003, Rovereto, Italy, February 24-26, 2003, LNCS 2602, 4-20.

[12] Kitano, H. Systems Biology: A Brief Overview. Science 295, 1662-1664, 2002.

[13] Maturana, H., and Varela, F. Autopoiesis and Cognition. D. Reidel, Dordrecht, Holland, 1980.

[14] McMullin, B., and Varela, F. Rediscovering Computational Autopoiesis. SFI Working Paper 97-02-012. Santa Fe Inst., Santa Fe, 1997. http://www.santafe.edu/sfi/publications/Working-Papers/97-02-012.pdf

[15] McMullin, B. Computational Autopoiesis: The Original Algorithm. SFI Working Paper 97-01-001. Santa Fe Inst., Santa Fe, 1997. http://www.santafe.edu/sfi/publications/Working-Papers/97-01-001.pdf

[16] McMullin, B. SCL: An Artificial Chemistry in Swarm. SFI Working Paper 97-01-002. Santa Fe Inst., Santa Fe, 1997. http://www.santafe.edu/sfi/publications/Working-Papers/97-01-002.pdf

[17] McMullin, B., and Gross, D. Towards the Implementation of Evolving Autopoietic Artificial Agents. http://www.eeng.dcu.ie/~alife/bmcm-ecal-2001/ bmcm-ecal-2001.pdf

[18] Mendes, P. GEPASI: a software package for modelling the dynamics, steady states and control of biochemical and other systems. Comput. Appl. Biosci. 9, 563-571, 1993.

[19] Pohjonen, R., and Kelly, S. Domain-Specific Modeling. Dr. Dobbs Journal (339), 26-35, August 2002.

[20] Rational Software. Rational Rose RealTime, 2003. http://www.rational.com/products/rosert.

[21] Rumbaugh, J., Jacobson, I., and Booch, G. The Unified Modeling Language Reference Manual. Addison-Wesley, Reading, 1999.

[22] Selic, B., Gullekson, G., and Ward, P. Real-time Object-Oriented Modeling. John Wiley & Sons, New York, 1994.

[23] Swarm Development Group. Swarm, 2003. www.swarm.org/index.html.

[24] Tomita, M. Whole-cell simulation: a grand challenge of the 21st century. Trends in Biotechnology 19, 205-210, 2001.

[25] Varela, F., Maturana, H., and Uribe, R. Autopoiesis: The Organization of Living Systems, its Characterization and a Model. BioSystems 5, 187-196, 1974.

[26] Webb, K., and White, T. UML as a Cell and Biochemistry Modeling Language. Carleton University Cognitive Science Technical Report 2003-05. http://www.carleton.ca/iis/TechReports