

**Autopoiesis**  
**A comparison of two computational models**

By Ken Webb

A research project submitted in fulfillment of the requirements for 07.498 as credit  
towards the degree of Bachelor of Arts with Honours

---

Professor Tony White, Project Supervisor

---

Second Reader

Institute of Interdisciplinary Studies  
Carleton University  
Ottawa, Canada

April 21, 2003

## Table of Contents

Abstract.....	2
Introduction.....	3
Motivations.....	4
Rational Rose RealTime.....	5
Original Cell Model (CM).....	8
Containment hierarchy and compartmental structure.....	10
Enzymes, transporters, and lipid bilayers.....	11
CM technical details.....	13
Autopoiesis.....	14
Autopoiesis - Varela model, and McMullin SCL model.....	15
Comparison between CM and SCL.....	18
Other relevant tools and concepts investigated in this project.....	19
Issues with Cell Model.....	21
Top-down vs. bottom-up.....	21
Cell model doesn't appear to do any useful work.....	22
Dealing with trillions of active entities.....	25
Adjacency vs. location.....	25
Interactions between active objects.....	26
History problem.....	27
Zombie cells.....	28
Not all conservation laws are adhered to.....	29
CM is not in a state of balance.....	30
Key details of actual biological entities are unknown.....	30
AutoCM - Integration of Cell Model with Autopoiesis and SCL.....	31
Lipids.....	31
SCL and AutoCM equivalents.....	40
Results - Is AutoCM an autopoietic system?.....	41
Future directions - Goal-directed uses of AutoCM.....	45
Conclusions.....	50
Bibliography.....	51

## Abstract

*This paper describes a biological cell simulation that incorporates the basic theory of autopoiesis, that replicates the Varela and McMullin SCL model of autopoiesis using a different technology, and that extends and scales that model in the direction of greater biological realism. The simulation was implemented using Rational Rose RealTime, a tool used for commercial development of concurrent real-time and embedded systems, rather than the cellular automata Swarm technology used by Varela and McMullin. Much of this paper deals with the resulting issues of object-oriented goal-directed top-down design versus the non-goal-directed bottom-up autopoietic organization of runtime. A suggestion for future work that could lead towards goal-directed use of an autopoietic organization is presented.*

## Introduction

Several years ago (Webb, 1999) I developed a simulation of a biological cell, using the Rose RealTime (RRT) software development tool, in an effort to simultaneously learn more about RRT, about biological systems, and about complex systems in general. One result of this effort was a set of new issues.

I have returned to that earlier system and the issues it evoked as the starting point for my honours project in cognitive science. I have chosen to integrate that work with the theory of autopoiesis proposed by neuroscientists Humberto Maturana and Francisco Varela (Maturana and Varela, 1980) in their attempt to characterize both life and cognition.

This honours project steers away from some of the more traditional approaches of cognitive science. Its perspective is much more that of artificial life (ALife) than of artificial intelligence (AI). It seeks to understand intelligence by understanding the biological processes that underlie the human variety of cognition. Its cognitive focus is on embodied intelligence rather than on the symbolic representational intelligence of traditional AI.

Much of this project is about biological cells. Neurons, the main focus of the neuroscience component of cognitive science, are one type of cell. I believe it is important to understand what neurons do in general, not just the part that we currently believe has to do with cognition. The networks of neurons in the nervous system, from which cognition is generally accepted to arise, are just collections of cells organized in a certain way. The present paper argues that cells, and thus neurons, are autopoietic system, and implicitly that autopoietic systems are an important approach to understanding cognition.

In this paper I will describe my original cell model (CM), including some of the biology behind it, and the RRT approach used in implementing it. I will then describe the theory of autopoiesis proposed by Maturana and Varela, a simple cell model demonstrating autopoiesis theory that was developed by Varela (Varela et al, 1974), and the modernized Substrate Catalyst Links (SCL)

version developed by Barry McMullin (McMullin & Varela, 1997; McMullin, 1997a; McMullin, 1997a; McMullin & Gross, 2001). Then I will discuss the issues that came out of my original CM system, in the context of autopoiesis theory and in comparison with Varela and McMullin's model. This will be followed by a discussion of my integrated Autopoietic CM (AutoCM) system developed as part of this honours project, the results of that work, and suggested future directions.

## **Motivations**

There was no single focus or rationale in developing the original cell model (CM). I was thinking about it abstractly in 1994, and it took on a concrete realization over a period of two years from 1998 to 2000. It was an informal personal project that evolved as my own interests cycled between several poles, including the following three:

- 1) I had a strong interest in complex systems, especially complex biological systems. I saw these as more complex than the digital switches I had worked on at Nortel. Also knowledge of biological cells is public, whereas digital switch technology is proprietary. I had a feeling that there were fundamental similarities between biological systems and today's most complex software and hardware systems, and I wanted to explore this common space.
- 2) As an employee of ObjecTime Ltd., I wanted to use the company's ObjecTime Developer (OTD) product (which later became Rational Rose RealTime) to build larger and more complex models to explore just what OTD (RRT) could do.
- 3) I wanted to explore what computer science can learn from a study of life and mind.

Cells, and the environment in which cells exist, are extremely complex entities, the understanding and simulation of which pose interesting challenges to the software architect. The ultimate challenge is to figure out how to integrate the solutions for the individual problems into a common architecture.

Not only are these systems exquisitely complex, but much of what is known about this complexity has been captured in well-written color-illustrated textbooks. These systems are more complex than and quite unlike current digital systems, and at the same time much better documented. In short, detailed information about these systems is readily available, in contrast to the lack of availability of detailed design specifications for proprietary technological systems.

Digital systems are often version 1.0, and are almost never more than version 10.x. Biological systems by contrast represent orders of magnitude more design cycles. These designs have been gradually honed over a billion years or so of evolution. They are complex, viable, robust, well-integrated with each other, and often surprising.

One challenge in the model was to integrate the diverse elements that became of interest as I investigated the realm of biology. This inevitably led to a search for unifying concepts on the one hand, and for ways of handling diversity on the other hand. Most computer models are developed to study one fairly well defined area, whereas it was the intention with CM to look more for architectural principles.

Another principle was to try to directly model the concrete reality described in biology texts, rather than using computer science abstractions such as artificial neural networks, genetic algorithms, cellular automata, swarm systems, and so forth. The approach with the original version of CM was qualitative rather than quantitative, informal rather than formal, and a personal learning experience rather than a focussed professional endeavor.

## **Rational Rose RealTime**

Rational Rose RealTime (RRT) is a design and implementation tool for the production of real-time and embedded software. It combines the features of the industry-standard Unified Modeling Language (UML) with the real-time specific features of the Real-time Object-Oriented Modeling (ROOM) approach developed at Bell Northern Research (Nortel) and ObjecTime Ltd. It is used

principally by companies that produce telecommunications and embedded applications. In both types of application the software is hidden away inside some piece of hardware and does not normally interact with a human through a monitor and keyboard. An RRT application's main function is to react to events in the environment, and internally-generated timeouts, in real-time.

Software developers design software with RRT by decomposing the system into an inheritance hierarchy of classes and a containment hierarchy of objects. Each object, or capsule as they are called in RRT, contains a state machine that defines how it reacts to externally-generated incoming messages (generated within other capsules or sent from external systems), and to internally-generated timeouts. All C++, C, or Java code in the system is executed during transitions from one state to another (which may be a self-transition to the same state). An executing RRT system is therefore an organized collection of communicating finite state machines. The RRT run-time scheduler guarantees concurrency by making sure that each transition runs all of its code to completion before any other message is processed.

The RRT design tool is visual. Capsules are dragged from a list of available classes into other classes. Compatible ports on different capsules are graphically connected to allow the sending of messages. State machines are drawn to represent the behavior of each capsule. Other useful graphical tools include use cases, and message sequence charts. External C++, C, or Java classes can be easily integrated into the system. The figure on the next page is a screen shot of RRT showing the list of capsules and classes, the finite state machine for CellBilayer, the hierarchical containment structure for EukaryoticCell, and a bit of C++ transition code.

Rational Rose RealTime - Cell9.tmdl

File Edit Parts View Browse Build Report Tools AddIns Window Help

### Capsules

- AutoCM
- Cell
  - EukaryoticCell
  - AnimalCell
  - PlantCell
  - Chromosome
  - CytoSkeleton
  - EukaryoticCell
  - State Diagram
  - Structure Diagram
  - Filament
  - IntermediateFilament
  - MicroFilament
  - MicroTubule
  - ProkaryoticCell
  - CellularSolution
  - Cytoplasm
  - Cytosol
  - ExoCytoplasmPassageway
  - GaseousSolution
  - InternalSpace
  - Matrixsol
  - MitochondrialIntermembrane
  - Nucleosol
  - PermeableToSmallMolecules
  - Solution
  - SpacePassageway
- CellNucleus
- Gaia
- Hormone
- Lipid
- Membrane
- CellBilayer

### State Diagram: Membrane / CellBilayer - Top State

Finite State Machine

```

stateDiagram-v2
    [*] --> initialize
    initialize --> Waiting4Either : smLevelExt
    Waiting4Either --> Waiting4Ext : smLevelInt
    Waiting4Ext --> Active : smLevelExt
    Waiting4Ext --> Waiting4Int : smLevelInt
    Waiting4Int --> Active : smLevelExt
    Active --> Waiting4Ext : smLevelExt
    Active --> Waiting4Int : smLevelExt
    
```

### Containment Hierarchy

```

classDiagram
    class EukaryoticCell {
        + /permExt
    }
    class Cytoplasm {
        + /nm3p
        + /nm3
        + /transChan
        + /erSp
    }
    class Nucleus {
        + /permIntS
        + /permS
    }
    EukaryoticCell --> Cytoplasm
    EukaryoticCell --> Nucleus
    
```

### Transition Specification for config

Code: **Transition Code**

```

configData = RECEIVE_PTR(Config);
fitnessData = configData->pFitness;
pOther = configData->pOtherConfig;
Phene *phene = &(configData->pPhene);
for (int i = 0; i < SmEnumSizeE;
    pSm->molecule[i].set(phene->sm);
// Now it's safe to incarnate the
frame.incarnate(lipidDisintegrat;
configID.ConfigSig(configData).s
    
```

The developer generates the executing system by selecting "Build" from a menu. RRT generates all required code from the diagrams, and produces an executable. The executable can then be run, tested, and debugged using the design diagrams as run-time monitors and probes.

I decided to use RRT to develop the Cell Model (CM) because of my several years of experience with the product, and because it allows the static hierarchical structure of biological systems to be very naturally represented. About 99% of the capsule classes in CM, and an even higher percentage of the run-time objects, directly represent identifiable biological entities such as Cell, Membrane, Enzyme, Nucleus, DNA, mRNA, etc.

### **Original Cell Model (CM)**

The original cell model (CM) contains much more than just a single cell. It includes various environments to test the cell. There is a circulatory system to move a population of cells continuously between an environment (a lung) rich in oxygen and low in carbon dioxide, and a second environment (a brain) low in oxygen and high in carbon dioxide. There are neurons which are specialized extensions of a cell, and a nervous system to test the transmission of chemicals from one cell to another. There is a universe with a sun as a source of energy, plants to convert solar energy into chemical energy (glucose), a digestive system to distribute this energy to cells, and various other components and subsystems. See the appendix for more details.

The single eukaryotic cell, the type of cell found in plants and animals, is the central entity in CM. A CM cell consists of a membrane, cytoplasm, and a nucleus. The membrane contains a lipid bilayer and transport proteins both of which actively control the movement of passive chemicals between the inside and outside of the cell. The cytoplasm contains numerous types of small molecules such as water and glucose, enzymes that actively convert small molecules from one form into another, and various specialized compartments such as mitochondria. The nucleus contains chromosomes with DNA. The figure on the next page compares the RRT structural view of a eukaryotic cell with a more pictorial 3D view.





## **Containment hierarchy and compartmental structure**

Natural systems are often thought of as containment hierarchies. System N contains many entities at level N - 1, and is in turn contained within an entity at level N + 1. These containment hierarchies can be many tens of levels deep.

Biological cells exhibit a strong form of containment that the biological literature refers to as compartmentalization or compartmental structure. Cells, as well as the organelles and many of the other entities that make up cells, consist of a solution enclosed in a semi-permeable membrane. The membrane provides a definite boundary, while allowing specific molecules to move in and out.

The **compartmentalization** of activities within the cell is one solution to the problem of concentration. ... Plant and animal cells have a variety of **organelles**, internal compartments that are delineated by membranes and are highly specialized for specific functions. ... This internal compartmentalization of specific functions makes it possible for large cells of plants and animals to maintain locally high concentrations of the specific enzymes and compounds involved in particular cellular processes. (Becker, 1996, p. 81-82)

Organisms also tend to have definite boundaries. Humans and other vertebrates have skin, invertebrates have exoskeletons, and trees have bark.

Organs and other biological entities intermediate between cells and organisms are often less definite. This is especially true of parts of the nervous system.

... the notion of **compartmentalization** ... is contrasted with the view that the brain operates as a holistic organ ... There is a compromise view, which seems consistent with the best knowledge in the field. It holds that some mental attributes are localizable to specific regions, or constellations of regions, within the brain. ... However, many functions ... are divided into subfunctions, which are distributed throughout the brain. (Solso, 2001, p. 45-46)

Non-membrane bounded components of cells, that is large and small molecules, are held together by electromagnetic forces. Molecules, especially DNA, RNA, proteins, and lipids, have their own containment hierarchies.

Rose RealTime (RRT) naturally supports containment hierarchies, one of the fundamental concepts of the object oriented paradigm. Containment hierarchies are developed top down, starting with a single top-level system entity.

One of the first steps in the construction of CM was to capture obvious important entity types as RRT capsules and classes, and then arrange these into a containment hierarchy. The resulting model structure corresponds very closely with the structure described in the biological literature (Becker, 1996; Purves, 1997). However, the above two quotes suggest why it was relatively straight forward implementing a cell using the compartmentalization concept, but quite difficult using this concept to implement the aggregate of neural cells that make up the nervous system.

### ***Enzymes, transporters, and lipid bilayers***

Enzymes, transport proteins and lipid bilayers are the active objects in CM. They interact with passive small molecules. Figure 1 shows conceptually how this works at runtime.

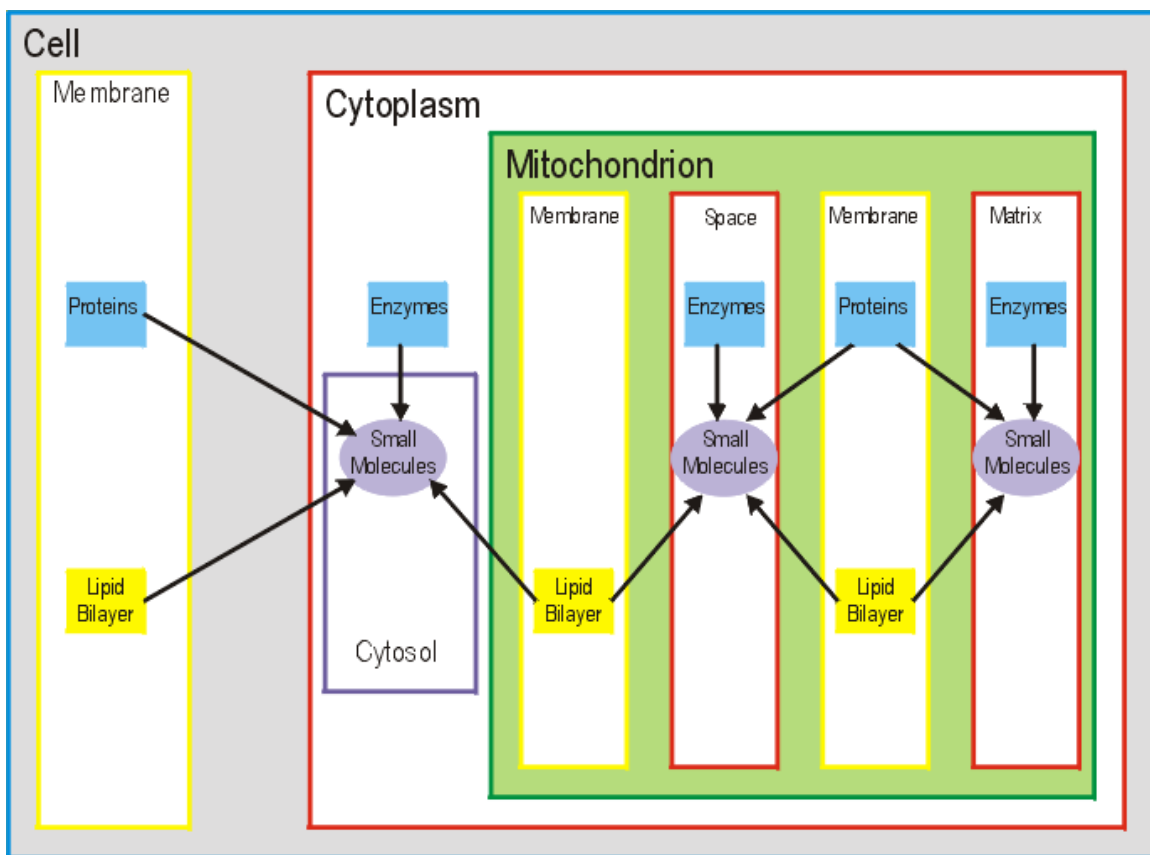
An initialization and configuration stage will have already exchanged RRT messages between adjacent objects, to set up a network of pointers or references from active objects to the small molecules they will be operating on. These references represent the direct contact that molecules have with each other at the molecular level of a cell, and are shown as arrows on Figure 1.

Various enzymes in the Cytoplasm act on small molecules in the Cytosol, resulting in a simple connected metabolism. The metabolism gradually converts glucose into pyruvate and also into lipid molecules. At the same time, the lipid bilayer and (transport) proteins in the Cell Membrane transport small molecules in both directions between the Cytoplasm and the external environment (not shown in Figure 1). One effect of this is to replenish the supply of glucose from quantities that may be present in the environment.

Also at the same time, Mitochondrion moves small molecules between the Cytoplasm and its own interior, in a two step operation. The Mitochondrion is an organelle completely contained within

the Cytoplasm. It contains a double membrane. The lipid bilayer and transport proteins within each Mitochondrion Membrane move specific small molecules back and forth. Enzymes within the membrane-bounded spaces (Space, Matrix) act on the small molecules. A major function of the Mitochondrion is implemented by gradually moving pyruvate into the Mitochondrion where it is broken down to produce ATP. The ATP then moves back to the Cytoplasm where it provides usable energy for many cellular processes.

Each active entity in the system only operates on what is locally available to it. The global metabolism emerges out of these local operations between adjacent entities.



**Figure 1** - CM structural containment, with references (arrows) from active objects (enzymes, transport proteins, lipid bilayers) to small molecule data structures.

### **CM technical details**

The next few paragraphs describe several more technical aspects of CM. They are included here because later sections of this paper refer to these technical details.

The rate at which each active object in CM acts, is specified by kinetic rate parameters (Becker, 1996, p.148; Mendes, 1993; Mendes, 1997; Mendes, 1998). The  $K_m$  and  $V_{max}$  rate parameters for an enzyme determine its average and maximum rate of reaction with small molecules. Lipid bilayers and other active objects in the system each have a single rate constant. The quantity of each small molecule is kept in a small molecule data structure within each entity such as Cytosol and LipidBilayer that contains small molecules.

The rate parameters, small molecule quantities, and other parameters are stored in the DNA Database, an offline relational (SQL) database. Visual Basic scripts write the parameters to a set of configuration files which are read by the CM system each time it starts up. Changes, additions and deletions to CM involving active objects and small molecules are made by editing the DNA Database and running a script.

The following is the C++ code for the simplest type of enzyme in CM, and is typical of (although simpler than) the code for all other enzyme types. This set of statements would be executed every timestep for any irreversible enzyme that converts one substrate into one product, and that is not associated with any activators, inhibitors or coenzymes. The programming variables `gene` and `sm` point to `gene` (enzyme) and small molecule values that have been read in from configuration files. `gene->substrateK` and `gene->substrateV` are the  $K_m$  and  $V_{max}$  rate parameters for the enzyme.

```
// Irreversible, 1 Substrate, 1 Product, 0 Activator, 0 Inhibitor, 0 Coenzyme
case Irr_Sb1_Pr1_Ac0_In0_Co0:
    s = sm->molecule[gene->substrateId[0]].get();
    nTimes = enzymeLevel * ((gene->substrateV * s) / (gene->substrateK + s));
    sm->molecule[gene->substrateId[0]].dec( nTimes );
    sm->molecule[gene->productId[0]].inc( nTimes );
    break;
```

## Autopoiesis

Neuroscientist Humberto Maturana, and later his student Francisco Varela, wanted to define and characterize both life and cognition. In the course of their analysis they came up with a theory that they have called *autopoiesis*. The term *autopoiesis* means "self-making". They "claim that the notion of autopoiesis is necessary and sufficient to characterize the organization of living systems" (Maturana & Varela, 1980, p. 82). Entities in an autopoietic system participate in the creation of other entities, and hence participate in creation of the autopoietic system itself in part by creating a boundary around it. Everything is created from within, except for the basic building blocks of matter and energy.

Maturana and Varela (1980, p. 78-9) provide the following oft-quoted rather dense definition.

An autopoietic machine is a machine organized (defined as a unity) as a network of production (transformation and destruction) of components that produces the components which:

- (i) through their interactions and transformations continuously regenerate and realize the network of processes (relations) that produced them; and
- (ii) constitute it (the machine) as a concrete entity in the space in which they (the components) exist by specifying the topological domain of its realization as such a network.

Fritjof Capra (1996, p. 162) does a good job of summarizing Maturana and Varela's point (i).

Autopoiesis, or "self-making," is a network pattern in which the function of each component is to participate in the production or transformation of other components in the network. In this way the network continually makes itself. It is produced by its components and in turn produces those components.

Barry McMullin (1999, p. 3) summarizes point (ii) as follows.

the system must ... establish some sort of boundary between "itself" and the rest of the universe in which it is embedded.

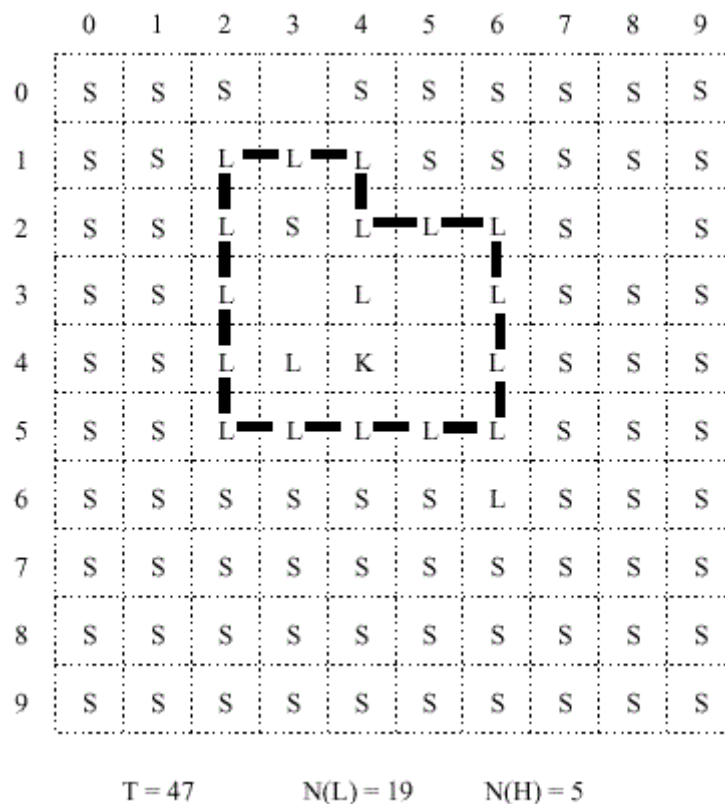
The biological cell has been taken by Maturana, Varela, and many other authors as the canonical exemplar (standard concrete example) of the autopoiesis theory. Maturana and Varela (1980, p. 90) state that "the cell is a molecular embodiment of autopoiesis". They discuss cellular processes

at a fairly abstract level at numerous points in their writings. Mingers (1995) uses the cell as his basic example. Capra (1996) refers extensively to cells in his extensive discussions of autopoiesis and self-making in general.

## Autopoiesis - Varela model, and McMullin SCL model

Varela produced a model of a cell as "a simple embodiment of the autopoietic organization"

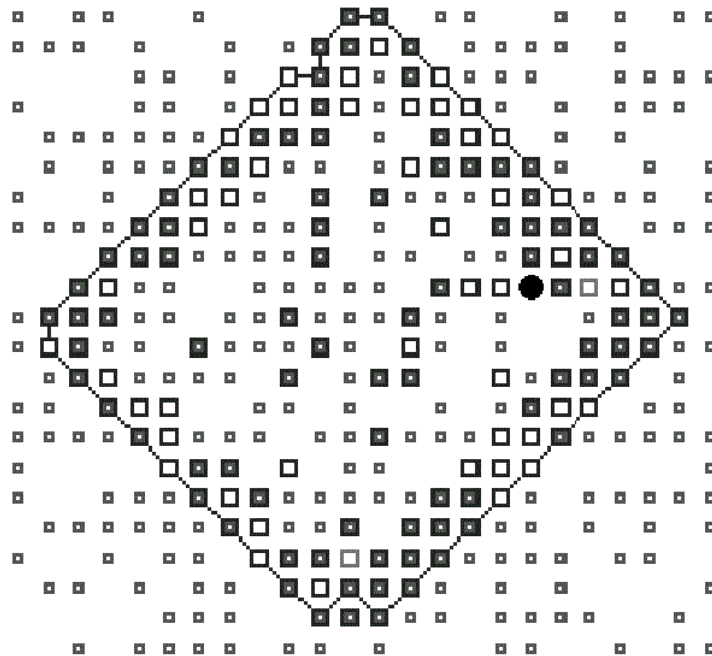
(Varela, Maturana & Uribe, 1974, p. 189). This model, implemented in Fortran, and as shown in Figure 2, contained three types of entities existing and randomly moving in a two-dimensional grid. A single catalyst converts a small population of substrates into links. The links are capable of bonding, and may form a continuous membrane around the internal catalysts and substrates.



**Figure 2** - Result of one run of the Varela, Maturana, & Uribe (1997) model. At timestep 47 a single catalyst (K) with substrate (S) and link (L) entities, is enclosed by a completed membrane consisting of bonded links. (McMullin, 1997a, p.24)

Barry McMullin has more recently produced updated versions of the Varela, Maturana and Uribe model, which he calls SCL and SCL-GRO. SCL (McMullin & Varela, 1997; McMullin, 1997a; McMullin, 1997a), standing for Substrate Catalyst Link, is a re-implementation of the 1974 model using Swarm and Objective C. SCL-GRO (McMullin & Gross, 2001) extended this to address various issues related to the take-up and maintenance of links in the membrane.

In addition to substrate, link, and catalyst entities, SCL also distributes holes throughout the grid. The role of these holes seems to be to provide space into which new entities can be created and to which existing entities can move. They fill otherwise vacant space. In a biological cell, water makes up about 75-85% of the cell (Becker, 1996, p.22), and has the role of filling otherwise vacant space. In subsequent sections, abstract holes will be thought of as representing water.



**Figure 3** - This is a screenshot from the [SCL] experiment with growth turned off: after 3130 time-steps the autopoietic agent is still intact. The small squares are S particles; the large and large dark squares are L and L + particles (without and with absorbed substrates respectively); the filled circle is the catalyst. Three distinct layers of the agent are clearly visible. (McMullin, 2001).



Figure 3 is a screenshot of SCL in action. At this point in the simulation, a continuous membrane of links has formed. There are three functional layers surrounded by an exterior that only contains passive substrate (S) particles. The innermost layer consists of a single catalyst, substrates, free links, and holes. The middle layer consists of links and substrates. These links are ready to become part of the membrane. The membrane layer consists of bonded links, and substrates on their way between the inside and outside of the cell.

The biological realization of the SCL catalyst contains just as much if not more constituent structure as does the emergent membrane. The membrane consists of individual links bonded together. Each link is created through the action of the catalyst on a substrate. The action of the catalyst eventually results in the creation and maintenance of the membrane. This membrane in turn should facilitate the creation of amino acid particles and the construction of additional catalysts from these. This reciprocal activity is not captured in the SCL model.

The purpose of a membrane is to separate two compartments, one on the interior of the membrane and another on the exterior. SCL-GRO does make a functional distinction between inside and outside, as is obvious from McMullin's screenshot (2001, p.4). The difference is by virtue of a single catalyst trapped inside the membrane.

It is possible that the catalyst substrate reaction would not take place without a sufficient concentration of chemicals, where this concentration is only achieved once the membrane is fully in place. SCL cannot be called a model of autopoiesis until the lipid bilayer completely surrounds some space. Until that time the internal kinetic reaction does not yet depend on the concentrating function of the membrane.

## Comparison between CM and SCL

Table 1 compares CM and SCL-GRO on a number of features.

Webb original CM	McMullin SCL-GRO
top-down	bottom-up
no autopoiesis (allopoietic)	simple autopoiesis; complete cycle
substrate molecules are statistical/passive	substrate molecules are individual/active
enzymes are individual/active	catalysts are individual/active
no lipids; only lipid bilayer	lipids are individual/active
active enzymes act on passive substrate	direct interaction of catalyst and substrate
detailed model; numerous entity types	simple model; few entity types
only adjacency is known	adjacency and location are known
graphics are external	graphics are built-in/integrated
many compartments	one compartment becomes two
models steady mature state of eukaryote	models creation of a primitive cell
can't easily have protein-protein interactions	could have protein-protein interactions
spatial quantities are not consistent	true spatial quantities
terminology: enzyme substrate molecule lipid water	terminology: catalyst substrate link hole

**Table 1** - Comparison of CM and SCL-GRO.

Table 1 is largely self-explanatory except for the reference to *spatial quantity*. A model is consistent in its spatial quantities if its volume and surface area are consistent with each other. CM allows a greater number of entities within a cell than would fit within the membrane space that could be created by the number of lipids in the membrane.

## **Other relevant tools and concepts investigated in this project**

As part of this honours project I briefly investigated a variety of other theories and techniques that might provide insights and practical assistance. The most pertinent of these are briefly described below, in alphabetical order.

### **Autocatalysis**

Stuart Kauffman has extensively explored the role of autocatalytic metabolic pathways in cells. He is especially interested in how these may have come about during the period when life was just getting started on the earth. (Kauffman, 1993; Kauffman, 1995)

### **Blackboard systems**

Blackboard systems make use of some global workspace that all entities within the system have access to. Entities do not pass messages directly. Instead entity A updates something in the blackboard that may later be used by entity B.

### **Chemical abstract machine (CHAM)**

Berry and Boudol have developed an artificial chemistry called Chemical Abstract Machine (CHAM). The abstract machine is "based on the chemical metaphor used in the Gamma language of Banatre and Le Métayer. States of a machine are chemical solutions where floating molecules can interact according to reaction rules. Solutions can be stratified by encapsulating subsolutions within membranes that force reactions to occur locally." (Berry & Boudol, 1992; Boudol, G., 2002).

### **Emergence**

Emergence is a general concept that will be used in this paper, but that is hard to find a definition for. John Holland (1998, p.225) says that "Emergence occurs in systems that are generated. The systems are composed of copies of a relatively small number of components that obey simple laws."

## **Gepasi**

Gepasi simulates chemical and biochemical kinetics on a computer. A simulation can have multiple compartments each containing multiple enzymes and substrates. It uses differential equations to perform the same type of calculations made in CM. I have used it to confirm the quantitative results of CM. (Mendes, 1993; Mendes, 1997; Mendes, 1998).

## **Luisi Minimal Cell Project**

Pier Luisi and his lab in the Chemistry Department at ETH-Zürich have been investigating questions related to the origin of life. In the Minimal Cell Project, they are trying to produce the simplest biological entity that can be considered alive, the "minimal necessary structural prerequisites for a semi-synthetic, functional cell". Luisi has published a paper with Varela, and has extended the idea of autopoiesis into chemical minimal autopoietic systems. Luisi's lab is interested in the spontaneous formation of self-organized structures, compartmentalization, and other issues that relate to CM. What is especially fascinating is that these are not computer models, but are actual chemical systems. (Luisi, various).

## **Shapiro artificial chemistry**

Ehud Shapiro at the Weizmann Institute of Science in Israel is doing research in two areas of potential interest. He is investigating the "construction of nanoscale programmable computing machines made of biomolecules", and the "development of a computer language for describing and simulating molecular processes and pathways in the cell while utilizing tools originally designed for the description of computer processes". (Shapiro website)

## **Swarm software package**

Swarm is a software package developed at the Santa Fe Institute and currently maintained by swarm.org. It allows the modeling of emergent systems using swarms of local entities.

Applications are developed using either Objective-C or Java. (Swarm website). McMullin wrote his SCL system using Swarm in Objective-C.

## **Issues with Cell Model**

This section contains a brief description of various questions, issues and problems that arose out of the work on the original Cell Model (CM). There is considerable overlap amongst some of these issues, but it will prove useful in later discussion to separate out different aspects of particular issues. We will return to each issue later in the paper to discuss it in terms of insights from autopoiesis theory and from the Varela McMullin computer model of autopoiesis.

### ***Top-down vs. bottom-up***

CM is top-down and bottom-up at the same time. Its top-down design architecture consists of a multi-level containment hierarchy created using RRT capsules. But, after initial configuration, it's run-time architecture is largely bottom-up. Perhaps the best way to characterize it is as bottom-up with top-down constraints. These constraints represent the billion plus years of evolutionary history that life has undergone, plus the laws of chemistry and physics that force things along certain pathways.

Real biological cells and organisms are largely bottom-up. Whatever organizational structure they have is created from the interactions of low-level entities. But every cell is created with the same constraints of history, chemistry, and physics mentioned above, in that it is created by dividing another cell in half. Every cell alive today is the result of a very large number of such divisions.

The problem with CM is that its bottom-up potential is not being fulfilled. The metabolic, genomic, and other bottom-up processes currently implemented are not sufficient to maintain the cell in a healthy state, they do not allow new structure to develop over time, and they do not allow an unhealthy cell to die. In short, there is no way for the top-down structure to change as a result of bottom-level activity.

An autopoietic system is inherently bottom-up. By definition it is autonomous. It must be free to determine its own boundary and details of its own structure. In a purely top-down model these are imposed externally and cannot be determined by the system itself.

To make the original version of CM autopoietic, the existing bottom-up processes must be enhanced and completed so that they can effect the initially imposed top-down structure. The existence at run-time of top-down entities such as LipidBilayer will need to depend on the quantity of molecules produced by other entities, such as the production of lipids by enzymes. If an entity such as LipidBilayer does not contain enough of the right causal elements, then it will cease to exist in practical terms at run-time. The definition of existence changes from what is imposed at design-time to what plays a practical (causal) role at run-time.

### ***Cell model doesn't appear to do any useful work***

CM attempts to implement a simulation of actual cellular structure and activity, as described in a cell biology textbook (Becker, 1996). But the end result is a system that doesn't appear to do any useful work in human terms. The model just seems to spin its wheels. It should be possible to somehow harness this activity for human goal-directed purposes. At present, CM is a piece of "wild software". One challenge is to figure out how to domesticate it.

Maturana and Varela make it clear that this lack of purpose is only what should be expected of a real or simulated biological system. "Living systems, as physical autopoietic machines, are purposeless systems" (p. 86). "A living system is not a goal-directed system" (p. 50). I see these statements as giving permission to continue to implement AutoCM as a system that does no useful work from the perspective of a human observer. The issue disappears when it is realized that this is how AutoCM should be.

When we talk of purpose, we are talking from the perspective of an external observer. Human, or possibly other, observers of autopoietic systems are an integral part of the Maturana and Varela theory. "*The observer is a living system and an understanding of cognition as a biological phenomenon must account for the observer and his role in it*" (Maturana & Varela, 1980, p. 9, original italicized). They want to distinguish our own human-centred descriptions and understandings of biological systems, from the way that those systems actually are. This is a

distinction between epistemology (what we can know), and ontology (what there actually is out there in the physical universe).

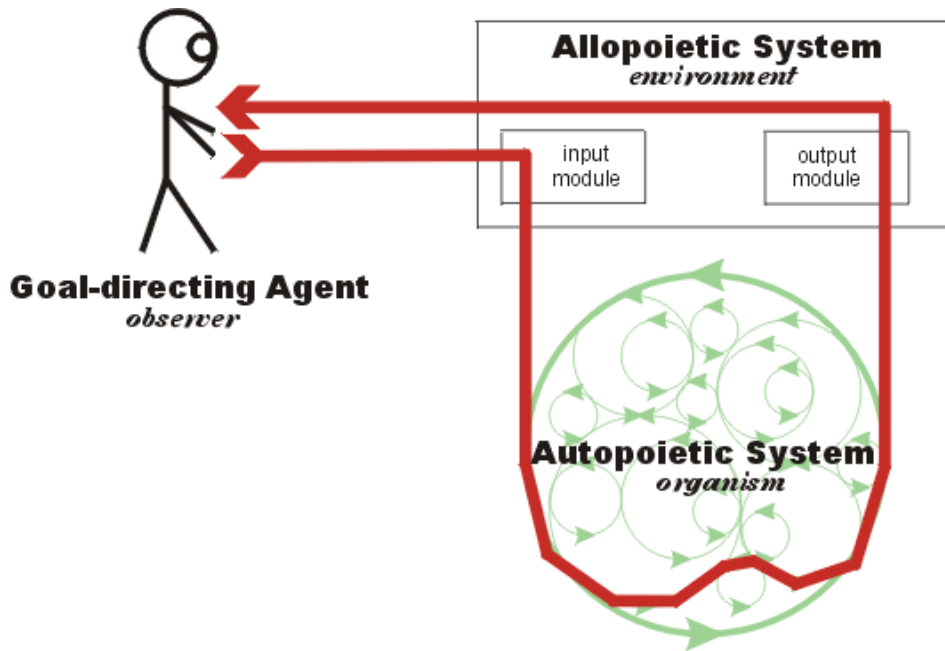
Maturana and Varela distinguish autopoietic systems from allopoietic systems. A car, an example of an allopoietic system (Maturana & Varela, 1980, p. 79), is a man-made machine whose components are produced by processes in a factory outside the car. Unlike an autopoietic machine such as a cell, the internals of a car cannot maintain the car's essential organizational unity. When the engine obtains energy by burning gasoline, it moves the car and the people inside it, but it does not participate in keeping the outer boundary (steel, plastic, paint, etc.) of the car in good order. By contrast, the cell's metabolic processes do participate in maintenance of the cell's outer lipid bilayer boundary.

Table 2 summarizes Maturana and Varela's (1980, p. 80-81) description of the characteristic differences between autopoietic and allopoietic systems.

<b>Autopoietic Systems</b>	<b>Allopoietic Systems</b>
maintain their own organization	produce something different from themselves
specify their own boundaries	have boundaries defined by the observer
do not have inputs and outputs	typically have inputs and outputs
are purposeless	have human goal-directed purpose
operate bottom-up	typically operate top-down
examples: cell, nervous system	examples: car, crystal

**Table 2** - Properties of autopoietic and allopoietic systems.

Figure 4 is a conceptual view of how autopoietic and allopoietic processes interact with each other. The allopoietic system is pictured using straight lines, with inputs and outputs, and as processes that have a start and stop. The autopoietic system is shown with curved lines, and as a set of interlocked cycles that have no beginning and no end. The allopoietic process slightly perturbs the autopoietic system. The figure also uses Maturana and Varela's (1980, p.8) terms "observer", "organism", and "environment".



**Figure 4** - Allopoietic and Autopoietic systems.

In a later section of this paper I will discuss some possible ways of applying these Maturana and Varela ideas on autopoietic and allopoietic processes, to allow AutoCM to perform some "useful work". These ideas could be implemented in a future version of the AutoCM software.

This lack of purpose makes CM difficult to test. There are few if any clear inputs and outputs associated with the system. This makes standard object-oriented testing, such as by generating use cases, difficult or impossible.

Maturana and Varela (1980) suggest that this is to be expected of a cell or cell simulation. "In terms of their functional organization living systems do not have inputs and outputs ... and it is only in our *descriptions*, when we include them as parts of larger systems which we define, that we can say that they do." By contrast, in "man-made (allo-referring) systems, ... input and output functions are all important through the purposeful design of their role in the larger systems in which they are included" (p. 51).



These statements do not eliminate the need to test CM, but they do at least explain why it is so difficult to test. And they suggest that it has to be tested by somehow making it a part of a larger more traditional allopoietic test harness.

### ***Dealing with trillions of active entities***

CM is able to handle trillions of entities by treating many things statistically. In the real world every molecule is an active concurrent object that can influence other molecules that it comes into contact with. In CM most of this active influence is lost by representing most entities as passive, statistical data. Only a few entities such as enzymes and other proteins, and lipid bilayers are individually active. Given the space and processing limitations of digital computers, it makes sense for CM to have taken this general approach.

Autopoiesis theory deals with the cell in an abstract way. It therefore has no need to deal with the issue of large numbers of concrete entities. SCL represents this abstraction using a very small (in the hundreds) number of entities, all of which actively interact with their neighbors.

### ***Adjacency vs. location***

Entities in CM that find themselves adjacent to each other are able to interact. Adjacency is implemented in one of two ways in CM. RRT capsules that can send messages to each other either directly or through a series of relay ports are considered to be adjacent. Adjacency is also implemented through the use of pointers. Active objects in CM act on small molecules through pointers. Many different active objects can simultaneously point to and act on the same collection of small molecules. The combination of message passing and pointers constructs an adjacency network at run-time. This network is strictly local in that no entity has global knowledge of any part of the network except its own local neighborhood.

I define location here as a global property, as x, y, and z coordinates within some global three dimensional space. Entities in CM do not have and do not need spatial coordinates, and therefore do not have location.

However, if we try to visualize the executing model as we would expect to see it through some advanced type of microscope, we would require a precise location for each entity. This is an issue for an external observer and not for the local entities within the system itself.

In his discussion of autopoiesis, Mingers (1995, p.38) reiterates what is already apparent.

The actual processes that occur in a living organism depend only on the immediate neighborhood interactions and reactions of the components involved and do not in any causal sense depend on a reference to, or representation of, or any supposed functions of, the system as a whole.

He implies that the adjacency vs. location issue is a natural outcome of the cell's being an autopoietic system. Adjacency has to do with the immediate neighborhood (local), while location (actual x,y,z coordinates) is a global concern. Autopoiesis theory also suggests that adjacency is in the domain of the components of the autopoietic system, while location is in the domain of the observer who has a global perspective.

All entities in the SCL model have both local adjacency and global location. Objects on the grid know which objects they are adjacent to. An observer can click on any object to determine its x,y location in space.

### ***Interactions between active objects***

All proteins in CM, such as enzymes and transport proteins, are active objects. At present they are only able to interact with passive data objects and not directly with other active objects. However, much of the processing that goes on in a real cell involves interactions between proteins, such as during DNA to mRNA transcription, during mRNA to protein translation, and during subsequent regulation of proteins.

SCL deals with the lipid bilayer as a collection of individual lipid molecules. AutoCM has been modified accordingly. LipidBilayer now contains lipid small molecules that are operated on by various active entities in the system. The quantity of lipids has a causal role in the activity of LipidBilayer.

The same type of thing could be done with proteins (and thus enzymes) and the amino acids that make up proteins. Enzymes could act on other proteins by acting on the amino acids that constitute them.

This suggests a generalization. All active objects in the system are composed of collections of small molecules. The active objects are polymers, while the small molecules are monomers that repeat. Becker (1996, p.30) states that there are three major monomer/polymer systems in a cell.

Table 3 adds Lipids/Lipid Bilayers as a fourth type.

Polymer	Repeating Monomer	Number of kinds of repeating units
Protein; Enzyme; Polypeptide	Amino acids	20
Nucleic Acids (DNA, RNA)	Nucleotides	4 in DNA; 4 in RNA
Lipid Bilayers	Lipids	various
Polysaccharides (Starch, etc.)	Monosaccharides (sugars)	One or a few

**Table 3** - Four types of small molecule.

As a general approach, active objects can have an influence on other active objects in AutoCM by having an effect on their constituent monomers. This would be an important enhancement in a future version of AutoCM.

### ***History problem***

To construct a cell of the complexity found in biology requires an ancestor cell. That earlier cell requires its own ancestor, which requires its own ancestor, and so on. Every actual biological cell does have an ancestor cell that specifies its basic organization, the quantities of each type of molecule within it, and its genome. These characteristics are passed down through the generations

by the conceptually straight forward process of cell division, in which one cell becomes two almost identical daughter cells.

CM is much simpler than a real biological cell, but it also needs to have its initial contents and organization specified. It is not clear what is the best way to do this. At present, a cell's initial conditions are specified rather arbitrarily. The correct values are simply not known.

There are several types of frozen history designed into CM. The structural history of real cells is captured in the RRT capsule hierarchy. Genetic history is captured in the DNA and the resulting function of each enzyme. The quantitative history is captured in the number of actual entities of each type at run-time initialization.

As with AutoCM, SCL also has to start its simulation at some arbitrary point in history. This is typically an arbitrary arrangement of substrate, catalysts, and lipids on the grid.

Maturana and Varela (1980) make a few points regarding the origin of autopoietic systems.

The establishment of an autopoietic system cannot be a gradual process; either a system is an autopoietic system or it is not. .. there are not and there cannot be intermediate systems. ... given the factibility of autopoietic systems, and given the existence of terrestrial autopoietic systems, there are natural conditions under which these may be spontaneously generated. (p.94)

The implication is that for a real system (although not necessarily for a simulation), there must be some point in time in its evolution or development at which it suddenly transitions to a state of being autopoietic.

### ***Zombie cells***

Non-viable cells in CM never die. They just keep on going. There is no concept of health in CM.

In the real world, cells of many types do end up dying, often in a planned way. Dead cells are broken up and their parts reused.

SCL is like CM. It also allows extremely dysfunctional cells to continue to exist. Both systems have the goal of creating a functional system, but must first go through a phase in which any system is allowed to continue on in hopes that it may eventually become functional. This is simply an artifact of trying to do in a computer what nature has had a billion years to do.

***Not all conservation laws are adhered to***

In many kinetic interactions in CM quantities of atoms are conserved. But this is not the case universally in the model. If all conservation laws were adhered to then it would be possible to determine if the system is performing correctly (no memory leaks, no bugs of specific types) by keeping track of the counts of atoms and other fundamental entities.

A model of an autopoietic system should have conservation laws that are adhered to. This is crucial for causality. It should not be possible to create or destroy matter or energy within the system. Only transformations of one thing into another should be allowed.

SCL follows this principle, but only at the level of molecules. Catalysts (enzymes) convert substrate molecules directly into lipids, without regard for the number of atoms contained within each type of molecule. This is a necessary simplification.

AutoCM also takes some liberties in the interests of simplification. But if it is to be used to do useful work, then it should adhere strictly to a set of conservation laws. These represent part of the physics of the system. They allow the system to evolve, using the constraints imposed by scarce resources (resources that can neither be created nor destroyed), and they make it possible to check for bugs in the software by checking to see if the expected amounts are correctly conserved. Conservation laws make the system more dependent on the environment (which encourages embodiment), and are one form of interaction between entities in the system.

### ***CM is not in a state of balance***

Actual cells participate in many different cyclical activities all of which work together concurrently to produce a stable configuration. Stability is maintained over time. This state of balance has been called homeostasis (Cannon, 1939), and is an important part of autopoiesis theory (Maturana & Varela, 1980, p.78).

CM implements somewhat arbitrary (but interesting) bits of cell functionality. No attempt has been made to connect all the functionality together. For example, many enzymes in CM produce small molecule substrates that are not consumed by any other enzyme or active object, and will therefore eventually accumulate to dangerous levels.

SCL implements one cyclical activity, the substrate-lipid cycle. This cycle is complete and can become in a state of balance. Lipids can be formed from substrate and can decay back to substrate. Lipids can be linked together into a lipid bilayer and can subsequently become unlinked. AutoCM has been enhanced by completing the substrate-lipid cyclical activity as implemented within SCL.

### ***Key details of actual biological entities are unknown***

When CM starts up it needs to define the quantities of the many small molecules and other entities within the model. It also needs to define the precise function of each active object. These specific quantities and precise functions are unknown. Biology is not yet at the point where these can be defined with the precision required of an executing model. Therefore, it is not currently possible for CM to start up in a stable configuration. It lacks the stability observed in a real cell. Each protein can be given discrete known functions to perform, but in fact each protein also has an indefinite number of other unknown interactions. This imprecision suggests that it is not a realistic enterprise to have CM model what actually goes on in a cell. Instead it should attempt to model some of the types of things that go in a cell, which requires a more abstract understanding of issues such as "what is life" and "what are the essential functions performed by a cell". The

best approach would be to have it evolve a stable organization from some arbitrary starting point. At present it doesn't have the ability to evolve its organization.

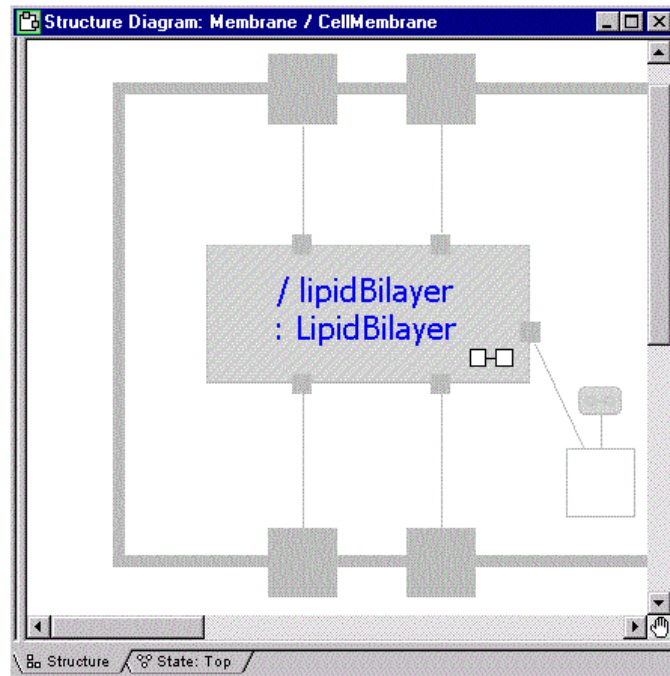
Autopoiesis theory and the SCL model deal with cells at a very abstract level. They are only concerned with those aspects of cells, or of other systems, that have to do with the definition of autopoiesis. Much of what is found in CM and AutoCM, including the issue of actual quantities of things, is irrelevant to autopoiesis theory and SCL.

## **AutoCM - Integration of Cell Model with Autopoiesis and SCL**

This section describes the autopoietic version of CM. AutoCM is an extension of CM that demonstrates the basic theory of autopoiesis, and implements the functionality found in SCL and SCL-GRO (McMullin & Varela, 1997; McMullin, 1997a; McMullin, 1997a; McMullin & Gross, 2001). To avoid confusion between the terminologies used in the two models, SCL catalyst, substrate, and link are given the equivalent CM names - enzyme, substrate molecule, and lipid. In addition, SCL holes are equated with CM water molecules.

### ***Lipids***

Lipids are the major constituent of lipid bilayers. They play the role of semi-permeable membrane in biological cells. They are of primary concern in the modeling of an autopoietic system. Figure 5 shows the structure diagram for CellMembrane as it appears in RRT.



**Figure 5** - CellMembrane contains a LipidBilayer, which in a real cell contains billions of lipid molecules. The lines running outward from LipidBilayer represent the connections with the inside and outside of the cell.

### Genesis of new lipids

CM does not generate lipids as part of its enzyme-catalyzed metabolism. Instead it uses the simplification of directly implementing an emergent structure called LipidBilayer whose responsibility is to actively control (or restrict or regulate) the movement of molecules between the interior and exterior of the space that it surrounds. In SCL, enzymes do convert substrate molecules into lipids, an essential element of any model of an autopoietic cell.

AutoCM generates new lipid molecules by introducing additional enzyme types. These enzymes act on various lipid values in the small molecule data structure of the cytosol. Lipids are represented statistically in the same way as other small molecules. Changes required at a technical level to generate lipids in AutoCM are:

- additions to the DNA database,
- additions to the small molecule data structure, and



- additions to the kinetic rate parameters.

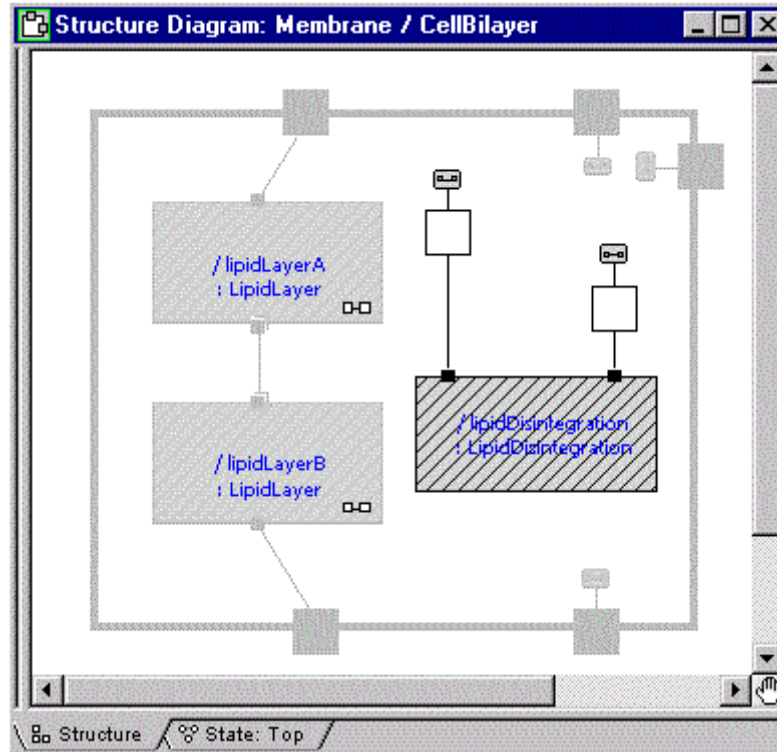
Technical details on the DNA database, the small molecule data structure, and kinetic rate parameters are included in the CM technical details section on page 13.

### Decay of existing lipids

CM does not have a capability to spontaneously, without the aid of an enzymatic catalyst, cause the decay of small molecules. SCL does have such a capability. It uses a decay parameter to specify a probability that a lipid will decay back to a single substrate particle.

It is important for AutoCM to have this decay capability. In real biological cells there is a constant decay of components at all levels. One of the requirements of an autopoietic system is that it outlive the lifetimes of its constituent components. Thus, it is essential that lipids can be created and destroyed while the lipid bilayer retains the same organizational structure.

AutoCM handles decay by introducing a decay active object (the LipidDisintegration capsule in Figure 6) that is correlated with time. During each timestep there is some finite probability that any given small molecule will degenerate into some degraded form. This is implemented by changing the values in the appropriate small molecule data structure (see CM technical details on page 13). The two lines with shapes at the end that project above the LipidDisintegration capsule in Figure 6 are message connections (RRT ports) that are used to get access to the small molecule data structures within the CellBilayer.



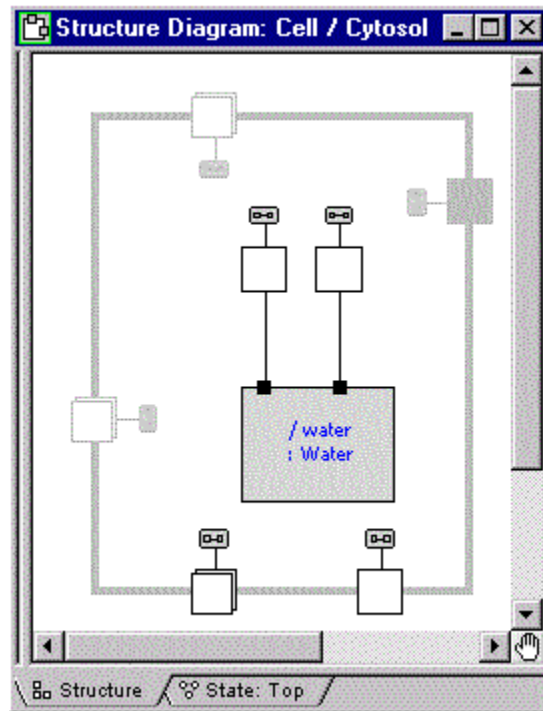
**Figure 6** - The new LipidDisintegration capsule in CellBilayer. Lipid Disintegration has access to the small lipid molecules within CellBilayer.

### Movement of lipids and retention in the vicinity of a lipid bilayer

Once free lipids exist within the cytosol, some mechanism is required to transport them to the vicinity of a lipid bilayer, and then to make sure that they remain in that vicinity until they can be incorporated into the bilayer.

Because CM is statistical and does not model the locations of entities, it does not implement movement directly. SCL does directly implement the random movement of individual enzymes, substrate molecules, and lipids within the cytosol. This random movement eventually brings newly created lipids into the vicinity of a lipid bilayer.

SCL-GRO implements an affinity feature to ensure that "if a link is in the neighborhood of a chain it may drift along the chain but will not move away from it" (McMullin, 2001, p. 2). Thus, as shown previously in Figure 3, lipids will stay at the edge of the cytosol once they are adjacent to an existing lipid bilayer.



**Figure 7** - The new Water capsule within Cytosol. Water has access to the small lipid molecules within Cytosol.

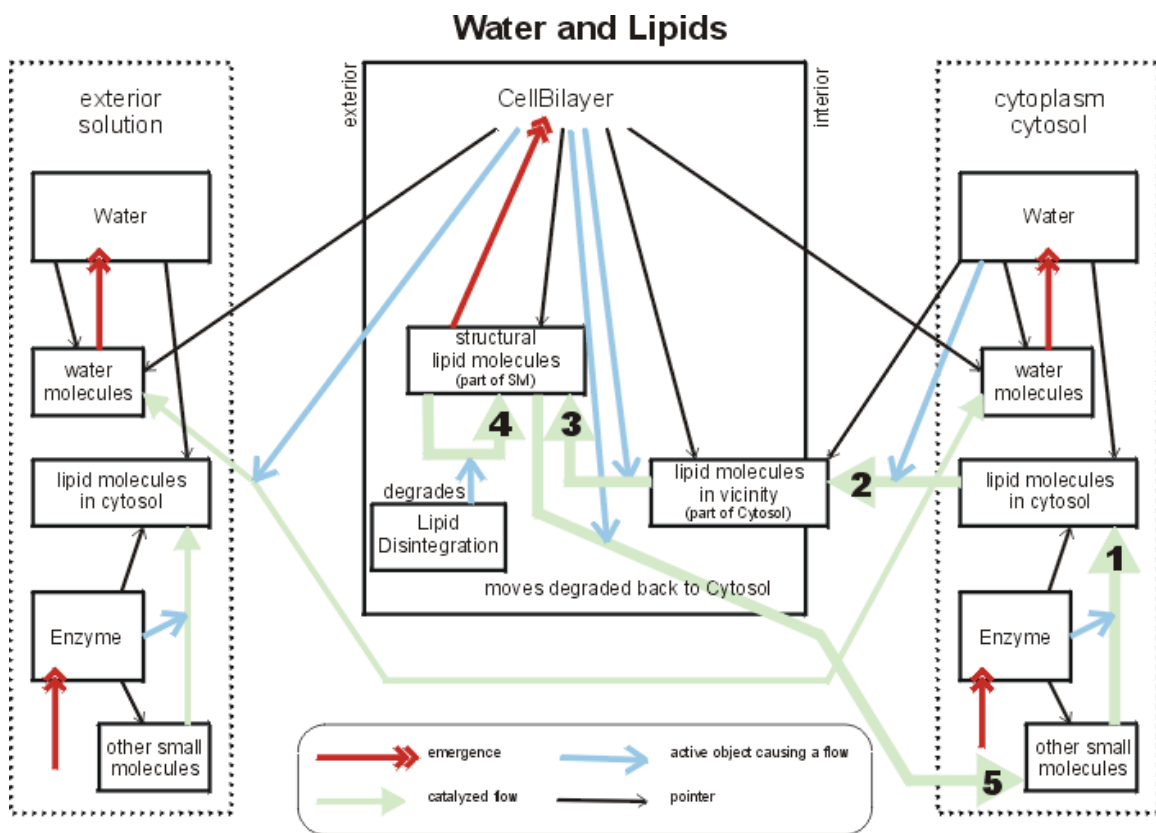
AutoCM implements a process equivalent to SCL-GRO affinity. It makes use of water to exert the required force on the lipid molecules. The inspiration for this approach comes from the section on the importance of water in Becker et al. (1996, p. 22-240). Hydrophobic molecules such as lipids

... disrupt the hydrogen-bonding structure of water and, for this reason, tend to be excluded by the water molecules. Hydrophobic molecules therefore tend to coalesce in an aqueous medium, associating with one another rather than with the water. This association is driven not so much by any specific affinity of the hydrophobic molecules for one another but by the strong tendency of water molecules to form hydrogen bonds and to exclude molecules that disrupt hydrogen bonding. ... such associations of hydrophobic molecules (or parts of molecules) are a major driving force in the folding of molecules, the assembly of cellular structures, and the organization of membranes. (Becker et al., 1996, p. 24)

AutoCM assumes random statistical movement of lipid molecules until they reach the vicinity of the lipid bilayer. At that point the cumulative effect of all the water within the cytosol is to keep

the lipids there until they can be incorporated into the lipid bilayer. This is implemented through a new active object called Water. During each time cycle Water moves some proportion of lipids from the small molecule data structure within the cytosol to the small molecule data structure with LipidBilayer. The quantity moved is proportional to the number of lipid molecules within Cytosol and the number of water molecules within Water.

Figure 8 is a conceptualization of most of the activity that takes place in AutoCM to implement the functionality found in SCL.



**Figure 8 - Water and Lipids.**

In Figure 8, CellBilayer is an active entity that derives its causal strength from the quantities of lipid molecules contained within it. If this quantity should reach 0, then the lipid bilayer would cease to exist in any functional sense. In this degenerate state, it would continue to function as a

kind of ghost entity in AutoCM with a role of very rapidly equalizing the quantities of all small molecules between what was its interior and its exterior. In its normal state, CellBilayer has the important role of transporting water and other small molecules bidirectionally between the interior and exterior compartments, and of transporting lipid molecules from its vicinity into itself. At this point the lipids are automatically part of CellBilayer. No additional link step is required as in SCL. This is because of the statistical nature of AutoCM.

In the figure, Water is an active entity that functions causally only in proportion to the quantity of water molecules in the cytosol. The role of Water is to transport lipid molecules from the cytosol to the vicinity of CellBilayer. If there are no lipids in the cytosol, note that it will benignly not produce any result, an important fact given that, as we will see in a subsequent section, this mode of unregulated lipid transport is uncommon in real cells. But it is necessary to make use of this mechanism in the much simplified world of AutoCM.

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

Figure 8 also shows Enzyme active objects, some of which catalyze the conversion of precursor substrate molecules into various types of lipid molecules within the cytosol. Enzymes are composed of particular configurations of amino acid constituents. As with Water and LipidBilayer, Enzymes derive their causal powers (their ability to have an effect on other entities) from the causal potential and quantity of their components.

The large numbers 1, 2, 3, 4, 5 in Figure 8 represent a cyclical sequence of steps that is found in both SCL and AutoCM. This is the cycle of lipid production, disintegration, bonding, and bond decay as shown in Table 6.

Figure 8 can be extended to the left or to the right. The Water entity shown to the left of LipidBilayer might be part of a larger membrane-bounded structure. In a multi-compartment model, the Cytoplasm and CellBilayer shown in the figure might be those of a mitochondrion organelle, while the exterior Water and lipid entities might be part of the overall cell. Thus, the figure shows part of a potentially recursive structure.

AutoCM Figure 8 shows all the entities that appear on McMullin's (2001, p.4) figure 1 (reproduced as Figure 3 in this paper). He identifies "three functionally distinct parts or layers" (p.3) which are also evident in AutoCM Figure 8. These correspondences are summarized in Table 4.

SCL Layer	AutoCM Entity	SCL Entity
<b>1. Reactor (inner)</b>	other small molecules	substrate
	lipid molecules in cytosol	link
	enzyme	catalyst
	water molecules	hole
<b>2. Free Links (middle)</b>	lipid molecules in vicinity	link in vicinity
<b>3. Membrane (outer)</b>	CellBilayer; structural lipid molecules	chain; bonded links

**Table 4** - Three functionally distinct layers in SCL and AutoCM.

One major difference between SCL and AutoCM is which entities are chosen to be causally active agents. SCL chooses substrate molecule, catalyst, and lipid (link), while AutoCM chooses Water, Enzyme, and CellBilayer. Enzyme and catalyst are the same, while Water and LipidBilayer emerge out of the collective actions of substrate molecules or holes, and of lipids.

This choice is related to the SCL focus on location of individual entities, while AutoCM deals with substrate, hole and lipid only statistically.

### Lipid transport mechanisms in real eukaryotic cells

In real cells, lipids are transported in several different ways. In the most common approach, lipids are directly created into the endoplasmic reticulum (ER) lipid bilayer close to the nucleus.

Vesicles consisting of lipids subsequently bud off the ER and are actively transported to various destinations including the cell bilayer. This is a very regulated process. The unregulated method implemented in SCL and AutoCM is by far the least common. It may be potentially dangerous for a cell to have a lot of individual lipid molecules floating around inside.

### Spherical shape of real lipid bilayers

AutoCM does not concern itself with the very interesting and important issue of how the spherical shape of the lipid bilayer comes into being. CM and AutoCM only assume that the lipid bilayer acts as a semi-permeable barrier between two compartments and makes no statement about the possible shape of that barrier.

SCL, on the other hand, must make some statement about the shape of this barrier because it directly and visually implements a location for every entity in the system. The shape that emerges in SCL-GRO is rectangular, approximating that of a diamond. McMullin (2001, p. 3) suggests that "this is apparently partly due to the affinity between free links and the membrane and the consequent inhibition of its inward motion, and partly a reflection of the underlying lattice geometry".

AutoCM uses water to exert an outward pressure on lipids adjacent to the lipid bilayer. This could be interpreted as an argument that water causes the lipid bilayer to have an outwardly curved shape. In fact, no such claim is made and AutoCM takes no position on the complex issue of what the shape must be.

## SCL and AutoCM equivalents

This section establishes that SCL and AutoCM are conceptually equivalent. They both deal with the same types of chemical species, the same types of chemical reactions, and produce similar structures. Where they differ is in the implementation.

SCL "involves three distinct chemical species" (McMullin, 1997, p.3), as shown in Table 5.

SCL	AutoCM
Substrate: S	substrate molecules within small molecule data structure (i.e. PhosphatidicAcid and Ethanolamine)
Catalyst: K	various enzymes (catalysts)
Link: L	various lipids (e.g. PhosphatidylEthanolamine)

**Table 5** - Chemical Species in SCL and AutoCM.

SCL "supports six distinct reactions" (McMullin, 1997, p.3), as shown in Table 6.

SCL	AutoCM
(1) Production $K + 2S \rightarrow K + L$	PhosphatidicAcidPhosphatase + (PhosphatidicAcid + Ethanolamine) $\rightarrow$ PhosphatidicAcidPhosphatase + PhosphatidylEthanolamine
(2) Disintegration $L \rightarrow 2S$	done by LipidDisintegrator capsule within CellBilayer PhosphatidylEthanolamine $\rightarrow$ (PhosphatidicAcid + Ethanolamine)
(3) Bonding	done by CellBilayer which moves PE from InVicinity within the Cytosol to PE within CellBilayer free lipids become part of the existing bonded lipid structure
(4) Bond Decay	done by LipidDisintegration, followed immediately by CellBilayer moving the disintegrated PhosphatidicAcid and Ethanolamine back to the Cytosol
(5) Absorption $L + S \rightarrow L^+$	These last two steps happen together, as CellBilayer moves substrate molecules bidirectionally between its interior and exterior, without explicitly having them spend any time inside CellBilayer.
(6) Emission $L^+ \rightarrow L + S$	

**Table 6** - The six reaction types and the equivalents in AutoCM.

SCL-GRO (McMullin, 2001) includes four additional features that were not in SCL, as shown in Table 7.



SCL	AutoCM
Affinity	Water provides this, as a pressure that keeps lipids in the vicinity of CellBilayer
Smart Repair	not required because AutoCM is not implemented using a lattice that can have holes
Improved mobility of links	links don't move in AutoCM because lipid bilayers are initially completely formed
Displace motion	this is part of what happens in AutoCM bonding

**Table 7** - SCL-GRO additional features and the equivalents in AutoCM.

"In order to qualify as an interesting instance of autopoiesis, its [the autopoietic agent or organization] longevity has to exceed that of its components (the links) which are designed to be unstable" (McMullin, 2001). In AutoCM, the lipid bilayer lasts indefinitely, as long as there are lipids inside it. Individual lipid molecules are continuously being added to the bilayer, spontaneously disintegrate back to substrates, and are subsequently removed from the bilayer. Thus, in AutoCM, the longevity of the bilayer is greater than the longevity of the lipids.

## Results - Is AutoCM an autopoietic system?

Varela (1974, p. 192) has presented a six-point key to determine whether or not a given system is autopoietic. It is difficult for me to objectively assess whether AutoCM is an autopoietic system using these criteria, because all six points are qualitative and subjective, and because of my own bias. This difficulty is compounded by the unclear definitions that Maturana and Varela provide for such key terms as *unity*, *interaction*, *constitutive elements*, etc., and the overall obscure style of their writing. Mingers (1995) provides some helpful discussion of these points and includes an analysis of a cell.

1. Determine, through interactions, if the unity has identifiable boundaries. If the boundaries can be determined, proceed to 2. If not, the entity is indescribable and we can say nothing.

By *unity* is meant "that which is distinguished by an observer ... [and] a whole distinguished from a background" (Mingers, 1995, p.13). Mingers (p.17) summarizes this first point as "specifying

that there is an identifiable entity with a clear boundary". In AutoCM, EukaryoticCell is an identifiable entity (RRT capsule) in which CellMembrane (and its constituents CellBilayer and lipid small molecules) actively plays the role of boundary by regulating what can and cannot move between the inside and outside of the cell, and by regulating the rates of these movements.

2. Determine if there are constitutive elements of the unity, that is, components of the unity. If these components can be described, proceed to 3. If not, the unity is an unanalyzable whole and therefore not an autopoietic system.

The required nature of these components is unclear. Maturana and Varela (1980, p.90-91) only mention "molecules (proteins, lipids, carbohydrates and nucleic acids)" in their discussion of constitutive relations, those relations "that determine the topology of the autopoietic organization, and hence its physical boundaries". These are the same four types of small molecules that are identified and discussed in the section *Interactions between active objects* and Table 3 on page 26 of this paper. SCL only has this type of component (catalysts and substrates). Mingers (p.18) on the other hand, referencing Zeleny and Hufford (1992), only mentions larger organelle-sized "components such as the mitochondria, the nucleus, and the membranous network known as the endoplasmic reticulum". In AutoCM both small molecules (including enzymes which are a type of protein, and lipids) and organelles (mitochondria) appear to be components according to the second of the six criteria.

3. Determine if the unity is a mechanistic system, that is, the component properties are capable of satisfying certain relations that determine in the unity the interactions and transformations of these components. If this is the case, proceed to 4. If not, the unity is not an autopoietic system.

According to Mingers (p.17), this means that the system "operates mechanistically, i.e., its operation is determined by the properties and relations of its components". No "special organizing forces" (Maturana & Varela, p.74) need be invoked. In the terminology used earlier in this paper, the system should be completely bottom-up, and should be free of top-down goal-directed constructs. AutoCM includes both a design- configuration-time top-down structure, and a run-time bottom-up structure. It can be argued that at run-time AutoCM is exclusively bottom-up

because the only system activity involves the actions of active objects such as enzymes. However, one of the active objects is the cell bilayer, a top-down entity. However, activity of the cell bilayer depends causally on the quantity of lipids, which is determined bottom-up.

4. Determine if the components that constitute the boundaries of the unity constitute these boundaries through preferential neighborhood relations and interactions between themselves as determined by their properties in the space of their interactions. If this is not the case, you do not have an autopoietic unity because you are determining its boundaries, not the unity itself. If 4 is the case, however, proceed to 5.

Mingers has little to say that can add to an understanding of this point. In AutoCM, the constituent lipids interact implicitly (they can't act explicitly because this is a statistical model) with each other to produce the actions of the lipid bilayer. The lipid bilayer interacts only with its immediate neighbors inside and outside the cell. The unity itself, and not the external observer or programmer (the *you* of point 4), produces the functionality of the lipid bilayer.

5. Determine if the components of the boundaries of the unity are produced by the interactions of the components of the unity, either by transformation or previously produced components, or by transformations and/or coupling of non-component elements that enter the unity through its boundaries. If not, you do not have an autopoietic unity. If yes, proceed to 6.

At run-time in AutoCM, the enzymes inside the cell create the lipids that later become part of and determine the extent of activity of the cell bilayer. Many of the components inside the cell (such as water and glucose) get there by passing through the cell bilayer from the outside.

6. If all the other components of the unity are also produced by the interactions of its components as in 5, and if those which are not produced by the interactions of other components participate as necessary permanent constitutive components in the production of other components, *you have an autopoietic unity in the space in which its components exist*. If this is not the case and there are components in the unity not produced by components of the unity as in 5, or if there are components of the unity which do not participate in the production of other components, you do not have an autopoietic unity.

All components of AutoCM participate in the production of other components, and all components are produced through the actions of other components. The exception is those components that are created at configuration-time when the system first starts up.

Mingers (1995, p. 16-20), with references to Zeleny and Hufford (1992), uses these six points to argue that a cell is indeed an autopoietic system. AutoCM is a simulation of a cell. If properly designed and implemented, and I believe that it is, then AutoCM is an autopoietic system.

McMullin (1999, p. 6, 10) compares the concepts of autocatalysis (Kauffman, 1993) and autopoiesis, and suggests a heuristic for determining if a system is autopoietic or autocatalytic. McMullin suggests that the two concepts are very similar, the distinction being that autopoiesis requires that the system boundary be created from within the system rather than allowing it to be imposed from the outside. His heuristic is to *mix* together two instances of the same system, which I interpret as meaning place them in intimate contact with each other. "The key question now is whether there will still meaningfully be two instances of the reaction network or just one. That is, in the absence of any imposed spatial separation mechanism, do the networks themselves maintain their individuality? I would suggest that, if they do, then this reaction network may reasonably be classified as meeting the autopoietic criterion 'for specifying the topological domain of its realization'; whereas if not, then the network should be regarded only as collectively autocatalytic". McMullin (p.10) presents some very tentative conclusions, and says that "On balance then, I would say that the self-sustaining agents in the SCL system do not pass my suggested heuristic test for 'full' autopoiesis. This must be considered at least a little controversial given that the model was designed explicitly to illustrate autopoietic organization (albeit in minimal form)". His discussion emphasizes how difficult it is to confirm if a system, even one as simple as SCL, is or is not autopoietic, largely I believe because of the informal and obscure way in which Maturana and Varela have presented the concept of autopoiesis.

In the case of AutoCM, when two or more cells are placed within the same external chemical solution and are allowed to indirectly influence each other, the multiple cells do maintain their own individuality over time. AutoCM would seem to pass the McMullin heuristic. This

difference between SCL and AutoCM comes about largely because of the way the two systems are implemented.

However, if in AutoCM a sufficient number of large cells were placed in the same solution as a single small cell, then eventually the small cell would lose all of its material and would merge with the solution and thus with the larger cells. It would be unable to maintain its individuality. It would also be interesting in AutoCM to see what would happen if two cells were placed in intimate contact with each other, such that each lipid bilayer had direct access to the lipids of the other cell. In this case there would probably be symmetrical movement of lipids between the two, and their individuality would be preserved.

Real cells maintain their individuality when in intimate contact with each other in a variety of ways. Almost all cells that are linked to neighboring cells "have some sort of structure exterior to the plasma [cell] membrane" (Becker, p.271). Lipid molecules migrate at a much faster rate (measured in fractions of a second vs. more than a week) laterally within a lipid layer than they do between two layers in a lipid bilayer (Becker, p.190). Real cells pass the McMullin heuristic.

### **Future directions - Goal-directed uses of AutoCM**

The present paper has already discussed Maturana and Varela's distinction between autopoietic and allopoietic systems. Let us assume that AutoCM is an autopoietic system or at least a simulation of an autopoietic system. Can we now create and attach a separate allopoietic system to AutoCM, such that AutoCM will perform useful computational work for the allopoietic system, and will still maintain its autopoietic organization? One important reason for posing this question is as a means of exploring autopoiesis, the main topic of the present paper, more deeply.

For present purposes, let us define "useful computational work" as the ability to perform single and multi operator arithmetic calculations, and return a correct result. Operations will be restricted to addition and multiplication, represented symbolically in the allopoietic domain as the

operators + and \*. There must be at least one and no more than two such operators in a requested calculation. Operands will be restricted to the positive integers between 1 and 5. Nesting of operations is allowed, but only to one level.

Table 8 shows several computations that AutoCM will be expected to perform, as well as the expected correct answers.

Input	Output
(2+5)	7
(2*5)	10
((2+3)*4)	20
(2+(3*4))	14

**Table 8** - Test inputs and outputs.

The above descriptive terms such as "useful", "calculations", "operator", "operand", "input", "output", etc. are all from the perspective of a human goal-oriented agent and an associated allopoietic system. A calculator can directly perform these calculations, as can a fairly simple traditional computer program written in a wide variety of computer languages. But it is not obvious how an autopoietic cell could be made to do this type of work.

These results suggest that it is plausible to suggest that real cellular processes could be manipulated so as to produce the types of simple calculations indicated above. AutoCM purports to be a simulation of a real cell. It is of course missing most of the complexity of a real cell, but it does have the autopoietic organization which Maturana and Varela hold is the hallmark of life.

I would like to propose one plausible mechanism by which a more advanced version of AutoCM could be made to do these simple arithmetic calculations. It is not important whether or not this particular mechanism should prove capable or not of doing the calculations. What is important is that there are potential mechanisms by which the normal, built-in, innate processes of an autopoietic system can be brought to bear to produce humanly useful work without destroying the autopoietic organization.

The proposed mechanism would start by having the human give the allopoietic system (AlloSystem) a simple arithmetic problem in standard symbolic form, such as  $(2+5)$ . AlloSystem would convert this to a virus, which is a natural allopoietic machine. The symbols  $(2+5)$  would be encoded in RNA as three viral genes. The RNA would be packaged inside a virus protein coat, and would then be injected into the autopoietic cell.

The three RNA strands (or possibly a single concatenated strand) would find their way to the ribosomes that line the external wall of the rough endoplasmic reticulum (rough ER) where they would be treated in the same way as any other RNA. The ribosome would automatically translate the RNA into three separate polypeptide (protein) strings. It would push these through the membrane to the interior (lumen) of the ER where the polypeptides would fold into proteins and might be subjected to various post-translational regulatory modifications. They would then be packaged as cargo into one or several vesicles.

The vesicles would bud off the external face of the ER, would be transported to the cis face of the golgi apparatus (GA). This is the first step in the cell's normal secretory pathway by which proteins and lipids are sorted and selectively transported outward from the nuclear region to other parts of the cell or to the exterior of the cell. The "golgi apparatus (GA) functions as a major biosynthetic organelle involved in the packaging and post-translational modification of newly synthesized proteins" and "serves as the cell's main sorting and distribution station for protein and vesicle traffic" (Farquhar & Hauri, 1997, p. 65). In our case, we want to make use of the pathway that leads to the exterior.

The arithmetic calculation would be performed by the operator protein (+) acting on the two operand proteins (2, 5). It is important that this only happen at the right time, while all three are together within the same vesicle. The ER, vesicle, and golgi compartments naturally serve to constrain, sort, separate, and generally carefully regulate cellular processes. Cargo is tagged to indicate its destination, and may be modified to allow it safe passage. The tagging is similar in

concept to the addition of headers and trailers, and various other modifications performed on data in telecommunication systems. Possibly the operator protein could be made inactive using normal cellular processes until the right time.

The + operator protein, once activated, would take whatever two substrate operand proteins it was packaged with, and would combine these into a single protein. In the current example, it would combine the two-unit-long protein with the five-unit-long protein to produce a protein that is seven units long. Possibly the operator protein would then be again deactivated or tagged for later destruction.

Eventually the secretory pathway would transport a vesicle containing the resulting seven-unit-long protein to the cell's outer membrane. The vesicle would merge with the membrane and spill its contents outside, a process called exocytosis.

At this point the allopoietic process would gain access to the resulting protein, and could read off the answer by counting its length. In an experiment with a real cell the operand proteins could possibly be tagged in some way so that the result would glow fluorescently or radioactively. The result, in the form of the symbol 7 would then be reported to the human who posed the original question.

The various compartments in this pathway, especially vesicles and golgi cisternae, are a key ingredient. They provide the sort of boundary that autopoiesis implies is necessary to allow specific processes to proceed. The boundary concentrates the entities required in the reaction, and excludes other entities that might interfere. The availability of temporary vesicles could allow for more complex computations than just  $(2+5)$ . They could for example help to distinguish  $((2+3)*4)$  from  $(2+(3*4))$ . Membrane bounded vesicles and golgi cisternae should possibly be thought of as a cellular analog of symbolic parentheses (). Every pair of parentheses together with its operator implies the need for a separate vesicle. Recursion is probably not possible, at least with real cells, because there is no autopoietic process inside a vesicle to produce membranes. A



vesicle lacks sufficient internal complexity. However, recursive processes can generally be transformed into iterative processes.

Every part of the process, from initial virus to ribosome to ER to vesicle to golgi to another vesicle to exocytosis would be just business-as-usual as far as the cell is concerned. The allopoietic request simply becomes part of the normal cyclical activity of the cell. The cycles just pull the "input" along, rather than being pushed into doing anything as would be the case in a traditional computer system.

A small amount of allopoietic input would only perturb the cell's structure slightly, but would leave intact its autopoietic organization. A real virus in a real cell might go well beyond this level of perturbation, by hijacking much of the cell's machinery for the purpose of reproducing a large number of copies of the virus. Eventually the cell outer membrane would burst, the new viruses would emerge, and the cell would die. "In a living system loss of autopoiesis is disintegration as a unity and loss of identity, that is, death" (Maturana & Varela, 1980, p. 112).

Unlike a traditional allopoietic calculation machine, in the autopoietic system there is no distinction between the hardware or software that the machine consists of and the programs and data that these operate on. The autopoietic system is fully embodied. Every computation or calculation it performs makes a real structural difference to it, and has the potential to disrupt its fundamental architectural organization, possibly to destroy it. Computation has consequences for an autopoietic system.

These potential consequences have a lot to do with the resilience/robustness/redundancy of living systems. A cell is constantly on the edge of the type of ordered computation performed by allopoietic machines and the forces of disorder that strive to tear it apart. This balancing act is what Chris Langton and others in the ALife community call the "edge of chaos". A cell must be resilient or it will not survive. It gets its resilience by virtue of its being an autopoietic system.

The resilience of autopoietic systems, coupled with the possibility that autopoietic processes can be harnessed or hijacked by allopoietic systems to perform useful work, suggests that autopoiesis may be a potent architectural principle to use in building complex computer systems.

## **Conclusions**

In this project I have integrated the basic ideas of autopoiesis theory into a pre-existing model of a biological cell. Through this integration I have been able to address many of the issues that arose during that earlier work. The resulting AutoCM system is an autopoietic system, able to maintain its own internal organization and able to maintain the boundary that keeps it separate from its environment. There is much work that remains to be done - to determine how autopoietic and allopoietic systems can work together, and in the context of cognitive science, to understand how our human cognitive allopoietic goal-oriented natures can be explained in terms of the autopoietic cells out of which we are constructed.

## Bibliography

- Adams, D. (1982). *The Hitchhiker's Guide to the Galaxy: Life the Universe and Everything*. Pan Books Ltd.
- Bachmann, P.; Luisi, P.; & Lang, J. (1992). Autocatalytic self-replicating micelles as models for prebiotic structures. *Nature*, 357, 57-59.
- Bachmann, P.; Walde, P.; Luisi, P.; & Lang, J. (1990). Self-Replicating Reverse Micelles and Chemical Autopoiesis. *J. Am. Chem. Soc.*, 112, 8200-8201.
- Becker, W., et al. (1996). *The World of the Cell, 3rd ed.* Menlo Park, CA: Benjamin/Cummings.
- Berry, G., & Boudol, G. (1992). The Chemical Abstract Machine. *Theoretical Computer Science*, 96, 217-248.
- Boden, M. (2000). Autopoiesis and Life. *Cognitive Science Quarterly*, 1, 117-145.
- Boudol, G. (2002). Abstract of Berry & Boudol, 1992. Retrieved November 10, 2002 from [www-sop.inria.fr/mimosa/personnel/Gerard.Boudol/tcs96.html](http://www-sop.inria.fr/mimosa/personnel/Gerard.Boudol/tcs96.html)
- Brooks, R. (1991). Intelligence without Representation. In Haugeland, J. (ed.). *Mind Design II*. Cambridge, MA: MIT Press.
- Cameron, W. (2001). Autopoiesis, Agency and Accident: Criteria for the Attribution of Life. *Systems Research and Behavioral Science*, 18, 447-461.
- Cannon, W. (1939). *The Wisdom of the Body, revised ed.* New York: Norton.
- Carlson, N. (2001). *Physiology of Behavior*. Boston: Allyn and Bacon.
- Capra, F. (1996). *The Web of Life*. New York: Anchor Books.
- Chopineau, J; Lesieur, S; Carion-Taravella, B; & Ollivon, M. (1998). Self-evolving microstructured systems upon enzymatic catalysis. *Biochimie*, 80, 421-436.
- Farquhar, M.; & Hauri, H.-P. (1997). Protein sorting and vesicular traffic in the Golgi apparatus. In Berger, E. and Roth J. (eds.). *The Golgi Apparatus*. Basel: Birkhauser Verlag.
- Fontana, W.; & Buss, L. (1996). The Barrier of Objects: From Dynamical Systems to Bounded Organizations. In J. Casti, J. and Karlqvist, A. (eds.). *Boundaries and Barriers*. Addison-Wesley. Retrieved August 22, 2002 from [www.iiasa.ac.at/Publications/Documents/WP-96-027.pdf](http://www.iiasa.ac.at/Publications/Documents/WP-96-027.pdf)
- Fontana, W., & Buss, L. (2002). website on AlChemY. [www.santafe.edu/~walter/AlChemY/alchemY.html](http://www.santafe.edu/~walter/AlChemY/alchemY.html)
- Holland, J. (1998). *Emergence: from chaos to order*. Reading, MA: Addison-Wesley.
- IBM. (2001). *autonomic computing: IBM's Perspective on the State of Information Technology*. Armonk, NY: IBM.
- IBM. Autonomic Computing Project website. [www.ibm.com/research/autonomic](http://www.ibm.com/research/autonomic)
- Kauffman, S. (1995). *At Home in the Universe*. New York: Oxford Univ. Press.
- Kauffman, S. (1993). *The Origins of Order*. New York: Oxford Univ. Press.
- Kelemen, J., & Sosik, P., eds. (2001). *Advances in Artificial Life: 6th European Conference, ECAL 2001, Prague, Czech Republic, September 10-14, 2001: proceedings*. Springer.
- Luisi, P., & Varela, F. (1990). Self-Replicating Micelles - A Chemical Version of Minimal Autopoietic Systems. *Origins of Life and Evolution of the Biosphere*, 19, 633-643.

- Luisi, P. (1993). Defining the Transition to Life: Self- Replicating Bounded Structures and Chemical Autopoiesis. In: *Thinking About Biology*, Eds. W. Stein and F.J. Varela, SFI Studies in the Sciences of Complexity. Addison-Wesley.
- Luisi, P. (1994). The chemical implementation of autopoiesis. *NATO ASI Series C Mathematical and Physical Sciences - Advanced Study Institute*, 446, 179-198.
- Luisi, P. (1998). About Various Definitions of Life. *Origins of Life and Evolution of the Biosphere*, 28, 613-622.
- Luisi, P.; Walde, P.; & Oberholzer, T. (1999). Lipid vesicles as possible intermediates in the origin of life. *Current Opinion in Colloid & Interface Science*, 4, 33-39.
- Luisi, P. (1997). Self-Reproduction of Chemical Structures and the Question of the Transition to Life. *Astronomical and Biochemical Origins and the Search for Life in the Universe*, Eds. C.B. Cosmovici, S. Bowyer, D. Werthimer, Editrice Compositori, Bologna, 461-468.
- Luisi, P. (2000). The Relevance of Supramolecular Chemistry for the Origin of Life. *Advances in Supramolecular Chemistry*, 6, 287-307.
- Maturana, H., & Varela, F. (1980). *Autopoesis and Cognition*. Dordrecht, Holland: D. Reidel.
- Maturana, H., & Varela, F. (1987). *The Tree of Knowledge*. Boston: Shambhala.
- Mavelli, F.; & Luisi, P. (1996). Autopoietic Self-Reproducing Vesicles: A Simplified Kinetic Model. *Journal of Physical Chemistry*, 100, 16600-16607.
- McMullin, B. (1997a). Computational Autopoiesis: The Original Algorithm. SFI Working Paper 97-01-001. Santa Fe: Santa Fe Inst. Retrieved April 2, 2002 from <http://www.santafe.edu/sfi/publications/Working-Papers/97-01-001.pdf>.
- McMullin, B. (1997b). SCL: An Artificial Chemistry in Swarm. SFI Working Paper 97-01-002. Santa Fe: Santa Fe Inst. Retrieved April 2, 2002 from <http://www.santafe.edu/sfi/publications/Working-Papers/97-01-002.pdf>.
- McMullin, B. (1997c). Modelling Autopoiesis: Harder Than It May Seem! Presented at Biology, Cognition, Language & Society: An International Symposium on Autopoiesis. Belo Horizonte, Brazil, November 1997. Retrieved August 24, 2002 from <http://www.eeng.dcu.ie/~alife/bmcm9704/ bmcm9704.html>
- McMullin, B., & Gross, D. (2001). Towards the Implementation of Evolving Autopoietic Artificial Agents. Retrieved August 18, 2002 from <http://www.eeng.dcu.ie/~alife/bmcm-ecal-2001/ bmcm-ecal-2001.pdf>.
- McMullin, B., & Varela, F. (1997). Rediscovering Computational Autopoiesis. SFI Working Paper 97-02-012. Santa Fe: Santa Fe Inst. Retrieved April 2, 2002 from <http://www.santafe.edu/sfi/publications/Working-Papers/97-02-012.pdf>.
- McMullin, B. website and SCL/autopoiesis software using swarm. [www.eeng.dcu.ie/~alife/](http://www.eeng.dcu.ie/~alife/)
- Mendes, P. (1993). GEPASI: a software package for modelling the dynamics, steady states and control of biochemical and other systems. *Comput. Appl. Biosci.*, 9, 563-571.
- Mendes, P. (1997). Biochemistry by numbers: simulation of biochemical pathways with Gepasi 3. *Trends Biochem. Sci.*, 22, 361-363.
- Mendes, P. (1998). Gepasi 3.20. Retrieved from [gepasi.dbs.aber.ac.uk/softw/gepasi.html](http://gepasi.dbs.aber.ac.uk/softw/gepasi.html)
- Mingers, J. (1995). *Self-Producing Systems - implications and applications of autopoiesis*. New York: Plenum Press.

- NetLogo. website and software. [ccl.northwestern.edu/](http://ccl.northwestern.edu/)
- Nomura, T. (2001). Miscellaneous - Formal Description of Autopoiesis Based on the Theory of Category. *Publication Lecture Notes in Computer Science*, 2159, 700-703.
- Purves, D., et al. (1997). *Neuroscience*. Sunderland, MA: Sinauer Assocs.
- Rational Rose RealTime. Rational Software.
- Ray, T. (1991). An Approach to the Synthesis of Life. In Langton, C., Farmer, J., & Rasmussen, S. (eds.). *Artificial Life II*. Addison-Wesley.
- Ray, T. (1995). An Evolutionary Approach to Synthetic Biology: Zen and the Art of Creating Life. In Langton, C. (ed.). *Artificial Life - An Overview*. Cambridge, MA: MIT Press.
- Ray, T. website and Tierra software
- Repast. website and software. [repast.sourceforge.net/](http://repast.sourceforge.net/)
- Shapiro, E. (2002). website. [www.wisdom.weizmann.ac.il/~udi/](http://www.wisdom.weizmann.ac.il/~udi/)
- Solso, R. (2001). *Cognitive Psychology, 6th ed.* Boston: Allyn and Bacon.
- StarLogo. website and software. [el.www.media.mit.edu/groups/el/Projects/starlogo/home.htm](http://el.www.media.mit.edu/groups/el/Projects/starlogo/home.htm)
- Swarm. website and software. [www.swarm.org/index.html](http://www.swarm.org/index.html)
- Thearling, K., & Ray T. (1994). Evolving Multi-cellular Artificial Life. In Brooks, R., & Maes, P. (eds.). *Proceedings of Artificial Life IV*. Cambridge, MA: MIT Press.
- Varela, F. J., Maturana, H. R., & Uribe, R. (1974). Autopoiesis: The Organization of Living Systems, its Characterization and a Model, *BioSystems* 5: 187-196.
- Varela, F. J. (1979). *Principles of Biological Autonomy*. New York: North Holland.
- Varela, F.J. (1996). The Early Days of Autopoiesis: Heinz and Chile. *Systems Research*, 13, 407-416.
- Walde, P.; Wick, R.; Fresta, M.; Mangone, A.; & Luisi, P. (1994). Autopoietic Self-Reproduction of Fatty Acid Vesicles. *Journal of the American Chemical Society*, 116, 11649-11654.
- Webb, K. (1999). *Cell Model simulation using Rational Rose RealTime*. Ottawa.
- Whitaker, R. (2001). website on autopoiesis. <http://www.informatik.umu.se/~rwhit/AT.html> or <http://www.enolagaia.com/AT.html>
- White, A. (2000). *SynthECA: A Synthetic Ecology of Chemical Agents - PhD thesis*. Ottawa: Carleton Univ.
- Winograd, T., & Flores, F. (1985). *Understanding Computers and Cognition*. Norwood, NJ: Ablex Publishing Corp.
- Zaretsky, A.; & Letelier, J. (2002). Metabolic Networks from (M, R) Systems and Autopoiesis Perspective. *Journal of Biological Systems*, 10, 265+.
- Zeleny, M. (1977). Self-Organization of Living Systems: A Formal Model of Autopoiesis, *International Journal of General Systems*, 4, 13-28.
- Zeleny, M.; & Hufford, K. (1992). The application of autopoiesis in systems analysis: are autopoietic systems also social systems? *International Journal of General Systems*, 21, 145-160.
- Zepik, H.; Blöchliger, E.; & Luisi, P. (2001). A Chemical Model of Homeostasis *Angew. Chemie Int.Ed.* 40 No.1, 199-202.