THE EVOGRID:
An Approach to Computational Origins of Life Endeavours

Bruce Frederick Damer, BSc, MSEE

The thesis is submitted to University College Dublin in partial fulfilment
of the requirements for the degree of Doctor of Philosophy
in the College of Human Sciences

School of Education Doctoral Studies
SMARTlab Practice-based PhD Programme

Ionad Taighde SMARTlab
An Coláiste Ollscoile, Baile Átha Cliath
Ollscoil na hÉireann

Head of School: Dr. Marie Clarke

Principal Supervisor: Professor Lizbeth Goodman

Doctoral Studies Panel Membership:
Professor Jacquelyn Ford Morie
Professor Sher Doruff
Professor Dominic Palmer-Brown

May 2011

Table of Contents

Abstract

The quest to understand the mechanisms of the origin of life on Earth could be enhanced by computer simulations of plausible stages in the emergence of life from non-life at the molecular level. This class of simulation could then support testing and validation through parallel laboratory chemical experiments. This combination of a computational, or “cyber” component and a parallel effort investigation in chemical abiogenesis could be termed a *cyberbiogenesis* approach. The central technological challenge to cyberbiogenesis endeavours is to design computer simulation models permitting *de novo* emergence of prebiotic and biological virtual molecular structures and processes through multiple thresholds of complexity. This thesis takes on the challenge of designing, implementing and analyzing one such simulation model. This model can be described concisely as: *distributed processing and global optimization through the method of search coupled with stochastic hill climbing supporting emergent phenomena within small volume, short time frame molecular dynamics simulations*.

The original contributions to knowledge made by this work are to frame computational origins of life endeavours historically; postulate and describe one concrete design to test a hypothesis surrounding this class of computation; present results from a prototype system, the EvoGrid, built to execute a range of experiments which test the hypothesis; and propose a road map and societal considerations for future computational origins of life endeavours.

Statement of Original Authorship

I hereby certify that the submitted work is my own work, was completed while registered as a candidate for the degree of Doctor of Philosophy, and I have not obtained a degree elsewhere on the basis of the research presented in this submitted work.

Signed:

Bruce Damer

Collaborations

With a decade of experience leading research and development projects for NASA I am well informed in the processes of defining and leading team projects in computer science and simulation. My role in this research project was to act both as Principal Investigator and Project Manager. My tasks included: deriving the research question, stating it as a testable hypothesis, performing the literature review and consulting with a global group of informal advisors, specifying the algorithms to be implemented in the prototype, designing the experiments to be run, building and operating the first computing grid to produce the initial data, plotting and interpreting the results and, finally, writing this thesis.

Collaborators on this work included three individuals. Under my direction, Peter Newman produced the software coding of the optimization algorithms and assisted me in the setup of the initial simulation grid, which I then operated to produce the first set of experimental data. Ryan Norkus produced some of the graphical treatments of explanatory diagrams in the thesis based on my sketches. Lastly, Miroslav Karpis built a 3D interface to visualize the activity in the output data sets, also under my direction.

List of Figures

Glossary of Key Terms

Ab initio – in Latin “from the beginning” within chemical experiments means that phenomena observed result from a system which starts from a basis of very simple molecules or free atomic elements. These phenomena are also described as emerging *de novo*.

Adjacent possible – from Stuart Kauffmann (Kauffman, 2000, p. 42) “the becoming of the universe can involve ontologically both the Actual and the Possible, where what becomes Actual can acausally change what becomes Possible and what becomes Possible can acausally change what becomes Actual.”

Artificial Chemistry – often abbreviated to *AChem*, is a system in computer software designed to simulate the dynamic motion and interaction of atoms, molecules or larger groups of molecules. Dittrich (Dittrich et al., 2001) defined an AChem as “a triple (S, R,A) where S is the set of all possible molecules, R is a set of collision rules and A is an algorithm describing the domain and how the rules are applied to the molecules inside (the physics).”

Artificial life – a field of computer science which seeks to simulate aspects of living systems within abstract computational universes. *Alife*, as it is abbreviated often seeks to model and study evolutionary processes.

Abiogenesis – the study of how living systems arose from non-living molecules.

Cameo simulation – a term coined by the author to refer to simulations of small volumes of artificial chemistries where the goal is the observation of isolated, limited phenomena such as the formation of the single type of molecule.

Cellular automata - a mathematical construct in which a regular grid of cells, each in a finite number of states, interact by changing their states based on a set of rules of interaction with a neighbourhood of cells adjacent to each cell.

Chemical equilibrium – the state achieved when the rates of conversion of chemical X to chemical Y and the backward conversion of chemical Y to X are equal.

Classical dynamics – a system employing Newtonian dynamics or “magnetic ball” metaphors in the interaction of particles such as atoms as opposed to *quantum dynamics* interactions.

Coarse graining – a method for modeling and simulating chemical interactions in which atoms or molecules are grouped together as units. These techniques offer significant cost savings over atom-scale simulation such as *Molecular Dynamics*.

Commodity cluster – a set of standard, commercially available computer systems networked together into a single computing resource to be applied to scientific and other problems.

Complexity – a broad term applied to a number of fields but in the context of biochemistry and simulation it is a property of a system to generate a number of structures (molecules) or relationships between structures (reactions). The larger variety of structures and relationships, the more complex the system is deemed to be.

Cyberbiogenesis – a new term coined within this thesis that describes a pathway that begins with a simulation of a molecular origin of life stage or series of stages, that continues into verification by bench chemical experimentation.

Distributed or grid computing – a branch of computer science that builds and studies software systems working across a distributed network of computers. These systems often operate by breaking up a large computational problem into pieces and perform computing on those pieces within different computers.

de novo – from Latin "from the beginning" used in biochemistry to indicate that complex molecules have been synthesized from the interactions of simpler molecules. Related to the term *ab initio*.

Emergence – within the fields of simulation, biochemistry, and evolutionary biology, the appearing of a new structure or behaviour that is a substantial departure in form or function from the simpler components which combined to make it occur.

Epistemology – in the scope of this work, the framework of acquiring knowledge through the scientific positivist approach utilizing a hypothetico-deductive method.

Ergodic – a hypothesis that says that over long periods of time particle which has access to microstates will with equal probability occupy these microstates over the whole phase space.

Fidelity – a measure of the quality of a simulation when compared with the physical phenomena being modeled, the closer the outcomes of the simulation are to predicting the behavior in physical reality, the higher the fidelity of the simulation.

Fitness landscape – a mathematical abstraction of the solution space occupied by adaptive systems such as those found in evolutionary biology or chemical systems. Fitness landscapes are expressed in terms of "hills" and "valleys" plotted from the scores generated by evaluations of a fitness functions for candidate solutions, which as an example might include a living organism or a chemical catalyst.

Genes of emergence – a term by the author in which the parameters to a *cameo* chemical simulation when coupled with a search function may be thought of as a genetic code expressing the potential for emergent phenomena in future executions of the simulation space.

Hill climbing - a mathematical optimization technique utilizing iteration to incrementally change one element of a solution and if that produces a more optimal solution, change that element again until no further optimizations can be found. This is a means for the discovery of *local optimal or maxima*.

Hypopopulated – a term by Kauffman referring to large chemical reaction graphs in which there are a sparse number of reactions occurring.

In silico – an expression meaning that an action is performed by software running on a computer system, often in contrast to the term *in vitro*.

In vitro – from the Latin "within glass" is an expression meaning that an action is performed within physical chemistry, such as in wet or bench chemistry in a laboratory setting.

Interstellar chemistry – the chemical regime of the space between star systems, usually characterized by the presence of atomic elements, dust and icy particles.

Local optima or maxima – a formalism in mathematics and computer science in which a solution appears to be optimal or at a maximum when compared with neighboring solutions. This concept is often applied to fields such as data mining, materials science, and evolutionary biology, all of which are characterized by large sets, or *fitness landscapes* of possible solutions.

Molecular dynamics - a computer science discipline of molecular modeling and computer simulation utilizing statistical mechanics. Molecular dynamics usually models chemical systems at the level of individual atoms, as opposed to *coarse graining* techniques which might model groups of atoms as a unit.

Ontology – is the philosophical study of reality and the nature of being and its basic categories. In the context of this work we use the ontological assumption that it is possible to know the mechanisms of life arising from non-life.

Optimization – a method or algorithm in a computer simulation system which is designed to improve the performance of that system by a significant factor. Performance improvements could include increased likelihood of an emergent phenomenon occurring, and a reduction of time or computing resources necessary for phenomena to occur.

Origin of life – the field of science which seeks to understand and test plausible pathways from a world of simple pre-biotic molecules and a world of biological entities, sometimes called *protocells*,

Physicodynamic – a term coined in (Abel, 2009b) which expresses the actions observable in nature which are entirely driven by physical processes, as opposed to models in science which are built from logical and symbolic formalisms.

Physics – a set of abstract, often formulaic representations of observed dynamical behavior in nature, such as the movement of and interaction between objects like atoms, balls or star systems.

Protocell – a term in origins of life research indicating a molecular complex exhibiting early traits of a living system, akin to a cell. A protocell may have the properties of encapsulating a volume, supporting metabolic energy and material handling, and the reproduction of the whole system into "daughter" protocells.

Quantum dynamics Interactions – is the quantum version of *classical dynamics* interactions often modeled at the level below individual atoms where motion, energy, and momentum exchanges are governed by the laws of quantum mechanics.

Ratchet – a mechanical device such as a gear that allows movement in only one direction due to a mechanism limiting backwards movement. This is applied to complexity problems and evolution through the observation that phenomena becoming more complex resist losing that complexity and are therefore said to be subject to a ratcheting effect.

Reaction Graph - a representation of a set of reacting chemicals that transform into one another.

Search – an *ontological* mechanism represented in a software algorithm that seeks to track the behavior of a data set from a simulated system and report that behavior to an end user.

Simulation – the implementation of a *physics* as an abstract model of reality in software and the execution and analysis of that software in computers.

Stochastic - from the Greek for *aim* or *guess* denotes random. A stochastic process is one based on non-deterministic or probabilistic inputs as well as by predictable factors.

Stochastic hill climbing – a method of *hill climbing* for navigating *fitness landscapes* which uses a local iterative optimization involving the random selection of a neighbor for a candidate solution but only accepting it if the neighbor is equal to or improves upon the current or parent solution.

Acknowledgements and Dedications

The process of researching cyberbiogenesis computational origins of life endeavours and then constructing and testing the EvoGrid prototype was a highly interdisciplinary activity. This work touched on a wide range of fields including computer simulation, complexity science, artificial life, biochemistry, origin of life biology, philosophy, religion, and ethics. Before undertaking this endeavour I sought out an independent, informal but significant advisory group to review and critique the research goals, and to suggest further readings and avenues of investigation. I would like to thank the following individuals for their irreplaceable counsel, direction, insights and, especially, help with testing and shaping the ideas in this thesis:

- Professor Richard Gordon, Professor of Radiology, University of Manitoba, for his guidance on complexity, concepts of genesis and emergence in nature, and for posing the challenge that led to this effort
- Mr. Tom Barbalet, founder of the Noble Ape project, publisher of Biota.org and host of the Biota podcast, for his continuous tracking and feedback on the EvoGrid concept and enterprise, including hosting public discussions on the concept through the Biota podcast
- Professor David Deamer, Professor of Chemistry, University of California at Santa Cruz, for advice in the principles of biochemistry and models for the origins of life, and for providing contacts with those working in the field.

In addition, I would like to acknowledge the following individuals for their many contributions:

- Professor Tom Ray, Department of Zoology, University of Oklahoma at Norman, for challenging me early on about what was being attempted and why it would be new from the perspective of progress in artificial life.
- Professor Steen Rasmussen of the Dept. of Physics and Chemistry, University of Southern Denmark, for providing insights into *in vitro*

chemical protocell models and the importance of simulation for the field.

- Dr. Martin Hanczyc, Dept. of Physics and Chemistry, University of Southern Denmark, for shepherding me into the experimental world of chemical protocell formation and behaviour.
- Dr. Harold Fellermann, Dept. of Physics and Chemistry, University of Southern Denmark, for providing first hand insight into the challenges of simulating chemical activities and for his exemplary work on lipid nanocell simulation which served as a major proof-of-concept for this work.
- Professor Anand Rangarajan, University of Central Florida, Gainesville, for providing a historical review and key problems of the field of complexity science and helping me to place the EvoGrid within this context.
- Professor Stuart Kauffman, formerly of the Institute for Biocomplexity and Informatics, University of Calgary, Canada, and currently Finland Distinguished Professor at Tampere University of Technology, for spending time explaining how the universe might be organized into formulaic, complex and novelty driven phases (the "Adjacent Possible"), depending on where and when you are looking. I also have to thank him for giving a major direction to the EvoGrid project as an experimental platform to explore his concept of the Adjacent Possible and a possible "Fourth Law of Thermodynamics".
- Professor Doron Lancet, Dept. Molecular Genetics, Weizmann Institute, Israel, for great encouragement by emphasizing the necessity for this work and for illuminating approachs to distributed artificial chemistry. I would also like to acknowlege him for coining the term "computational origin of life endeavour".
- Dr. Penny Boston, Dept of Earth & Environmental Science, New Mexico Tech, for offering the key unanswered question of this research: whether to simulate the chemistry or construct a more tractable abstract universe to show the way to complex self organization.
- Dr. Rachel Armstrong, Bartlett School, University College London, UK, for leading me into the world of chemical protocell research through her energy, knowledge and personal connections in the field.

- Dr. Neil Datta, then at Imperial College London and now an independent researcher, for discussions on mathematical representations of complexity and for reviewing the clarity and presentation of research in the draft thesis.
- Professor Piet Hut, Institute for Advanced Study, for encouraging me to pursue institutional affiliation and the PhD degree and introducing me to key people and resources at the Institute.
- Professor Freeman Dyson, Institute for Advanced Study, for providing kind enthusiasm, effort and guidance about the true messiness of nature as contrasted with the overly neat toy universes often realized through computing.
- Mr. George Dyson for providing a compelling first hand account of the history of computing at the Institute for Advanced Study and Nils Aall Barricelli's early digital experiments in artificial evolution.
- The Shelby White and Leon Levy Archives Center at the Institute for Advanced Study for assistance with and access to the Robert Oppenheimer director's files on the Electronic Computer Project, including John von Neumann's correspondance, notes, reports and output from Nils Aall Barricelli's numerical symboorganisms program.
- Dr. Nick Herbert, physicist and author, for personal guidance and encouragement and insights into the non-intuitive world of quantum dynamics.
- Dr. Brian Allen, MAGIX Lab, UCLA, for bringing to my attention the key observation of the property of thresholds of complexity in any complex, lifelike system.
- Dr. Karl Sims, GenArts, Inc., Cambridge MA, for hiking up the mountain with me at the Burgess Shale, providing inspirational early work in artificial evolution, and continued enthusiasm for my efforts.
- Dr. Tim Taylor, independent researcher, for his earlier influential work in the simulation of bio-inspired systems.
- Mr. Tim McFadden, independent researcher, San Jose, for bringing to my attention Shannon information theory and alternative ways of considering information in nature.
- Dr. Peter Bentley, University College London, UK, for presenting challenges regarding the viability of the von Neumann computer architecture to simulate nature.

- Mr. Gerald de Jong, Rotterdam, the Netherlands, for building and fielding accessible works of artificial evolution to a World Wide Web audience.
- Professor Addy Pross, Department of Chemistry, Ben-Gurion University of the Negev, Israel, for taking the time to explain passionately his perspective on the genes-first model of the origin of life.
- Professor Robert Shapiro, New York University, for kind guidance of my understanding of the arguments on all sides of the origins of life question.
- Dr. Jim Cleaves, Geophysical Laboratory, Carnegie Institution for Science, Washington D.C, for providing key historical perspective and materials in origins of life thinking.
- Dr. Andrew Pohorille, Exobiology Branch, NASA Ames Research Center, for providing many writings about early membrane and protocell development and a perspective from Astrobiology.
- Dr. Sandra Pizzarello, Arizona State University, for providing insights into extraterrestrial sources of bio-relevant materials.
- Dr. Wes Clark and Dr. Maxine Rockoff for their repeated hospitality in Brooklyn and for asking key questions and suggesting valuable personal connections.
- Mr. Fred Stahl, Author, Arlington, VA, for providing key historical materials on his work in the early 1960s that picked up where John von Neumann left off.
- Professor Emeritus Richard Dawkins, Oxford University, UK, whose hospitality, early encouragement and challenge in defining a living system and simulating evolution led directly to the formulation of the EvoGrid effort.
- Last but not least in this list I would like to thank Dr. Chris Langton, who warmly welcomed me as a visitor to the Santa Fe Institute in 1994 and over the years supported the Digital Biota conferences and our frequent meetings in virtual worlds.

I would also like to acknowledge the support team at my company DigitalSpace who produced the computer software coding of the EvoGrid prototype under my direction. This implementation of my designs permitted testing of the hypothesis proposed in this thesis. Peter Newman worked tirelessly to assemble standard open source components and then code the unique connective algorithms to implement the first prototype. He also assisted me as I set up and operated the initial computing grid which generated the first results. Ryan Norkus is specially acknowledged for taking my scribbled sketches and producing many fine explanatory figures, 3D computer graphics and fully rendered movies. I also would like to thank Miroslav Karpis who volunteered his time to build the WebGL 3D interface to allow me to see and run additional analysis on the frames of atoms and molecules. Lastly I offer my gratitude to John Graham of Calit2 at the University of California at San Diego for working tirelessly to set up and support the second EvoGrid simulation network.

Next I would like to express immeasurable gratitude to Basit Hamid, Founder and CEO of Elixir Technologies Corporation for his belief in me through twenty-five years of friendship, for providing funding and logistical support for this PhD work including travel and fees, and for his lifelong encouragement of me to pursue academic excellence.

My supervisory team including Director of Studies Professor Lizbeth Goodman, Professor Sher Doruff, Professor Jacquelyn Ford Morie and external supervisor Professor Dominic Palmer-Brown are all thanked profusely for their time and careful review over the years and for always being there and coming through as I have advanced through the process.

I would also like to acknowledge and thank my wife, Galen Brandt, for providing her unwavering support for my life and health throughout this process and serving as a valued sounding board for these ideas. My brother Dr. Eric Damer is also thanked for doing a comprehensive pass over the thesis for syntax and clarity.

This work is posthumously dedicated to Douglas Adams and Professor Stephen J. Gould both of whom I sought contact with during the early research for this effort. Mr. Adams was a keynote speaker at the second conference in my Digital Biota series and would have appreciated the planet-sized nature of this computing challenge about life, the universe and everything. Professor Gould, while explaining to me how he was not very digital, patiently listened to my explanations of how we might simulate the Burgess Shale ecosystem and still wanted to be kept closely informed. So, Professor Gould, this is my belated report on progress so far.

A final dedication goes to my parents, Enid and Warren Damer. Enid instilled in me my work ethic and Warren my love of ideas. One of the last conversations with Warren before his passing was to update him on the progress of this PhD and I know he would be proud of and curious about this work.

Introduction

Preamble and Overview

The research question of this thesis was discovered through a *thought experiment* surrounding the prospects of the discovery, through digital simulation, of plausible pathways to an origin of life on Earth. From this thought experiment and a review of a number of cognate fields emerged a design for an initial prototype (the EvoGrid) as a possible first step toward a full computational origin of life endeavour. Evaluating the strengths and limitations of the prototype implementation then provided the basis to enumerate a roadmap and open questions for future researchers undertaking to simulate life's origins. An overview of the contents of the thesis follows:

- As contextual background, a history of the earliest concepts of using computers to simulate living systems;
- A thought experiment on the vision of simulating life's origins;
- A review of the main cognate fields approaches to computational origins of life endeavours including a novel map of the interrelationships of the cognate fields.
- A listing of the basic principles, assumptions and challenges facing such endeavours.
- A set of design choices in computing frameworks for origins of life endeavours.
- A prototype implementation (the EvoGrid) built, executed and results then analyzed to illustrate a few of the challenges that would be faced by future origin of life simulation efforts.
- A road map and enumeration of some open questions on the future evolution of these efforts.
- An exploration of likely philosophical, societal, ethical and religious questions and controversies posed by the prospect of an "artificial genesis".

The Origins of the Concept of Using Computers to Simulate Living Systems and Evolution

The modern quest for the understanding of possible mechanisms behind the origin of life, or in other words the *transformation of nonliving matter into living matter,* has been passed down to us from chemistry's precursors, Middle Ages alchemists (O'Connor, 1994). The mathematician Rene Descartes wrote in the seventeenth century of the then prevalent theory of spontaneous generation that "it is certainly not surprising that so many animals, worms, and insects form spontaneously before our eyes in all putrefying substances" (Margulis and Sagan, 2000, p. 64). Charles Darwin challenged the assertion of spontaneous generation in his *On the Origin of Species* (Darwin, 1859) arguing that species evolved from previous generations through a process of natural selection. In a letter to botanist Joseph Hooker, Darwin (1871) contemplated a chemical origin for life:

> It is often said that all the conditions for the first production of a living organism are present, which could ever have been present. But if (and Oh! what a big if!) we could conceive in some warm little pond, with all sorts of ammonia and phosphoric salts, light, heat, electricity, etc., present, that a protein compound was chemically formed ready to undergo still more complex changes, at the present day such matter would be instantly devoured or absorbed, which would not have been the case before living creatures were formed.

Work on the chemical origins of life progressed in the following decades through the early twentieth century work of Oparin (Oparin and Morgulis, 1938) and J.B.S. Haldane (Haldane, 1927) with hypotheses and experimentation regarding the conditions of the oceans and atmosphere, or what became popularly known as the "primal soup" of the early Earth. In 1953, chemists Stanley Miller and Harold Urey, reported their groundbreaking synthesis of amino acids within a chemical environment that simulated the estimated atmosphere on the early Earth (Miller, 1953). The Miller-Urey experiment caught the public imagination and sparked the quest for the origins of life in the test tube, inspiring decades of work on the chemical origins of life.

By the time of the Miller-Urey experiments, the quest was poised to move from the realm of speculative chemistry into the domain of digital experimentation with the arrival of the new medium of binary, electronic computation. George Dyson, son of the renowned physicist Freeman Dyson, has written extensively about the origins of the modern digital computer and early origins of life research at the Institute for Advanced Study (IAS) in Princeton, New Jersey. We will next summarize Dyson's recounting of this history from his book *Darwin Among the Machines* (Dyson, 1997). The author of this thesis also made a number of trips to the IAS during the conduct of his research which included two meetings with Freeman Dyson.

Central to this new development was the great mathematician John von Neumann, who had been a professor at the IAS since 1933 and had participated in such early computer projects as the Electronic Discrete Variable Automatic Computer (EDVAC) built for the U.S. Army by the University of Pennsylvania. Von Neumann had strong support for his work following World War II from the new director of the IAS, Robert Oppenheimer, who had left the Los Alamos Laboratory in New Mexico where he had been scientific director of the atomic bomb project. With Oppenheimer's sponsorship (and carefully choreographed protection from the Board of Trustees of the Institute), von Neumann led a team of scientists and engineers to create what might be considered the progenitor of the modern digital computer (Dyson, 1997, pp. 93-110). Simply called the IAS machine or, by those associated with the Institute, the Electronic Computer Project (ECP) machine, it was introduced to the world in mid -1952 (see Figure 1 and Figure 2).

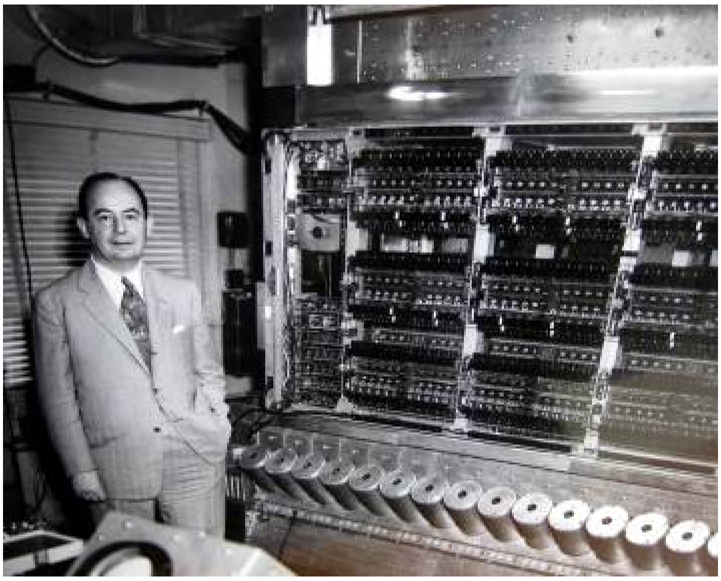


Figure 1 John von Neumann and the electronic computer at the Institute for Advanced Study, Princeton, New Jersey (photo courtesy The Shelby White and Leon Levy Archives Center, Institute for Advanced Study).

Figure 2 The building at the Institute for Advanced Study which housed the Electronic Computer Project (photo by the author).

The first two substantial programs coded to run on the machine were computation in aid of thermonuclear testing for the Department of Energy and weather prediction for the US Army. In the spring and summer of 1953, however, mathematical biologist Nils Aall Barricelli visited the IAS to begin a new line of research. The ECP's *Monthly Progress Report* for March, 1952, noted that:

> A series of numerical experiments are being made with the aim of verifying the possibility of an evolution similar to that

> of living organisms taking place in an artificially created universe. (as cited in Dyson, 1997, p. 111)

Barricelli proceeded to code what became known much later as an “artificial life” program onto punched cards and fed them into the ECP machine. The author visited the IAS archives in the spring of 2009 to view and study these materials first hand, including Barricelli’s original punched or “key” card deck (Figure 3).

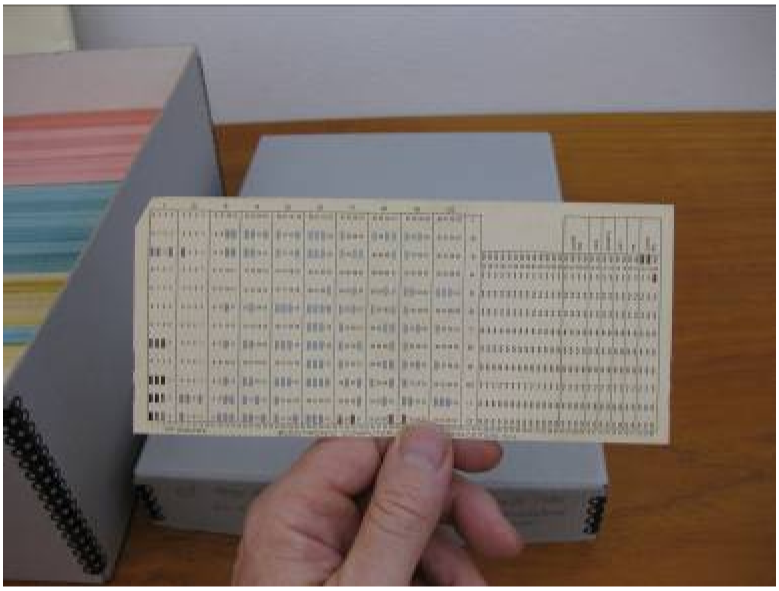


Figure 3 Punched card from the numerical symbio-organisms program for the IAS machine (photo by author).

Calling it an “experiment in bionumeric evolution”, Barricelli was investigating the role of symbiosis in the origin of life and came to believe that his five kilobyte universe of “numerical symbio-organisms” exhibited the key criteria of a living, evolving system. Barricelli ran his program over several weeks, executing thousands of iterations of arrays of 512 numbers. He reported the results in a chapter of a major report on the ECP machine (Barricelli, 1953) which will be described next.

Figure 4 The author with the “Barricelli blueprints”, punched card outputs of the memory of the program imaged on paper (photo courtesy The Shelby White and Leon Levy Archives Center, Institute for Advanced Study).

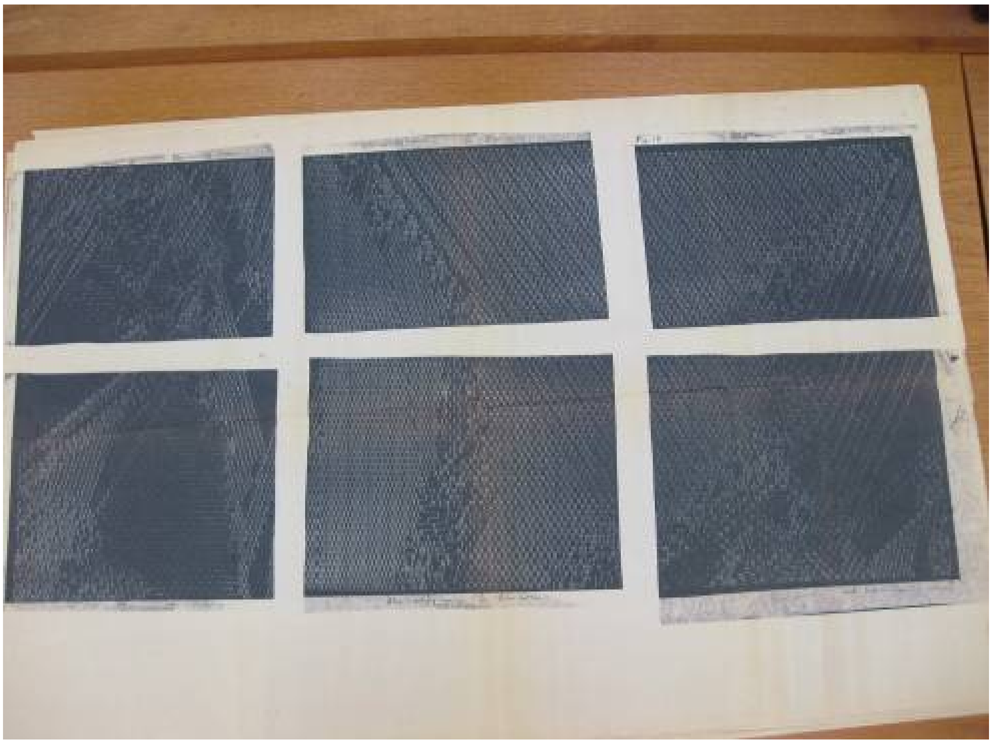


Figure 5 Close-up view of six of the Barricelli blueprints (photo by author).

Barricelli himself described his “Barricelli blueprints” as “output cards, punched with the contents of half the memory, when abutted top-to-bottom, present five generations of the 512 locations, in proper array… reproduced photographically and further assembled” (p. II-63;

see also Figure 4). Figure 5 shows several such generations, each with 512 locations. Barricelli coded his system to fit into the approximately five-kilobyte memory of the ECP machine. Barricelli wrote that “the code was written so that various mutation norms could be employed in selected regions of the universe… attention was paid to coding for maximum speed of operation, and for the convenient re-use of the output data as input after interrupted operation” (p. II-63). Barricelli’s *norms* are defined as reproduction and mutation rules for the numbers occupying the 512 locations in memory. In general, we can derive that he was executing a serial process of examining numbers each representing “organisms” and permitting them to change location, applying mutations to the numbers and dumping the entire “frame” of memory for examination and loading again to be restarted. Elsewhere in his report he described a key feature of any system intended to increase its organizational complexity (that is, the search for ever higher local maxima within a fitness landscape): “the evolution of an organism may for a long period of time stop in a relative maximum of fitness… but a change in the conditions of the universe, e.g. in the kind of concurrent organisms, may sooner or later level the maximum making further evolution possible” (p. II-87).

George Dyson (p. 117) summarized Barricelli’s results as follows:

> Barricelli knew that "something more is needed to understand the formation of organs and properties with a complexity comparable to those of living organisms. No matter how many mutations occur, the numbers... will never become anything more complex than plain numbers."(Barricelli, 1962, p. 73). Symbiogenesis--the forging of coalitions leading to higher levels of complexity--was the key to evolutionary success, but success in a closed, artificial universe has only fleeting meaning in our own. Translation into a more tangible phenotype (the interpretation or execution, whether by physical chemistry or other means, of the organism's genetic code) was required to establish a presence in our universe, if Barricelli's numerical symbioorganisms were to become more than laboratory curiosities, here one microsecond and gone the next.

Barricelli’s vision was profound, and provided a roadmap for future efforts in origin of life endeavors. The design principles Barricelli employed included:

1. a teleological goal to produce a system within which *de novo* emergence of complex phenomena could be observed;
2. a computationally optimized simulation of a relatively small set of objects organized in discrete locations of a simple universe and able to interact with each other while affected by global parameters;
3. the capacity for visual inspection and continued execution, when reloaded;
4. the quest of ever higher local maxima of some predefined set of goals or observation criteria constituting what later came to be called an *artificial fitness landscape*;
5. the capacity for the emergence of discrete “species” (types of object), but also the capacity for the system to devolve and lose such organization.

We can see in Barricelli’s vision and in Dyson’s analysis a sense of both the conundrums and long-term promise of digital simulation in support of origin of life endeavours. In Barricelli’s digital world, simple digital universes with their simple physics produced simple results. Such a view would eventually give way to a more complex conceptualization, but Barricelli’s ideas influenced developments in the field for some sixty years.

However compelling they may be to theorists or computer programmers, these worlds are of fleeting curiosity and of little utility to the study of life and its origins. Some means to achieve ever more complex virtual organisms and to test these creations in our universe (in physical chemistry or otherwise) are required to make this endeavour relevant to the broader quest for understanding living systems.

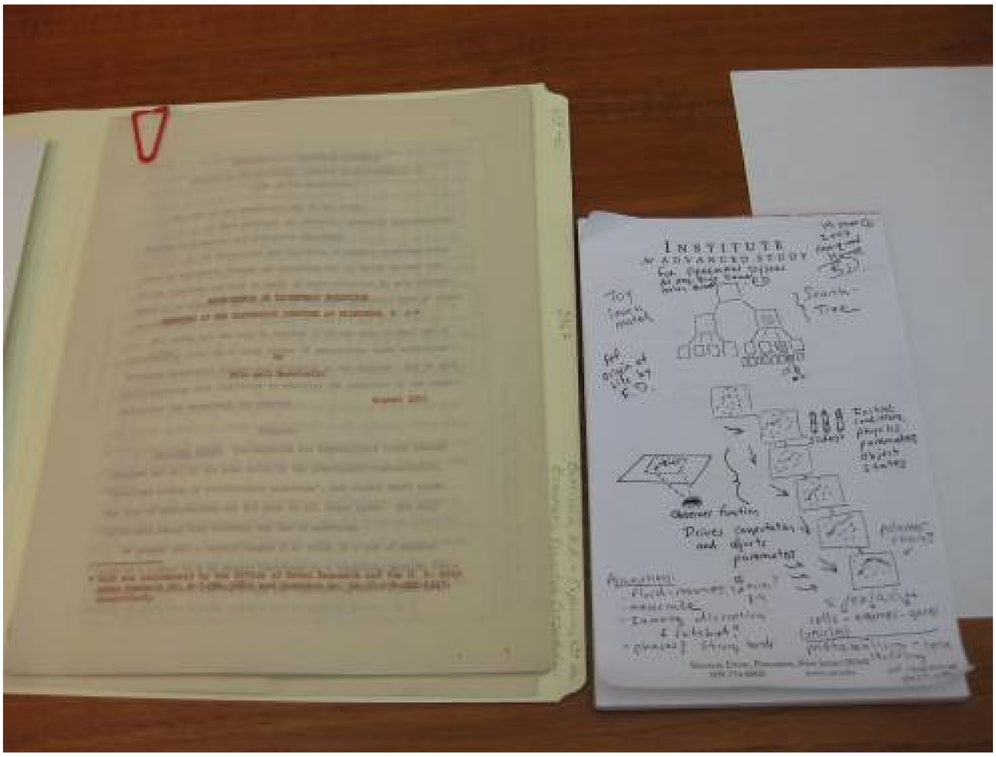

INSTITUTE FOR ADVANCED STUDY
Search Tree
Observer function
Drives computation and objects parameters

Figure 6 Barricelli’s original 1953 report on his Numerical Symbioorganisms project, along with the author’s design for the EvoGrid search tree function (photo by author).

Barricelli’s original August 1953 report (Barricelli, 1953) pictured in Figure 6 is set side by side with the author’s original design for the EvoGrid, the prototype implemented for this research. The parallels and differences between these approaches will be described below.

It was both inspiring and instructive to handle the materials of the first implementation of the long pursued dream of using computers to simulate living systems, especially systems that might express the properties of evolution. While Barricelli’s work ended decades ago, the design principles he established while striving to shoe-horn his life simulation into von Neumann’s first computer are still relevant today. For, despite advances in computing power and programming techniques we are still living with von Neumann’s fundamental computer architecture. This architecture consists of a few central processing units sequentially processing serial, branching instructions, and reading and writing to primary and secondary memory caches. The massive parallelism and other properties which permit Nature to “compute” cannot yet be matched within our early twenty-first century digital universes.

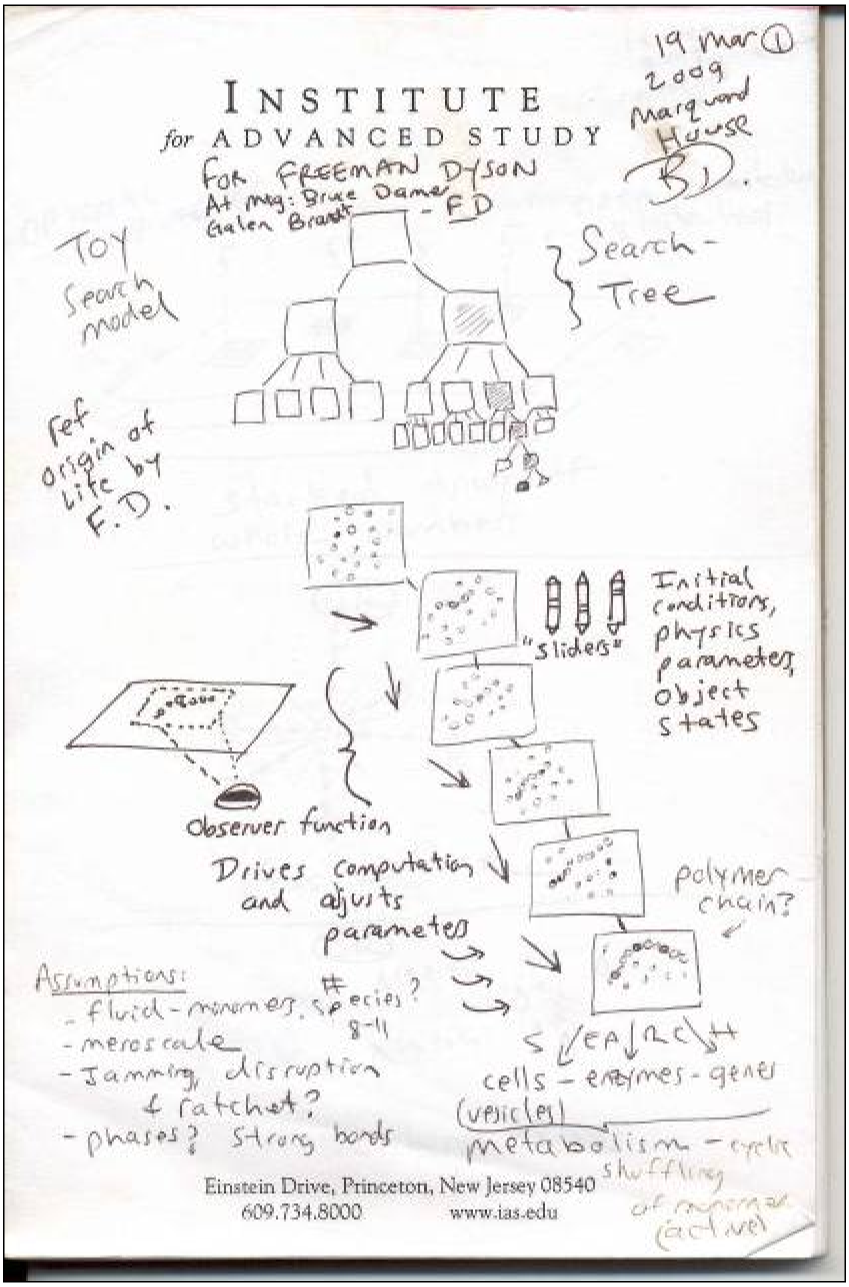


Figure 7 The author’s original sketch for the EvoGrid drawn in preparation meetings with Professor Freeman Dyson on the 11th and 19th of March, 2009 at the Institute for Advanced Study in Princeton, NJ.

The EvoGrid’s fundamental design and simulation criteria are shown in the sketch in Figure 7. This sketch was produced for a pair of meetings with Professor Freeman Dyson at the IAS. Dyson was a contemporary of both von Neumann and Barricell but by his own admission and to his regret did not get involved in computers. Dyson had been trying to bring Biology into the Institute for some years and had himself made serious investigations into origin of life thinking, including his “double-origin” hypothesis spelled out in (Dyson, 1999, p. 10). This hypothesis proposes that elements of a living system (amino acids, nucleotides, containers, metabolism, heredity mechanisms) might have arisen separately, replicated and developed without the exact precision derived from gene expression, and then been combined to create the first living systems. Chapter 4 of this thesis illustrates several chemical models along the lines of the double-origin hypothesis. Dyson also used the techniques of mathematically described “toy universes” in his thinking about the subject, which predisposed him to like the EvoGrid project as proposed. Dyson gave substantive input, summarized in Appendix B.2. The main point made by Dyson was that our simulations had to reflect the truly messy state of nature. One of his messier models for the prebiotic milieu is that of a large number of interacting molecules akin to “dirty water” contained in a “garbage-bag world” (p. 37). This figure is reproduced in Figure 39 in section 2.2 and forms the basis for the optimization used in this work

Traveling back and forth across the lawn between the office of Freeman Dyson and the IAS archives to view the Barricelli materials the author was struck by the similarity of design choices which were made by Barricelli as he had already intuited for the EvoGrid. As we have listed previously and shall explore further in Chapter 2, those common design choices included highly optimized execution of sets or “frames” of objects (simulated molecules) in an inheritance (or generation) hierarchy with varying simulation parameters and driven by a search and re-execution designed to permit the seeking of higher fitness maxima. The design sketch produced for Dyson and shown in Figure 7 illustrates the concept behind these computing frames operating in a search-driven inheritance hierarchy. The major additions to the author’s 2009 design over Barricelli’s 1953 architecture were the addition of fully automated search and restarting of simulations. In Barricelli's day the search for interesting phenomena in the simulation was accomplished manually by viewing the optical imaging of punched cards, and restarting simulations required manual feeding of cards back into the computing machinery. By the year 2010 a grid of computers could be used to run automated observations and the scoring of frames could then be used to select for automatic continuation of their execution. Barricelli's universe was made up of a two dimensional array of integers; the 2010 universe was also represented by numbers, but much more complexly structured to represent a three dimensional volume of virtual atoms.

The Development of the Modern Field of Artificial Life

Let us now continue our explorations of the historical underpinnings of the computational simulation of life. Later in the 1950s, John von Neumann proposed concepts of self-reproducing automata in a work which was published posthumously (von Neumann and Burks, 1966). Inspired by this vision, in 1960 Wayne State University student researcher Fred Stahl implemented one of the first von Neumann inspired self-reproducing cellular automata (CA) systems on an IBM 650 mainframe, a direct successor to the ECP machine (Stahl, 1961). Stahl's universe went beyond Barricelli's in terms of complexity as it featured an implementation of Turing's notion of a universal machine implemented for each of his simulated creatures using the computer's instruction set (Turing, 1950, Turing, 1937). Stahl's universe featured analogs for food and competing creatures capable of reproducing and mutation. This was followed in 1970 when *Scientific American* columnist Martin Gardner popularized British mathematician John Conway's "Game of Life" (Gardner, 1970) and brought the concept of cellular automata (CA) to the public's imagination. In his work, Conway was seeking to implement a simplified version of von Neumann's self-reproducing automata. The implementation of CAs was quite possibly the first true "experimental" environment for complex artificial worlds. CAs consist of arrays of cells in a set of two or more discrete states. State changes of cells depend on a set of coded rules that garner information from the state of neighboring cells. Decades of studying behavior in larger CA systems on faster computers has lead some, notably researcher and businessman Stephen Wolfram (Wolfram, 2002), to make some very large claims: that CA's are the fundamental operating units of the universe.

By the 1980s John von Neumann's original design for the electronic computer had come to dominate the computing world and began appearing on desktops as microcomputers. These tiny machines allowed intellectual successors to Barricelli such as researcher Chris Langton to work late into the night and code their own renditions of *life as it could be* while coining a term for a new field: *Artificial Life* (Langton, 1986, Levy, 1993). Artificial Life, sometimes abbreviated as AL or Alife, has a close cousin, artificial intelligence (AI) which is aimed at representing conscious thought. To avoid confusion, Alife is focused on a bottom-up approaches, hoping to simulate living systems at their simplest (Langton et al., 1992).

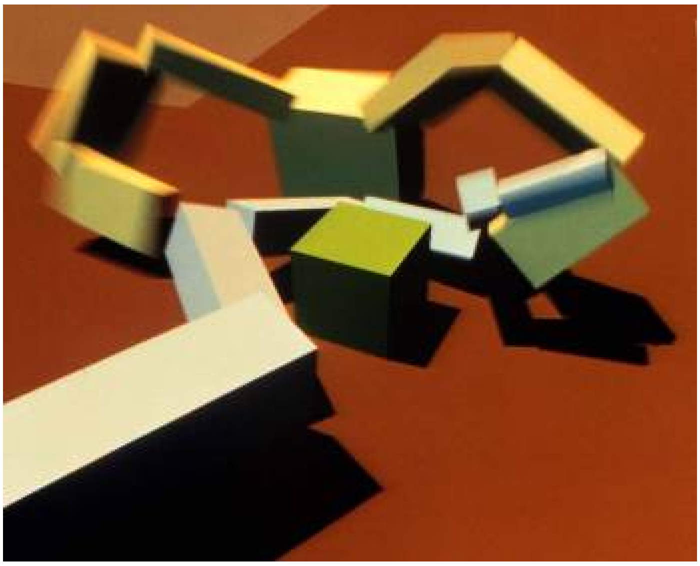


Figure 8 Karl Sims' evolving virtual creatures (image courtesy Karl Sims)

Karl Sims, for example, took this “bottom up” approach to new levels, using visual simulations to illustrate simulated evolution in a simple virtual universe with physics. His highly innovative work in the early 1990s combined the simulation of a genotype (a coding generating a directed graph) with the expression of a phenotype (groups of three dimensional hinged blocks) which was then subjected to mutation and selection pressures through competition for resources (Sims, 1991). Sims’ work was also one of the first Alife systems designed to run on a dedicated supercomputer, the Connection Machine, which supported thousands of individual von Neumann-type processing units.

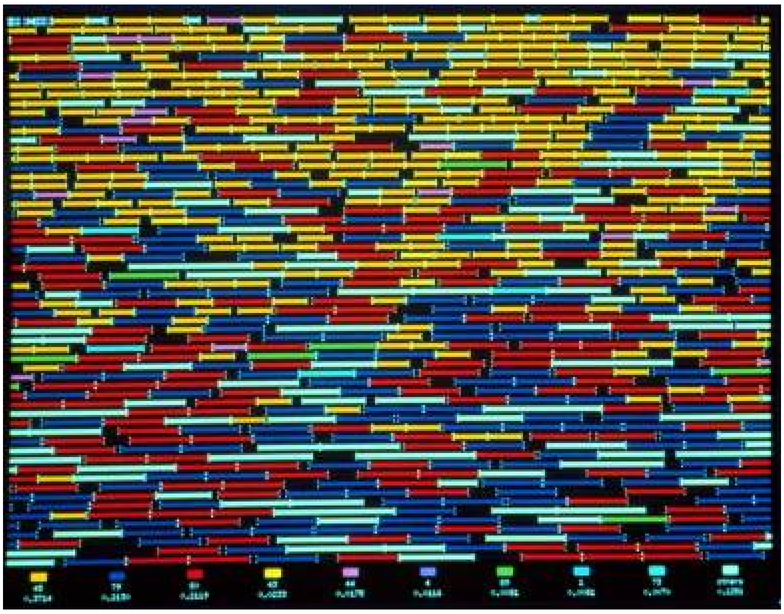


Figure 9 The Almond interface to Tierra (image courtesy Tom Ray)

During the same time period, work on the computer simulation Tierra concentrated solely on genotypic competition and evolution and was a direct descendent of the work of Barricelli (Ray, 1991). Tierra represented its universe as strings of data constantly seeking computing resources and available space to make copies (Figure 9). Random mutations were possible during copying and Tierra showed a number of fascinating emergent phenomena including the spontaneous rise of parasitism, which Barricelli also hinted at seeing in his first and subsequent experiments. Inspired by the increasing prevalence of another form of Alife, computer viruses, Tierra was adapted to run on networks and showed how a topology of competing “islands” of computing energy could shape the dynamics of populations (Ray, 1998).

In the 1990s there was great anticipation that increasing computing power would soon support simulated abstract ecosystems teeming with binary activity, which biologists would come to recognize as true living systems. However, it is the opinion of the author that by the turn of the last century Alife development had stalled. The strongest indication of this was the tendency for each Alife environment to fill up with a set of interesting phenomena but then show no ability to extend to a set of more complex phenomena. In other words, the Alife environments failed to show development through multiple levels of complexity sometimes referred to as “open ended” evolution.

This opinion was born out by a survey of the research community held at the Seventh International Conference on Artificial Life in the summer of 2000. The survey addressed issues including artificial life’s main successes, failures, open scientific questions, and strategies for the future (Rasmussen et al., 2003a). When the question “what are artificial life’s most significant failures?” was asked, the responses were, in order of priority:

> …too little theoretical or experimental grounding for the work done (“no rigor”), no coherent agreement on which scientific problems the community should address (“no direction”), and insufficient connection to other scientific fields (“unrelated”) (p. 218).

When polled on the key scientific issues to address, respondents replied that open-ended evolution was central. Next in importance was developing theory to supplant the proliferation of ad hoc work. Finally, respondents wanted a deeper understanding of life, either through devising a better definition or by creating life, and better understanding of dynamical hierarchies (p. 221).

During this time, the author held four conferences, the Digital Biota series I-IV (Damer, 1997-2001), in which Alife systems were presented by many of the leading practitioners of the field. By the fourth conference in 2001 it was clear that this lack of progress was a major stumbling block of non-trivial proportions. While there were many proposals for moving beyond this obstacle (Campbell, 2003) the author believed that the source of the problem was that the virtual worlds driving Alife experiments were too poor to support open ended emergence. Another way of saying this is that the physics used to compute the virtual reality of these worlds was too simplistic. We do know, however, that the real-world physics of chemistry is able to support the open-ended complexification of life. The natural question one might then pose is: *must we create virtual environments modeling living systems closer to the reality of physical chemistry?* Taking on this question could address many of the above survey conclusions of Rasmussen et al. by aligning Alife with another scientific field: biochemistry. Next, rigour can be introduced by taking on the challenge of chemical simulation. Finally, a specific scientific goal can be selected, for example, the observation of a system exhibiting open-ended growth of complexity. In this way, the system proposed in this thesis could provide one avenue around the stumbling block currently faced by the Alife field.

A New Synthesis: Computational Abiogenesis

In the 2000s interest was again growing in creating chemically-based or *in vitro* experiments supporting origins of life endeavors. One class of experiments undertaken during this time sought to observe the formation of protocells (Rasmussen et al., 2008). Protocells are loosely defined as chemical structures exhibiting at least some properties of a living cellular system. In parallel to this development in the laboratory, massively distributed peer-to-peer computation, large scale centralized grids, and special purpose computational chemistry computing hardware were being put into operation. These computing environments were capable of hosting viable realistic simulations of very small volumes of interacting molecules over short but biologically significant time scales (Shaw, 2009). At the start of the second decade of this century, a true synthesis of *in silico* simulation as a tool to design and predict the outcomes of *in vitro* experimentation seems to beckon to us from just over the horizon.

This synthesis holds the promise of new tools for chemistry akin to those delivered by computer aided design (CAD) and enjoyed by other fields such as product manufacturing, architecture, and in the case of the experience of the author, the design of spacecraft and missions (Damer et al., 2006). Numerous discrete element (sometimes called “quantum” or “particle”) simulators have been implemented in recent years, many taking advantage of common 3D game hardware known as graphics processing units (GPUs)(Nvidia, 2011). Many of these GPUs feature hardware supporting particle simulation software for the generation special effects. In the past decade the author engaged in related simulation work with teams at NASA and the Colorado School of Mines surrounding the simulation of robotics and granular materials for computer aided design in Lunar exploration (Taylor et al., 2005). Today, chemists are beginning to be able to use GPUs and other tools of computation to simulate particle-oriented frameworks involving individual chemical reactions and larger molecular structures such as proteins that give a measure of prediction of outcomes in chemical *in vitro* experiments (Phillips et al., 2005).

Such a synthesis also brings up a new and tantalizing possibility:

> *Could we actually one day digitally simulate a complete plausible step-by-step chemical scenario for an origin of life on Earth? And if we could carry out such a simulation while remaining faithful to the chemistry, could we then reproduce this particular pathway to life from non-life on the chemistry workbench?*

The computing part of this challenge was perhaps most definitively issued by Richard Gordon in the book *Divine Action and Natural Selection: Questions of Science and Faith in Biological Evolution* (Gordon, 2008). In his chapter titled "Hoyle's Tornado Origin of Artificial Life: A Programming Challenge", Gordon challenges the artificial life community to develop a computational environment to simulate an origin of artificial life from artificial non-life (pp. 354-367):

> I would like to suggest that artificial life (Alife) enthusiasts take up Fred Hoyle's challenge (Hoyle, 1984), that in a way they simulate a tornado going through a junkyard of parts, and come up with something we would all agree is alive, in the Alife sense, from components that are not alive in the Alife sense...

This author's response to Gordon's challenge was detailed in another chapter "The God Detector" (Damer, 2008, pp. 66-85) in the same volume (and also included in full in Appendix A.3 of this thesis):

> What I am proposing is to engage all of the best programmers, artists and philosophers of our generation to create a gigantic network of software and computers, working to create a sort of "Evolution Grid" or "EvoGrid". This EvoGrid would start out as God the Mechanic (like Karl Sims' creatures) in which we build the simulation, set the initial conditions and then let the artificial ecosystem go from there.

As described previously we adopt the term *cyberbiogenesis* to capture the synthesis of the two approaches: computational and chemical abiogenesis combining digital simulation with atomic realization. This word is a cousin of Mereschkowsky's term *symbiogenesis* (Mereschcowsky, 1909) which Margulis (Margulis, 1997) argues was a primary driver of evolution. Symbiogenesis holds that living systems, such as cells for example, evolve through emerging symbiotic relationships of parts that once were free standing organisms. Individual organelles such as the mitochondrion ceased being a separate organism and instead became the energy system for most eukaryotic cells. Cyberbiogenesis is related in that the computer, or cyber, simulation of emergent forms would become the building blocks of chemical experimentation. As multiple stages in an origin of life model are simulated and replicated chemically, there would emerge a symbiotic relationship between the discovery system simulation and the emerging chemical model.

The scope of constructing an end-to-end cyberbiogenesis system would likely be much more challenging than the recently completed project which sequenced the Human genome (Watson and Cook-Deegan, 1991) but is possibly realizable within this century. The chemical fabrication aspect of the cyberbiogenesis challenge is perhaps best captured by the field of synthetic biology. The recent announcement by Craig Venter of the *in vitro* substitution of a synthetically created genome into a living cell (Venter et al., 2001) seems to suggest that the fabrication of significant additional parts of living cells might also be possible. To compute an *in silico* origin of life faithful enough to physical laws of chemistry to be reproducible *in vitro* is perhaps one of the most audacious applications of technology in the history of human civilization.

This thesis will focus on the computational aspect of cyberbiogenesis, first explored through some thought experiments, and then through the process of design and implementation of an early computer prototype simulation. We will conclude by illuminating a possible road map for future efforts and considering some of the many scientific and technological open questions as well as societal issues that might emerge around such an effort. We hope that this thesis will lend some shape to cyberbiogenesis as a possible grand challenge for the coming century for those who might choose to continue the endeavour.

A Thought Experiment

In mid 2008 the author engaged in a *Gedankenexperiment* (thought experiment), drew storyboards (Figure 10) and requested a collaborator to produce a short animated movie (Damer et al., 2008) designed to illustrate the concept of cyberbiogenesis.

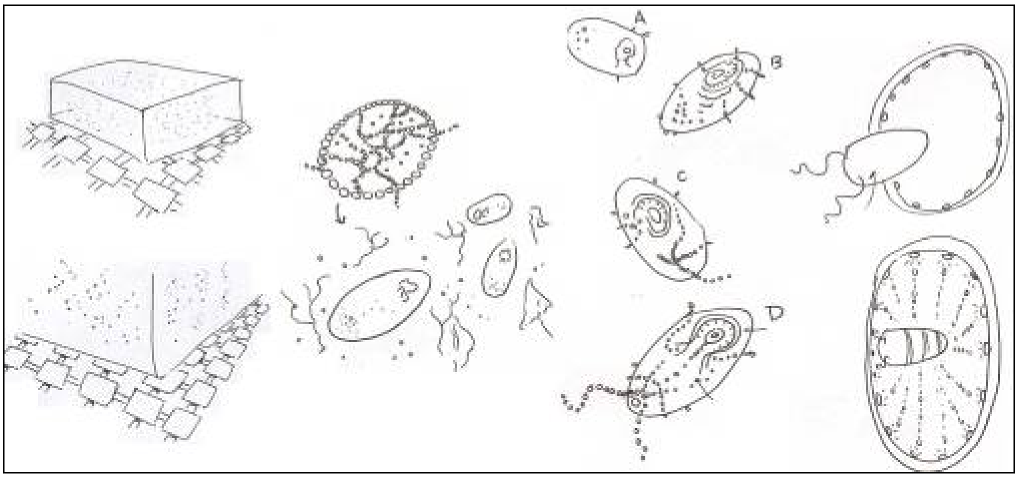


Figure 10 The author’s original sketches of the thought experiment

Figure 11 through Figure 20 below depict and describe scenes from the film which provides a visual cartoon of the thought experiment which imagines a completely realized cyberbiogenesis system.

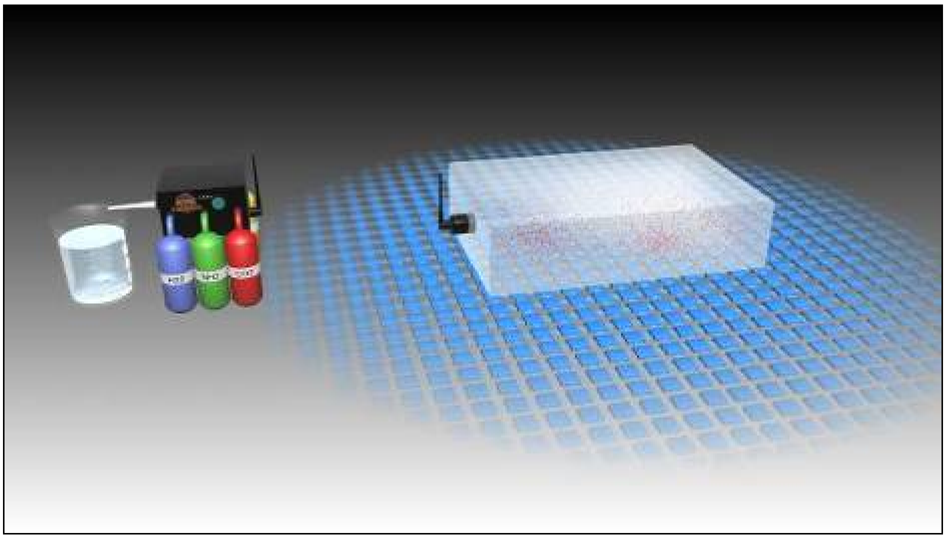


Figure 11 The conceptual cyberbiogenesis setup: on the right is the *in silico* molecular simulation space underlain and powered by numerous microprocessors; on the left is the molecular assembler and *in vitro* test beaker

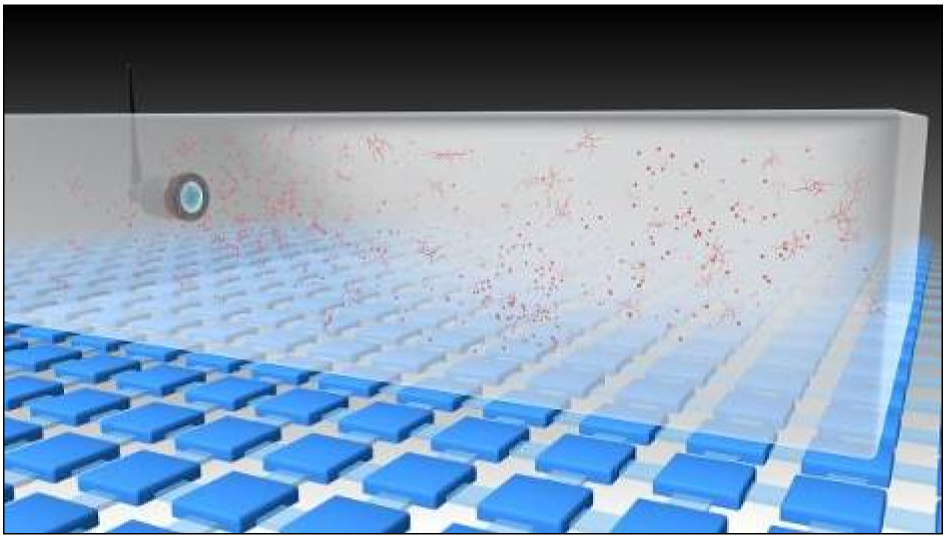


Figure 12 The simulation space is depicted rendering the physics of an aqueous chemical environment

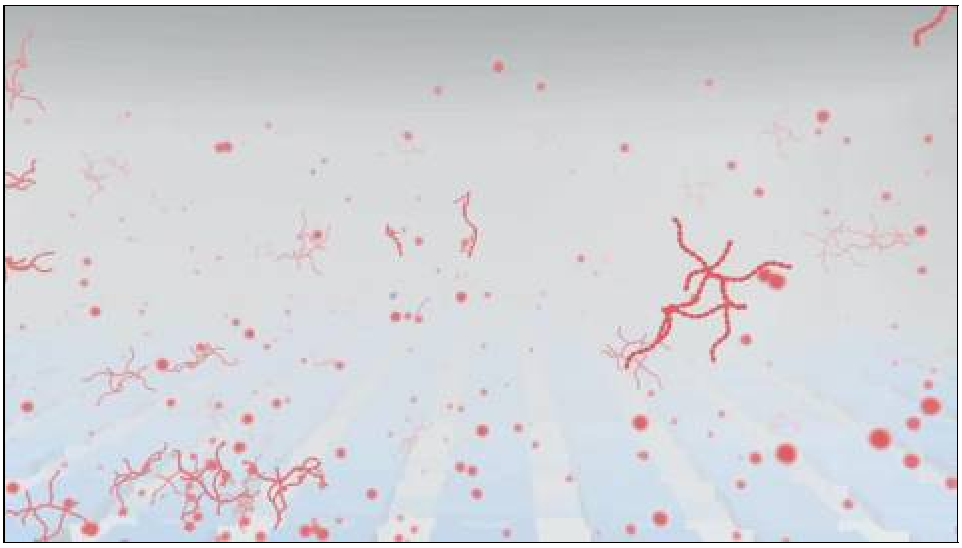


Figure 13 The formation of virtual molecules and some self organization occurs in the simulation space

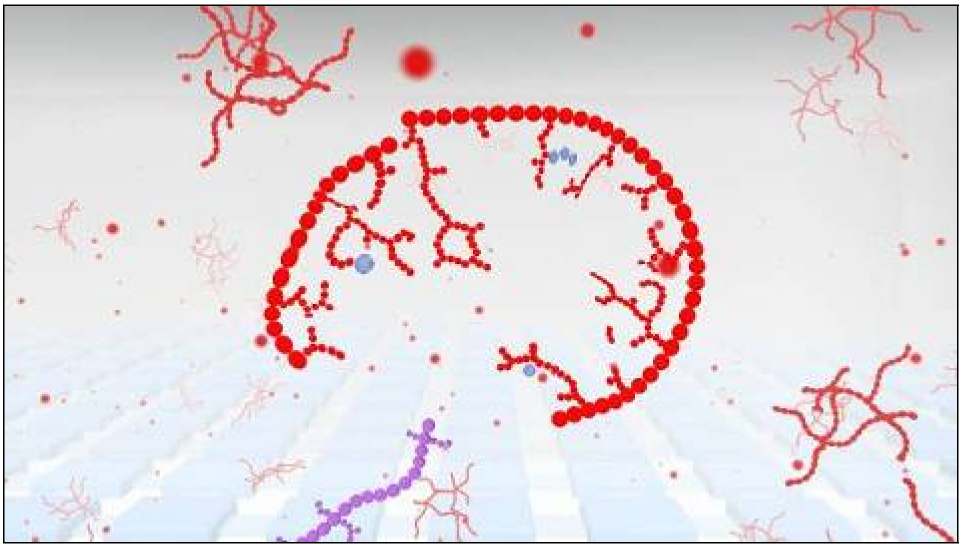


Figure 14 The formation of a vesicle is observed with the accidental capture of some other molecular machinery (on the lower left center)

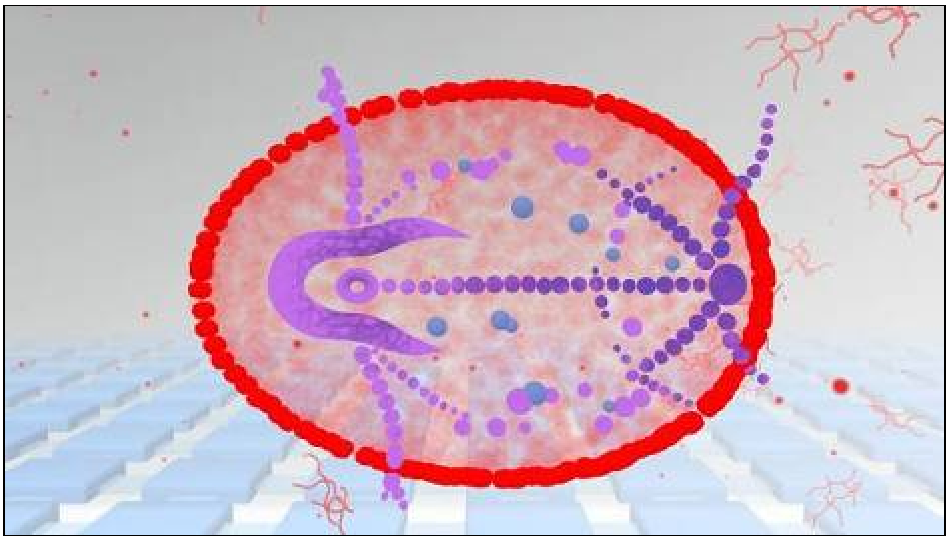


Figure 15 The accidental virtual symbiotic entity is capable of a sufficient ensemble of lifelike behaviors including compartmentalization, metabolism and replication with a mechanism for genetic heredity and so Darwinian natural selection has led to its growing sophistication

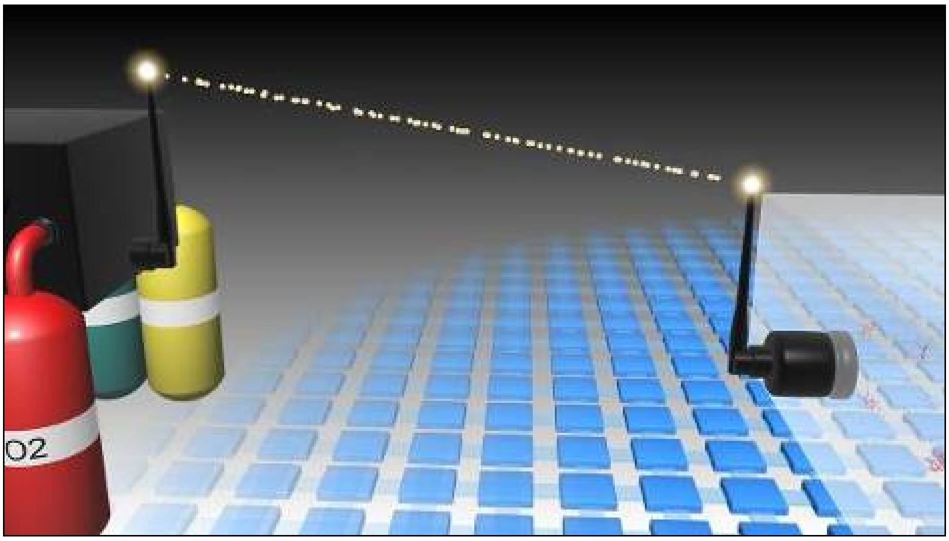


Figure 16 A sufficiently evolved entity is selected for digital decomposition and transmission from the *in silico* simulation to the molecular assembler

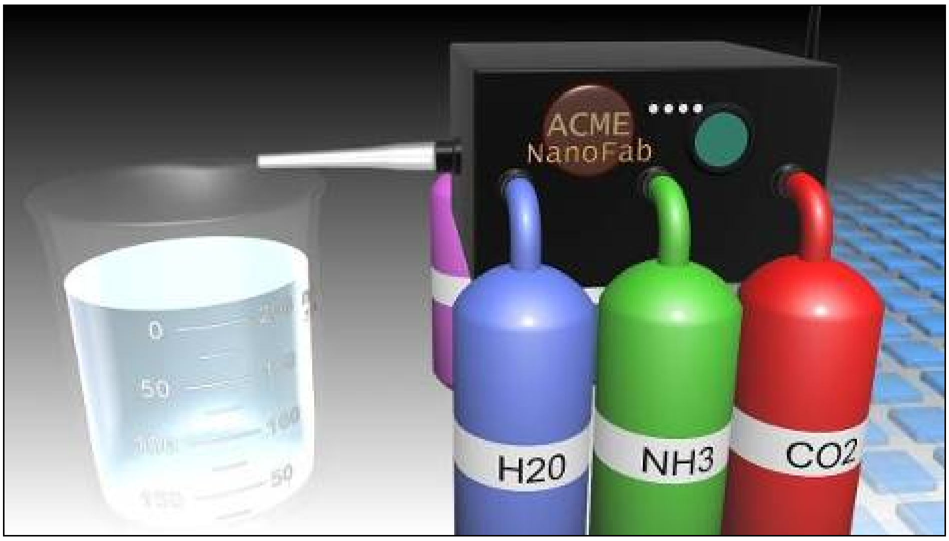


Figure 17 The hypothetical molecular assembler carries out a process akin to 3D printing and combines basic chemical elements to synthesize a molecular rendition of the virtual entity

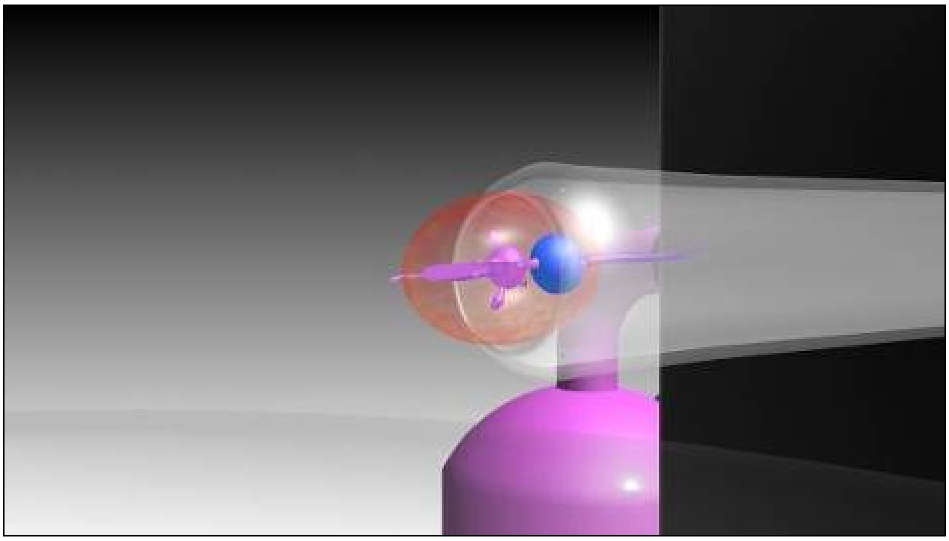


Figure 18 The fabricated entity emerges to drop into the beaker of formulated chemicals matching the environment in the original digital simulation

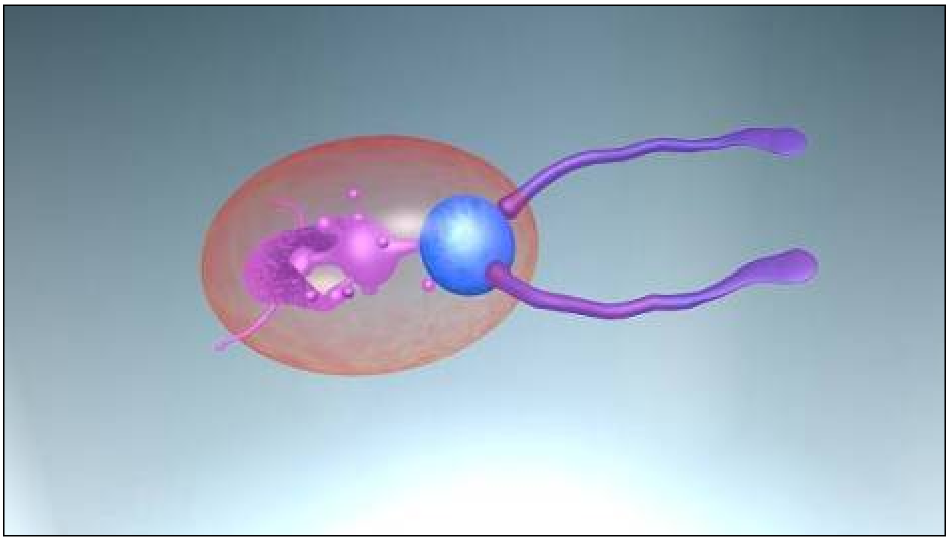


Figure 19 Within the *in vitro* environment, the molecular version of the entity starts to function as a new form of “living” entity

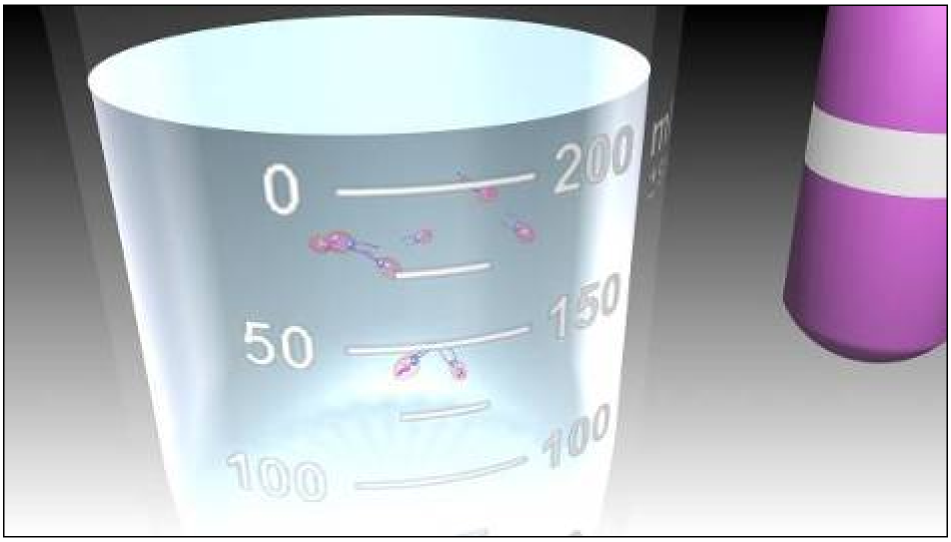


Figure 20 The entities advance further and begin to reproduce in their new environment, completing the cyberbiogenesis cycle

How realistic and realisable is this thought experiment? According to Nobel laureate Richard E. Smalley a “black box” nano-scale molecular assembler is nowhere near to becoming a reality (Baum, 2003). However recent progress in function representation (FRep) of 3D virtual objects (Pasko et al., 2008) paired with digital materialization made possible by universal desktop fabrication (Vilbrandt et al., 2008) shows promise in this direction. In either case, near term progress towards this goal still needs to be made in the domain of computational simulation. Given the scope, time and resources available at the beginning of this research project, it was thought that a reasonable goal for the work might be to produce a prototype that would reach the step depicted in Figure 13 above: a few simple molecules forming from an atomistic soup.

Research Methodologies and Methods to be Employed

Prior to launching into main discourse of the thesis it is worth clarifying the methods and methodology that are employed. In the map of the research topic we use the ontological assumption that it is possible to know the mechanisms of life arising from non-life. The epistemological framework for this work is based on a positivist approach utilizing a *hypothetico-deductive* method (Popper, 1959) to create a chemical model in software and then test it, observing from an objective distance using a methodology of controlled simulations and quantitative analysis (specifically pattern matching through time).

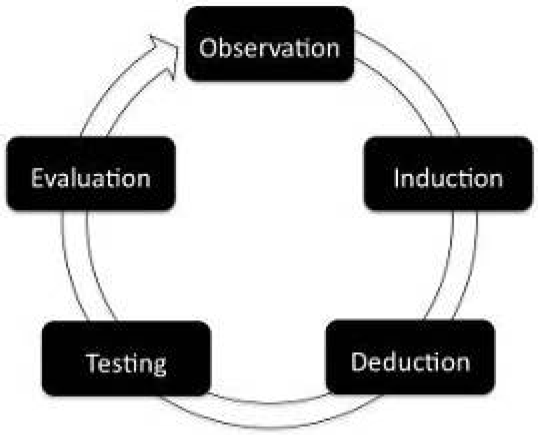


Figure 21 A visual representation of A.D. de Groot's empirical cycle

The Hypothetico-Deductive scheme traces its origins from natural philosophy and is effectively depicted in Figure 21 by A.D. de Groot’s empirical cycle (de Groot, 1969). In this reiterative cycle we start with observations followed by induction (inferring these observations within the concept of an emerging theory), leading directly to deduction (in which the researcher proposes a hypothesis), followed by testing and evaluation (success or failure of the hypothesis) and a re-visiting of the original observed phenomena (hopefully with a new light of understanding).

This research project utilizes quantitative rather than qualitative research methods. Quantitative methods will search for causal links between phenomena, relationships between independent and dependent variables. A primary goal is to produce results which are generalize-able and reliable, i.e.: its results should be independently reproducible with a guarantee that a significant bias did not enter into the results.

Bias could be introduced due to differential sampling of the running simulation. Sampling would change the nature of computation in the simulation, and regular sampling might create a pattern of behaviour that is outside the allowable "physics" of the space. In digital simulations sampling is often accomplished through bulk scanning of large datasets for pattern matching throughout the population of objects. In order to reduce the chance of bias, controlled sampling of periodic offline snapshots or "dumps" from the live simulation can be done, such that the simulation computing space is not engaged directly by the analyzers. This approach should also permit independent testing by collaborating institutions and support the comparison of duplicated simulation spaces. The approach will be discussed in more depth in chapter 2.

Numerical arrays, two dimensional graphs, tree representations and three dimensional visualizations will be used to present results from the directed searches. Variables might include relative patterns through time (patterns of reactions), behaviours (movement of a consistent set of objects from one location to another), or the creation or destruction of objects (population growth or decline). Independent variables will include time or spatial coordinates; dependent variables will include the density and velocity of objects (in this case simulated molecules). One particularly interesting dependent variable might be the level of energy (collective motion representing heat) in the system versus observed emergent phenomena. There might, for example, be an inverse relationship between heat and the formation of molecular bonds, i.e. after some point at higher temperatures fewer bonds might form. Management of heat within molecular dynamics simulations would then be identified as an example of the effect of a global property.

The project employs statistical analysis methods and packages analyzing test runs using computing grids inspired by UC Berkeley's BOINC network (Anderson, 2004). The project hosts a collaborative web site at www.evogrid.org employing a Wiki used for the research journal, documentation, access to the simulation code and executables, data and analysis, research findings, and advisory and supervisory commentary. A key background resource to this project is the Biota Podcast, an online audio discussion forum hosted by Tom Barbalet (Barbalet, 2008) which during the period 2006-2010 served as an important public sounding board for the development of these ideas. The full listing of Biota Podcast episodes relevant to this research is provided in Appendix B.3. Another background resource for this endeavour is a personal recollection of the origins of the EvoGrid provided by the author in Appendix A.1. Also of interest is a September 2009 article on the project which appeared in the New York Times (Markoff, 2009) reproduced in Appendix A.4. These materials are good background sources informing the next section.

Map of Personal Research Background

Figure 22 maps the research and development background of the author and is useful in understanding the origins of the research concept and proposed goals, methods and methodologies:

1. Work on various computer software systems for research and commercial applications.
2. A practice in the graphic arts informed visualization of the problem, and the design and functioning of the architecture.
3. Development of graphical user interfaces (Damer, 2001) when they were transitioning from research workstations at Xerox to the personal computer.
4. Years of study of biological systems including hosting the Digital Biota conference series (Damer, 1995).
5. A practice of virtual worlds research and development in the 1990s (Damer, 1997, Damer, 1996, Damer et al., 2000, Damer, 2003b, Damer, 2010).
6. The Nerves platform (Damer and Furmanski, 2005) built by the author and designed for biological simulation (for more background see the *Nerve Garden* book chapter in Appendix A.2).

7. The Digital Spaces 3D virtual worlds simulation platform (Damer et al., 2006) funded by NASA and providing prior experience in building simulation frameworks (Farkin and Damer, 2005).
8. The hypothesis for the research work developed in parallel and together with the tools.
9. The combined research experience combined to form the basis for the building of the prototype EvoGrid (Damer et al., 2011b, Damer et al., 2010, Markoff, 2009).
10. Simulation and networking experience informed the testing of the hypothesis.

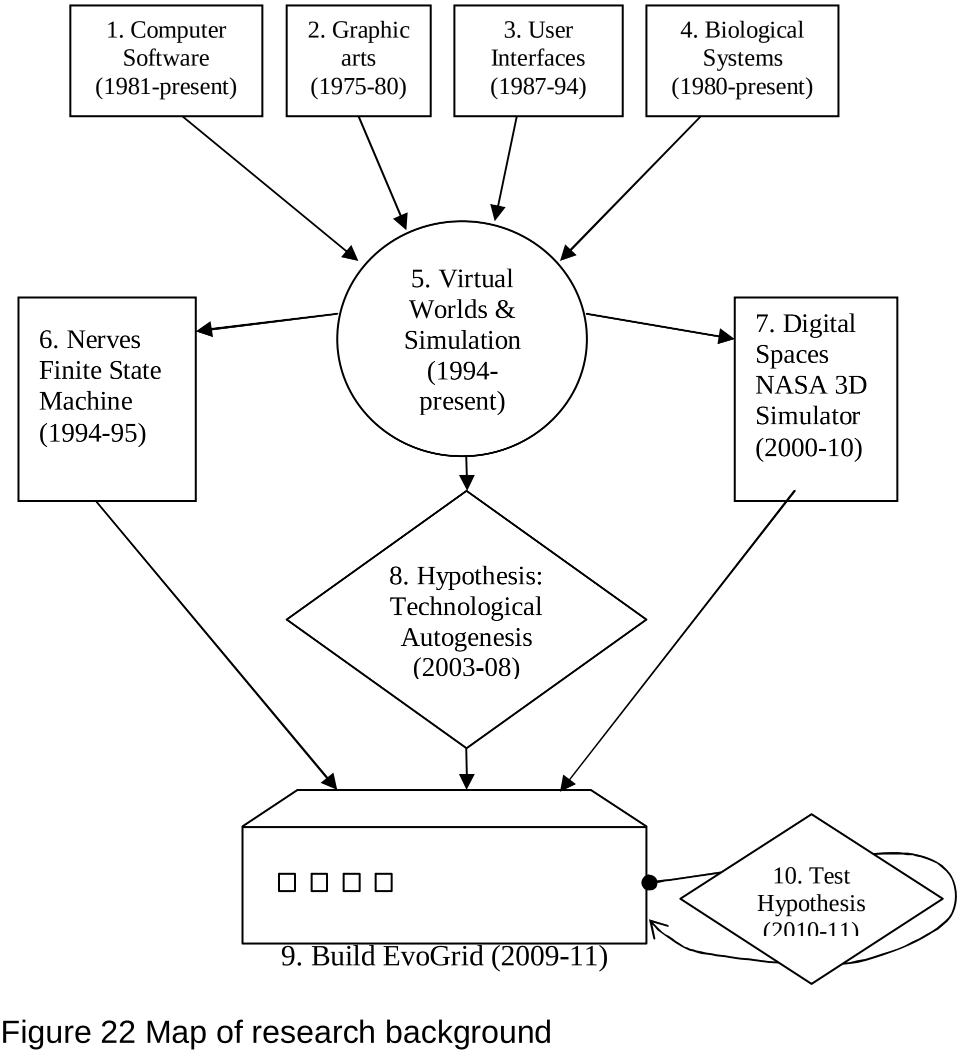

1. Computer Software (1981-present)
2. Graphic arts (1975-80)
3. User Interfaces (1987-94)
4. Biological Systems (1980-present)
5. Virtual Worlds & Simulation (1994-present)
6. Nerves Finite State Machine (1994-95)
7. Digital Spaces NASA 3D Simulator (2000-10)
8. Hypothesis: Technological Autogenesis (2003-08)
9. Build EvoGrid (2009-11)
10. Test Hypothesis (2010-11)

Figure 22 Map of research background

Thesis Roadmap and Contributions to Knowledge

The following four chapters of the thesis are organized to present a body of literature, analysis and conclusions which constitute the following contributions to knowledge:

Chapter 1: Framing the Challenge of Computational Origins of Life Endeavours – *where we will state the hypothesis of this work and provide a literature review of the centrally cognate fields that inform cyberbiogenesis and computational origins of life endeavours. We will conclude with a contribution to knowledge by illustrating a unique mapping among these fields.*

Chapter 2: Design for the EvoGrid Simulation Framework and Optimizations – *where we engage in a deductive approach to arrive at a viable computing architecture, optimization techniques and good practices for computational origin of life simulations and conclude with the contribution to knowledge of one design for the EvoGrid.*

Chapter 3: The EvoGrid Prototypes: Implementation, Testing and Analysis – *where we describe the construction of two versions of the prototype EvoGrid framework and illustrate through testing the computational cost savings and gains in emergent complexity that inform applications of this technique. The contribution to knowledge will be to address our hypothesis and determine whether the implemented framework and optimizations perform as predicted and whether they could be beneficial to future computational origins of life endeavours*.

Chapter 4: Limitations, Roadmap, Open Questions and Broader Considerations for Endeavours Seeking to Compute Life’s Origins – *where we present conclusions based on our testing of the hypothesis to provide another contribution to knowledge in the form of a technical roadmap and a series of open questions for emerging endeavours involved in computing life’s origins. We also list and describe scientific, philosophical, religious, and ethical conundrums posed by this line of research.*

Chapter 1: Framing of the Challenge of Computational Origins of Life Endeavours

Introduction

In their seminal paper Open Problems in Artificial Life (Bedau et al., 2000) the authors set a challenge in the second open problem to “achieve the transition to life in an artificial chemistry *in silico*” (p. 364) while also identifying that “[b]etter algorithms and understanding may well accelerate progress… [and] combinations of… simulations… would be more powerful than any single simulation approach” (pp. 367-68). These authors also point out that while the digital medium is very different from molecular biology, it “has considerable scope to vary the type of ‘physics’ underlying the evolutionary process” and that this would permit us to “unlock the full potential of evolution in digital media” (p. 369).

As we concluded in the previous section, a fruitful way forward for the Alife and origins of life fields might be to adopt the goal of building a class of simulations at some level of chemical reality. This branch of simulation, which we term *cyberbiogenesis,* would seek to study complex, emergent phenomena not in abstract universes as epitomized by the work of Barricelli, Stahl, Conway, Langton, Ray, or Sims, but with universes modeled as closely as possible on physical chemistry. Once armed with the tools of chemical simulation, experiments could then be set up to model scenarios the might lead to living molecular structures arising from their non-living forebears. This could then be considered the technical basis for the emerging simulation field termed: Computational Origins of Life (COoL) endeavours (Shenhav and Lancet, 2004).

There already exists a sub-branch of simulation science called artificial chemistries (AChems). (Dittrich et al., 2001) defined an AChem as “a triple (S,R,A) where S is the set of all possible molecules, R is a set of collision rules and A is an algorithm describing the domain and how the rules are applied to the molecules inside (the physics)." Current state-of-the-art AChems implement solutions from techniques employing abstract cellular automata to the simulation of chemistry at the quantum level. Recently there has been a rapid growth in projects utilizing the intermediate technique of *molecular dynamics* (MD) utilizing large, centralized, general-purpose computer clusters or purpose-built hardware such as Anton, an MD supercomputer (Shaw and Dror, 2008). Simulating benchmark experiments comprising thousands of atoms on a *commodity cluster* produces a number of nanoseconds of real-time equivalent chemistry per day (Bowers et al., 2006). Optimized software running on dedicated systems like Anton can deliver milliseconds of real-time equivalent chemical behavior for these same benchmarks in just a few weeks of computation (Shaw, 2009), a full two full orders of magnitudes of improved performance. Computational investigation into early stages of the origin of life will involve simulating millions or billions of atoms in supramolecular complexes over biologically significant time frames of seconds, minutes, hours or even days. However, there is no understanding of how long the transition from non-living to living molecular systems to life took, so any time estimates are pure speculation.

To meet these substantial challenges, proposals to unify efforts into larger COoL endeavours have been brought forth in recent years. In (Shenhav and Lancet, 2004) the authors proposed utilizing the Graded Autocatalysis Replication Domain (GARD) statistical chemistry framework (Segre and Lancet, 1999). These authors have developed a hybrid scheme merging MD with stochastic chemistry. In GARD many short MD computations would be conducted to compute rate parameters or constraints for subsequent stochastic simulations. Thus, a federation of simulations and services was conceived which would also involve interplay with *in vitro* experiments. It is this vision for unifying efforts in COoL that has inspired this work in which we propose a framework for distributing and searching a large number of small chemistry simulation experiments.

As stated by Shenhav and Lancet, “the prebiotic milieu could best be characterized by a dense network of weak interactions among relatively small molecules” (p. 182). Simulating a soup of millions of atoms represents a scale of complexity beyond even the ambitious targets set by the builders of Anton. While simulating an entire plausible pathway to life *in silico* seems like a journey of a thousand miles, the first few steps can still be undertaken with some confidence. Innovations in distributed simulation architectures supporting AChems and employment of optimization techniques may be of value to any COoL endeavour and are the focus of this thesis. We posit that techniques employed in other fields of computer science, including distributed computing, search, branching and inheritance over volumes of simulation data, hill-climbing, and backtracking can be applied to the simulation of chemistries and produce significant computing savings and experimental flexibility. We believe that such a marriage of methods can yield valuable tools to tackle challenging problems in the science of life’s origins, particularly in exploring emergent phenomena. More on why we believe that these are the approaches that should be considered will be presented in this chapter and in Chapter 2 of this thesis.

1.1 Hypothesis

With the above historical and contextual framing in place, we are now ready to state the hypothesis engaged by this work:

Hypothesis

> *Distributed processing and global optimization employing search coupled with stochastic hill climbing can produce significant performance improvements in the generation of emergent phenomena within small volume, short time frame molecular dynamics simulations over non-optimized solutions*.

Benefits to Science

> *A method and platform for optimizing computation to select for pathways leading to de novo emergent structures and processes in simulated chemistry could benefit future systems supporting cyberbiogenesis computational origins of life endeavours.*

To test this hypothesis, a prototype system will be designed, implemented and run through a series of trials with control simulations in place. The above hypothesis can be falsified or validated within the context of such an experimental implementation by comparing the execution of the simulations with and without employing the optimization techniques. The strengths of the resulting system will be enumerated alongside the limitations of the approach and an indication of pathways for improvement. The lessons learned can then be enumerated along with a map of considerations for any future project to engage in serious simulations supporting investigations of life's origins. With the hypothesis stated, the balance of this chapter will consist of a literature review that will inform a design exercise for the prototype, its initial implementation, testing, and the analysis of results which allow us to test our hypothesis and suggest a road map for future efforts.

1.2 Literature Review of Cognate Fields

The literature on the "origin of life" is vast, with over eight hundred books having this term in their titles (Damer, 2011). Thus the literature review must necessarily focus tightly on a few subject areas relevant to the hypothesis. We therefore set the following end goal: explore and draw from the cognate fields those insights that inform the design of computational systems which could take up the mantle of MD simulations in which emergent phenomena might be observed. It is a system supporting emergent phenomena that would form the basis for later endeavours attempting to deal with the challenges of simulating an origin of life.

To meet the above goal, this thesis draws together three primary cognate fields in the following order of priority of treatment and interrelationship:

1. *Complexity Systems Science* which yields guidance in the implementation of techniques to optimize for the emergence of phenomena within artificial chemistry simulations;
2. *The Prior Art and Current Practices of the Computational Simulation of Chemistry* ranging from abstract universes to the high fidelity physics of molecular dynamics;
3. *Past and Current Approaches to Parallel and Distributed Simulations of Artificial Chemistries and Molecular Dynamics* which inform topological design considerations of optimizations within simulation networks which are the tools proposed for use in the testing of this hypothesis.

1.2.1 Complexity Systems Science

It has been a goal of the young field of complexity systems research to apply itself to real world problems, but also to develop a comprehensive mathematical model that would explain such phenomena as self-organization. Melanie Mitchell writes in her book *Complexity, a Guided Tour* (Mitchell, 2009):

> In my view complex systems science is branching off in two separate directions. Along one branch, ideas and tools from complexity research will be refined and applied in an increasingly wide variety of specific areas… physics, biology, epidemiology, sociology, political science, and computer science, among others... The second branch, more

> controversial, is to view all of these fields from a higher level, so as to pursue explanatory and predictive mathematical theories that make commonalities among complex systems more rigorous, and that can describe and predict emergent phenomena (p. 301).

It may well be that complex systems realized through computer simulation lie in between Mitchell's two branches, applying some mathematical constructions and theories but seeking correspondence with real phenomena. Seeking emergent phenomena within statistically simulated chemical systems might well qualify as one of these in-between systems. Mitchell and others suggest that for complexity systems science a base of theory might not be up to the task of completely prescribing emergent complexity and self-organization. The simulation of chemistries as proposed in our hypothesis puts this effort solidly in the realm of complex systems. This section will examine the current state of complexity science through the thoughts of three complexity theorists and its bearing on the goal of this thesis.

Tamas Vicsek

Hungarian biophysicist Tamas Vicsek (Vicsek, 2002) wrote an essay in *Nature* titled "Complexity: The Bigger Picture" in which he observed (p. 131):

> In the past, mankind has learned to understand reality through simplification and analysis. Some important simple systems are successful idealizations or primitive models of particular real situations, for example, a perfect sphere rolling down on an absolutely smooth slope in vacuum. This is the world of Newtonian mechanics, and involves ignoring a huge number of simultaneously acting other factors. Although it might sometimes not matter if details such as the billions of atoms dancing inside the sphere's material are ignored, in other cases reductionism may lead to incorrect conclusions. In complex systems, we accept that processes occurring simultaneously on different scales or levels matter, and the intricate behaviour of the whole system depends on its units in a non-trivial way. Here, the description of the behaviour of the whole system requires a qualitatively new theory, because the laws describing its behaviour are qualitatively different from those describing its units.

Vicsek has therefore concluded that complex systems require a new theory, but he also goes on to suggest the following:

> What we are witnessing in this context is a change of paradigm in attempts to understand our world as we realize that the laws of the whole cannot be deduced by digging deeper into the details. In a way, this change has been invoked by development of instruments. Traditionally, improved microscopes or bigger telescopes are built to understand better particular problems. But computers have allowed new ways of learning. By directly modelling a system made of many units, one can observe, manipulate and understand the behaviour of the whole system much better than before, as in networks of model neurons and virtual auctions by intelligent agents, for example. In this sense, a computer is a tool improving not our sight (as in the microscope or telescope), but our insight into mechanisms within a complex system. Further, use of computers to store, generate and analyse huge databases [allows us to discover] fingerprints of systems that people otherwise could not comprehend.

Therefore, in the absence of the qualitatively new theory, Vicsek directs that a meaningful next step in the field of complexity studies is to develop digital modeling and simulation of complex systems from which this theory might one day emerge. A key underpinning for the effort undertaken in this thesis is that computers can “store, generate and analyse huge databases” that “people otherwise could not comprehend”. This new capability that is the generation and automated analysis of large datasets representing models of complex systems is one potential future source of a full theory of self-organization within complex systems science. Despite this promise the computational cost of simulating most systems in nature puts them outside of the reach of our computational machinery despite sixty years of solid progress. Therefore we suggest that to emulate complex phenomena in nature we must start by simulating very small natural systems.

Anand Rangarajan

During the process of designing and executing the first version of the EvoGrid, a meeting took place with University of Central Florida at Gainesville complexity researcher Professor Anand Rangarajan (Rangarajan and Damer, 2010). In the paraphrased transcription of a personal communication during this meeting, Rangarajan notes the following:

> For a large scale goal, for example, the formation of lipid molecules, the way you are approaching it is hill climbing on a fitness landscape. Engaging in stochastic hill climbing ostensibly in search of a particular goal in which you are going to truncate paths is a type of teleology which encourages self-organizing phenomena. By teleology I mean that you have to have some sense of the larger goal you want to reach and you are trying to see along the way if interesting things are happening. This is not how you would expect evolution would work. In your design you are also doing back tracking and branching. Branch and bound techniques for optimization discovered many years ago could bound problems because of mathematical knowledge of when a branching was no longer necessary. You cannot do this I suspect because your fitness landscape and optimization problem is probably too complicated. So my suspicion is you will be doing a type of simulated annealing or Markov chain Monte Carlo type of stochastic hill climbing where you have many paths that climb up the mountain but your objective function itself might also be changing underneath you. Perhaps some global parameters will need to be tuned, also from a teleological sense. What I then suspect is that you are going to get a set of unique paths that climb up this mountain. And along the way you are trying to look for emergent phenomena.

Rangarajan has pointed out a whole range of considerations about building digital systems to explore complex phenomena. The first is that the exercise of building an “exploration” system such as we are considering is in part a *teleological* exercise, or one involving an *end purpose*, from the Greek *τέλος, telos*, root: *τελε-* (Dictionary, 2011, Meriam-Webster, 2011). The scientific use of the term teleology is that we are building a system that we hope aids behavior to tend towards certain end conditions. There is a subtle balance to be achieved in the creation of any “artificial” universe. A researcher wants to be able to observe complex phenomena so sets up his or her universe to encourage the emergence of those phenomena. However, if proscriptive code is built to guide the universe on a deterministic pathway to that behavior then nothing can be learned about possible new, emergent phenomena. The design of systems like the EvoGrid is therefore an exercise employing a certain amount of *influence* to achieve a non-trivial amount of *emergence*.

Another point made is that the computing space, which for simulating chemistry would consist of thousands of objects rapidly interacting, is a very large fitness landscape. In other words, the number of pathways to what would be considered “maxima” or “optima” is very large. So if the formation of a particular molecule from this virtual soup takes tens of thousands of chance interactions, there is no way to mathematically model or predict when the particular interaction that forms the molecule will actually occur. There would also be no way to predict what configurations of whole populations or types (“species”) of molecules might occur, and when they might arise. Thus, such systems are unbounded in mathematical terms.

Rangarajan goes on to suggest that with the above constraints, one approach would be to employ hill-climbing using some kind of random process behind the choice of paths, such as a Monte Carlo method. He asks whether or not the underlying *fitness landscape* is also changing. In a simplistic chemical system where atoms form into molecules, contributing or removing atoms, breaking bonds or adjusting heat within the system, this landscape would in fact be changing.

In a subsequent conversation Rangarajan mentioned the work of (De Bonet et al., 1997) on the MIMIC system. MIMIC showed promise in the ability to correlate, or cluster pathways that find their way to maxima. Rangarajan suggested using this kind of method although we considered that it may be beyond the scope of this work. Rangarajan also brought up theoretical concepts from Seth Lloyd and others (Lloyd, 2006). He suggests bearing in mind Lloyd’s concepts of *logical depth* wherein the shortest (most compressed) program that can generate a phenomenon is sought. A long term goal of systems like the

EvoGrid would be to provide an experimental framework allowing these generative programs to be divined from the data. Rangarajan concludes with encouraging remarks echoing Vicsek. He calls for a move away from abstract models described by statistical mechanics and toward an experimental framework:

> We know that there is this world of abstract statistical mechanics but we need to build a computer architecture that is sustainable, that can run for ten years, that can use off the shelf components and provide a true experimental framework.

Stuart Kauffman

Renowned theoretical biology and complexity theorist Stuart Kauffman has perhaps done as much thinking and computational experimental work as anyone into questions of how emergent phenomena and self-organization emerge *de novo* from both natural and artificial systems. Kauffman has a strong focus on origin of life theory and takes the stance that life emerged from a spontaneously self-organizing network of chemical reactions which formed closed systems he terms *autocatalytic sets* (Kauffman, 1993). Catalysis in chemistry is the process wherein a molecule is involved in the formation or breaking of bonds of other, smaller molecules. From digestion to the formation of proteins, the very building blocks of cells, catalysts power living machinery. Kauffman proposes that the propensity of the universe is to self-organize. A review of the voluminous corpus of Kauffman’s work is beyond the scope of this thesis. However, there is a recent direction to his work relevant to the hypothesis presented by this thesis which we will discuss next.

In the interview with Rangarajan above, we were introduced to the concept of fitness landscapes. Let us now reexamine these landscapes with respect to their relevance to biology. In Charles Darwin’s travels around the world as a young man aboard the ship *The Beagle* he visited the Galapagos Islands off the coast of Ecuador. Darwin’s observation of populations of finches captivated him and years later helped him develop the theory of natural selection (Lack, 1940). We present a brief hypothetical version of the radiation of Darwin’s finch sub-species to illustrate concepts of a fitness landscape and of hill-climbing.

Galapagos Finches with large beaks became adapted to trees with particularly hard nuts on one of the small Galapagos Islands whereas finches on another island evolved longer beaks to better feed on prickly pear cactus. Each sub-species of finch had therefore achieved its own “local maxima” and was doing quite well in the niches of the fitness landscape of these islands. Should the nut shells become harder or the cactus become scarce, each type of finch would have to adapt again or fail to survive. A finch with a longer, strong beak might well be able to survive on hard nuts as well as cactus and become prevalent on both of the islands. In this case the genes of that finch would have found a way forward up the “hill” of a higher peak of fitness. Those finches which were still perched on their lower fitness peaks would face imminent danger of extinction. This example illustrates the interplay between the information system (the genes) and the environment, both of which are ever-changing. The “algorithm” traversing this landscape is the actual bird, the phenotype expressed by its genotype.

In his book *At Home in the Universe* (Kauffman, 1995, p. 248) Kauffman gives us a complexity theorist’s view into the nature of fitness landscapes and the entities that reside on them:

> …on random landscapes, local hill-climbing soon becomes trapped on local peaks far from the global optimum. Therefore, finding the global peak or one of a few excellent peaks is a completely intractable problem. One would have to search the entire space to be sure of success. Such problems are known as *NP*-hard… No search procedure can guarantee locating the global peak in an *NP*-hard problem in less time than that required to search the entire space of possibilities… the real task is to search out the excellent peaks and track them as the landscape deforms.

Any computer simulation involving many interacting objects could be seen as an embodiment of a fitness landscape. With a large number of objects able to interact in a number of complex ways, such as atoms forming molecules, this landscape would be, in Kauffman's term, *rugged*. A rugged landscape would have many peaks and valleys, with potentially a few excellent (highly optimal) peaks present, or perhaps a globally "best" peak (see B in Figure 23). Note that the term K in the figure represents Kauffman's richness of epistatic couplings (Kauffman, 1995p. 183). A search procedure that hopes to track emergent structures or behaviors across a sea of these peaks could be cast as an algorithmic entity attempting to find local maxima. To avoid the pitfalls of *NP*-hard spaces and make progress on these kinds of problems within a reasonable time frame, care must be taken in the design of the entire system.

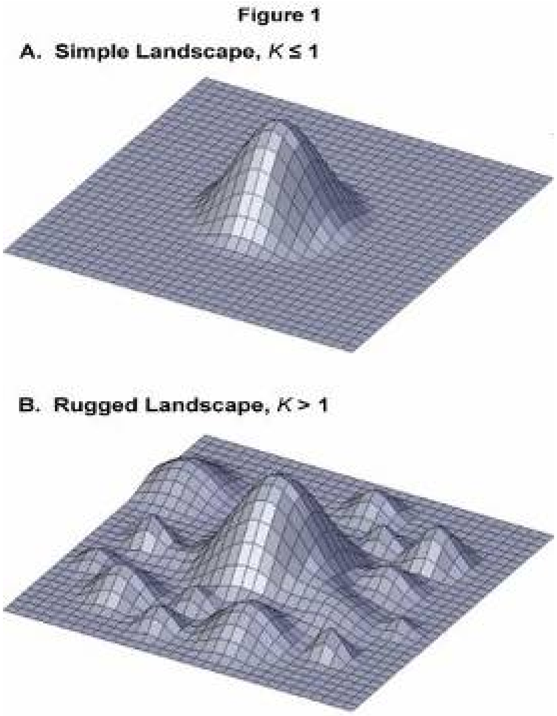


Figure 23 Simple and Rugged fitness landscapes as defined by Kauffman (Corman, 2011)

Over several decades, Kauffman and his collaborators pioneered the simulation of such sets utilizing a computably tractable method called Random Binary Networks (RBNs). RBNs are interconnected nodes that pass messages to each other, causing state changes (to either an on or off state). RBNs have been used to show that complex phenomena emerge and self-organize spontaneously within loosely coupled networks of a large number of relatively simple objects. In recent years, Kauffman has begun to move beyond RBNs and determined that the simulation of chemical reaction networks shows great promise to illustrate another concept: the Adjacent Possible (Kauffman and Damer, 2011b):

> I have this dream that one could really show on a "hypopopulated" vast reaction graph that fluctuations did NOT die out, but advanced into different Adjacent Possibles.

What Kauffman is referring to is a theory of the arising of acausally-driven novel organizational phenomena which he calls the Adjacent Possible (Kauffman, 2000). Relating to the concept of selection on fitness landscapes, the adjacent possible may be briefly summarized as follows: the net effect of beneficial mutations that permit clustering around maxima then lowers the probability of regression back down into less optimal states. In a kind of *ratcheting* process Kauffman explains that the “ever enlarging space of possibilities [expands] into an ever larger adjacent possible” (p. 42). From this arises Kauffman’s feeling that “I can sense a fourth law of thermodynamics for self-constructing systems of autonomous agents” (pp. 42-43). It should be pointed out that Kauffman’s views remain controversial within the scientific community (Mitchell, 2009, p. 286). However, despite this, his approach represents a substantial school of thought within complexity science, which as yet has no one central theory. It is for this reason that we utilize Kauffman’s conceptual scaffolding to support a part of the value proposition of the approach taken by the EvoGrid.

In a dialogue with Kauffman, the themes and research of this thesis work were shared. Subsequently the author was invited by Kauffman to join a working group being established at the CERN physics laboratory in Geneva, Switzerland to build systems (Kauffman and Damer, 2011ab). Kauffman introduced the author and the project to this group as follows:

> …Bruce Damer, who is working on chemical reaction dynamics on vast chemical reaction graphs with only small amounts of matter on them, i.e. hypopopulated by matter reaction graphs. My own bet is that here fluctuations do not damp out and we get non-ergodic behavior within lifetime of [this] universe. Bruce, welcome to the group. Your work bears on Origin of Life in terms of reaction dynamics in the messy world of prebiotic chemistry from which messy life emerged.

To understand the references here, let us review Kauffman's recent proposal of an experimental simulation designed to test his theory of the adjacent possible. This testing setup would involve a computing environment in which the six most common types of atom in organic chemistry are simulated using computably tractable "quantum-lite" bond interactions. The formation of compounds would then be represented through the standard chemists' tool of *reaction graphs*. Kauffman specifies this system in detail:

> How will our reaction system behave? Actually, no one knows. We have a vast reaction graph and a tiny amount of matter "on" it, it is "hypo-populated". No one has studied such systems. In this toy - real world, we get to place atoms and specific kinds of organic molecules in our reaction system. Now the mathematical way to study such a classical reaction system is by what is called a "chemical master equation". This fancy sounding thing is just this: For any distribution of atoms and molecules on the reaction graph, write down all the single next possible reactions that can happen. Then a neat algorithm, the Gillespie algorithm, simulates the behavior of this system by choosing at in a biased random way, depending upon the number of copies of each kind of molecule, which of the reactions occur and when it occurs. Then one repeats this zillions of times, burning up lots of computer time. The atoms and chemicals "flow" stochastically - that is, non-*deterministically - across the reaction graph. ((Kauffman, 2010)*

What Kauffman then proposes is that in a thermodynamically closed environment, the arising of unique patterns of molecules, which then predetermine further unique assemblages in an acausal (non-deterministic) manner, would accumulate and that the system would never reverse and run back down to equilibrium (“the fluctuations did NOT [emphasis by Kauffman] die out, but advanced into different Adjacent Possibles”). If the experiment showed this ratcheting property, it could imply that the universe itself is predisposed to promote the arising of ever more improbable outcomes and that this effect is not reversible. This would be a fundamental new result that has implications for the understanding of the origin of life as well as for the arising of all other complex organized phenomena throughout the history of the universe. What is exciting about Kauffman’s ideas, and they are at present just ideas, is that the abiogenesis of living systems is a natural outcome of a functioning yet uncharacterized “law” operating constantly in the universe.

This communication from Kauffman came late in the process of this research, long after the prototype was built and while it was in its final month of experimental trials. As we shall see in Chapters Two and Three, his experimental setup eerily corresponds to the actual prototype system that was built. This late input will permit us to speculate somewhat more courageously about the importance to science of efforts like the EvoGrid and provide some clear objectives in our road map in the final chapter. One last point that is valuable to make emerged from the author’s meeting with Professor Freeman Dyson in Princeton, New Jersey, in early 2009. As discussed previously, Dyson pointed out that any simulation had to be very messy to represent the messiness of nature echoing Kauffman’s later communication that the EvoGrid as envisioned could indeed simulate “the messy world of prebiotic chemistry from which messy life emerged”.

Summary

Concepts from key thinkers and open questions in complexity systems science provide important insights into the framing of the design of a system like the EvoGrid. Concepts of fitness landscapes and the algorithms that move about on them form the basis for understanding the process of optimizing the occurrence of emergent phenomena. The EvoGrid could be seen as an experimental system to permit testing of theories in complexity systems science. The validation of the value of the EvoGrid effort by key researchers in complexity systems science provides additional motivation around the value to science and contribution to knowledge of the work.

1.2.2 Prior Art and Current Practices of the Computational Simulation of Chemistry

Simulation science and its associated technological realizations in computer architecture are a continuous quest to move abstract models and computer codes ever closer to usefulness in understanding and predicting structure and behaviour in the physical world. Eugene Wigner wrote in 1960 in his article ‘The Unreasonable Effectiveness of Mathematics in the Natural Sciences’ that “… the enormous usefulness of mathematics in the natural sciences is something bordering on the mysterious and… there is no rational explanation for it” (Wigner, 1960). Today, aircraft, automobile and buildings are very completely simulated prior to being built and tested. The power of simulation is only now being applied to chemical processes. This section will briefly review historical approaches to the problem and the current state of the art.

Environments on the surface of planets are bounded by and made up of solids and gaseous phases of matter. At one level, however, they consist of different densities of bonded and free atoms in constant motion. Underlying this basic picture of reality lies yet more complex interaction and entanglement of subatomic particles: the longer range forces of electromagnetism, gravity and the not-so-subtle physical effects of temperature and pressure gradients. One of the key questions builders of simulations must ponder is: *at what level to simulate?*

1.2.2.1 The History and Current State of Chemical Simulation

Early efforts to simulate the interaction of large numbers of particles were often limited to the most mathematically and computably tractable data structures of the day. Matrix algebra consisting of the manipulation of vectors and its implementation in low scale integrated circuits provided a major boost to early supercomputing efforts. The first purpose-built vector-based supercomputers designed by Seymour Cray (Damer, 2003a) were early systems capable of running artificial chemistries. Figure 24 shows an example of one of these systems from the author's private collection.

Figure 24 Cray-1S Supercomputer from the author's private collection

While the Cray supercomputers focused their particle simulations toward large science problems of the day, such as: modelling thermonuclear explosions, climate, or aerodynamics, a related approach helped initiate the simulation of chemistry.

1.2.2.2 Lattice and Cellular Automata Techniques

Also based on the ease of manipulating numerical arrays was the lattice gas and cellular automata approach as described by (Hasslacher, 1987, p. 175-217). Figure 25 shows the conceptual framework of a three dimensional cubic lattice on the left with particle motion and nearest neighbour encounters between particles occurring through “nodes” depicted on the right.

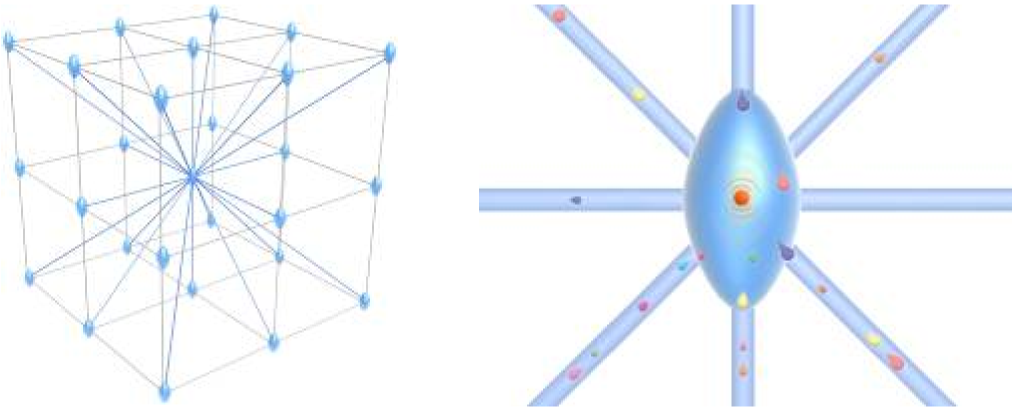


Figure 25 Concept of a Gas Lattice simulation

(Fredkin, 2004) proposed the twelve edge-centred directions but most lattice gas implementations use eighteen (face and edge centred). Applications of lattice models for artificial chemistries go back to von Neumann (von Neumann and Burks, 1966) who sought to design a self-replicating machine known today as a cellular automata, or CA. It was von Neumann who first linked CAs and simulation to a key property observed in biology, that of self-replication. As previously discussed, it was Barricelli who built the first biologically-inspired simulation on von Neumann’s digital computer at Princeton (Barricelli, 1953). In the 1960s Richard Gordon (Gordon, 1966, Gordon, 1967, Gordon, 1968b, Gordon, 1980) pioneered the use of lattice computation in the simulation of chemical activities, writing that “The internal structure of digital computers is particularly well suited for representing lattices and therefore the physical or chemical adsorption of atoms or molecules onto lattices” (Gordon, 1968a).

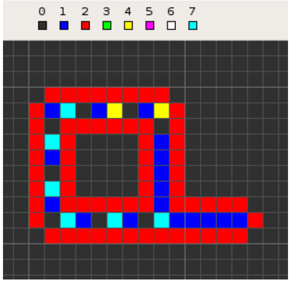


Figure 26 Langton Loop generated by the Golly CA program

CAs are organized typically as a two or three dimensional grid of finite state machines linked locally to their nearest neighbours. Finite state machines can be simple (bits turned on or off or a set of symbols) or they can be complex (reaction graphs as in Kauffman's random binary networks). The time evolution of the states in each machine and how the machines affect their neighbours determine the overall state of the space of the CA. Chris Langton developed his well-known "self-replicating loops" in the early 1980s (Langton, 1984). This was an eight state CA of 86 cells and able to self-replicate in 151 steps. Figure 26 shows a "Langton Loop" generated by the widely used Golly CA program (Trevorrow and Rokicki, 2011). The typical structure of a CA is evident here with eight types of state shown active within this particular CA.

1.2.2.3 Tuple-based Systems and Dissipative Particle Dynamics

As computing power increased in the 1980s and 1990s it became viable to move away from fixed grids of locations in space as epitomized by gas lattice and CA systems and move toward *tuple* systems, wherein particles were assigned a set of three coordinates and able to assume any location in space. An early example of a computably tractable tuple-based artificial chemistry came with the development of dissipative particle dynamics (DPD) presented in 1992 (Hoogerbrugge and Koelman, 1992) as a method for simulating complex fluids at the mesoscopic level. In the origins of life field, (Fellermann, 2009b) and (Solé et al., 2009) have adapted DPD work in coarse-graining and scaling in DPD and produced some early computer models of protocell replication, notably the self-replication of lipid aggregates. Fellermann writes:

> …DPD are instances of coarse-grained modeling techniques in which the spatial structure of molecules is represented explicitly, though not in full atomistic detail. Instead, groups of atoms within a molecule are lumped together into point particles, usually called beads. These beads are then connected by elastic springs to form the whole molecule. Small molecules such as water are considered to be combined into a single bead by groups of 3 to 5 molecules.

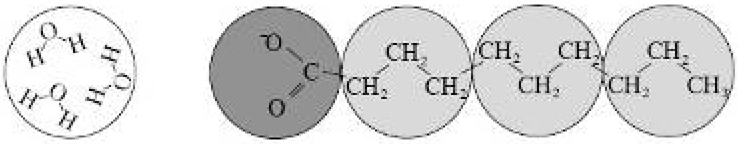


Figure 27 Example coarse-grained representation of water (on left) and decanoic acid (a fatty acid surfactant) on the right

Figure 27 from (Fellermann, 2009a, p. 25) illustrates the method of coarse graining: three types of beads represent a grouping of three water molecules, the hydrophobic tail of the surfactant and the hydrophilic carboxyle group. DPD methods have been successful in generating mesoscale simulations such as the metabolism and fission of a nanocell, which will be covered in Chapter 4 of this thesis. Before moving on to the next technique, it is worth stating why coarse-grained methods are favoured for some applications, especially when involving macromolecular structures as described here: computational tractability. Fellermann writes that “the time scale of MD is an expected three orders of magnitude slower than DPD (Groot and Warren, 1997)!”

1.2.2.4 Molecular Dynamics

Molecular dynamics (MD) simulations are employed to model the motions of molecular systems at an atomic level of detail. MD systems are used to simulate proteins, cell membranes, DNA and other cellular processes (Shaw and Dror, 2008, p. 91). The key computational cost of MD is that in modelling the random, heat-driven motion within a gaseous or aqueous environment every atom is interacting with a large proportion of the other atoms within a single second. From Shaw (p. 91):

> Millisecond-scale simulations of a biomolecular system containing tens of thousands of atoms will in practice require that the forces exerted by all atoms on all other atoms be calculated in just a few microseconds—a process that must be repeated on the order of 10^{12} times. These requirements far exceed the current capabilities of even the most powerful commodity clusters or general-purpose scientific supercomputers.

(Andrews and Bray, 2004) provide a lucid description of the challenge of all atomistic simulation systems such as MD:

> Using an intuitive picture of chemical reaction systems, each molecule is treated as a point-like particle that diffuses freely in three-dimensional space. When a pair of reactive molecules collide[s], such as an enzyme and its substrate, a reaction occurs and the simulated reactants are replaced by products. Achieving accurate bimolecular reaction kinetics is surprisingly difficult, requiring a careful consideration of reaction processes that are often overlooked. This includes whether the rate of a reaction is at steady-state and the probability that multiple reaction products collide with each other to yield a back reaction. Inputs to the simulation are experimental reaction rates, diffusion coefficients and the simulation time step. From these are calculated the simulation parameters, including the 'binding radius' and the 'unbinding radius', where the former defines the separation for a molecular collision and the latter is the initial separation between a pair of reaction products.

It is important to note that every set of codes developed to solve the above problem uses different models. Many share common attributes such as the heat, or kinetic energy of the system (motion), and reactions (formation or breaking of bonds) which relate to the concept of binding and unbinding radii and collision energies. Some MD codes do not explicitly deal with chemical reactions but concentrate on the geometries of pre-built molecules.

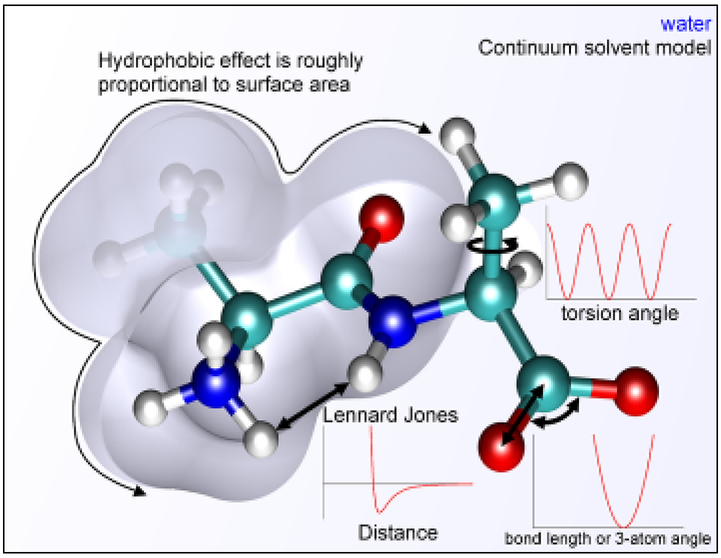

water
Continuum solvent model
Hydrophobic effect is roughly proportional to surface area
torsion angle
Lennard Jones
Distance
bond length or 3-atom angle

Figure 28 An alanine dipeptide molecule used to illustrate physics on this scale: atoms are modeled as charged spheres connected by springs which maintain bond lengths and angles while interacting with each other via Coulomb's law simulated through a molecular mechanics potential energy function.

As is illustrated in Figure 28 above (diagram courtesy Wikimedia Commons (Edboas, 2011)), if chemical reactions are supported, then their codes must deal with a steady stream of reaction products (molecular assemblages) which then express complex internal interactions (Lennard-Jones forces, for example) and external interactions (the hydrophobic effect, for example). A full elucidation of the operation of MD simulations or artificial chemistries is beyond the scope of this thesis. To a great extent we are treating these MD code frameworks as “black boxes” within the central emphasis of this work: distributed computing and optimization methods. It is, however, valuable to consider some of the leading MD codes available, as this will factor into our particular choice of “black box”.

ESPResSO

There are a number of packages built under open source (GNU General Publishing License-GPL) which support MD. We are reviewing these packages as they are all candidates for this research since they do not require commercial licensing fees. The first is ESPResSo

(Extensible Simulation Package for Research on Soft matter), which is a feature-rich package supporting MD as well as DPD developed by (Limbach et al., 2006). ESPResSo supports an external scripting language (Tcl) which allows the framework to be extended and reactions added. As we are considering each of these MD simulation systems as “black boxes” with inputs and outputs for the purposes of distributing them within a framework, we will not go into the internal details and capabilities of each.

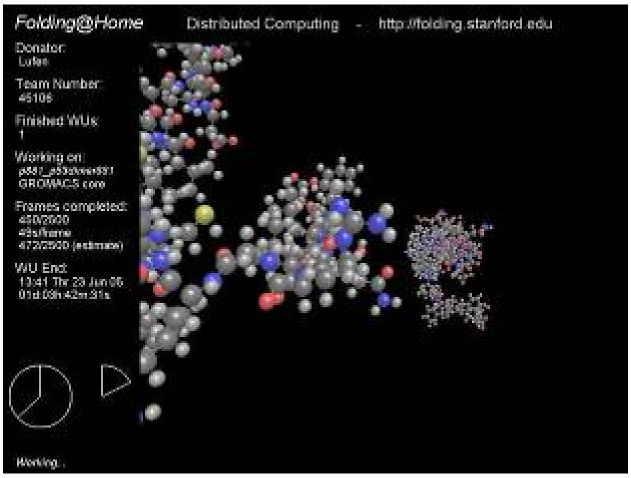
LAMMPS

The second framework we will consider is LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator) an efficient and feature-rich MD package that also has support for DPD (Plimpton, 1995). LAMMPS is designed for single processor desktop computers but also for multiple processor implementation (MPI) and operation over a network. LAMMPS is extensible via C++ application programming interfaces (APIs).

NAMD

The third framework we will examine is NAMD (Not just Another Molecular Dynamics program). (Phillips et al., 2005, p. 2) give an introduction to NAMD stating that a decade ago the platform “permitted simulation of a protein-DNA complex encompassing 36,000 atoms (Kosztin et al., 1997)“ whereas more recent releases “permitted the simulation of a protein-DNA complex of 314,000 atoms (Villa et al., 2005)”. As NAMD is an example of a system designed to be run on large commodity PC clusters we will treat it more in the next section.

GROMACS



Folding@Home Distributed Computing - http://folding.stanford.edu

Donator:
Lufen

Team Number:
45106

Finished WUs:
1

Working on:
p881_p53dimer881
GROMACS core

Frames completed:
460/2500
49s/frame
472/2500 (estimate)

WU End:
13:41 Thr 23 Jun 05
01d:03h:42m:31s

Working..

Figure 29 GROMACS visualization from the Folding@Home project (credit, team member Lufen, Folding@Home)

GROMACS (Groningen Machine for Chemical Simulations) is the last GNU-licensed MD package we will consider and is the code we selected for the implementation of the EvoGrid prototype. GROMACS was first developed in Professor Herman Berendsen's group in the department of Biophysical Chemistry of Groningen University (van der Spoel et al., 2005). GROMACS is consistently rated as the fastest MD simulation system running on commodity computers (van der Spoel, 2011), which was a major consideration for this project. GROMACS also has a previous history as an engine behind distributed computing based on its use in the Folding@Home project (see Figure 29) which will be discussed in the next section.

1.2.2.5 The Scale Continuum in Artificial Chemistries

As we can see there is a continuum of AChems ranging from the abstract and more computationally tractable to higher fidelity vis-à-vis chemistry yet much more computationally intensive. There is another continuum: scale. Sometimes referred to as multi-scale or multi-physics, this is the challenge of simulating at different levels of granularity. Obviously the coarse-graining of dissipative particle dynamics discussed earlier in this section is one example: simulating solvents (water in this case) by aggregating water molecules may serve some utility at one level, but simulating each individual water molecule may be required at another level, and at another level individual water molecules are forever exchanging atoms with one another by making and breaking hydrogen bonds. Several approaches to the problem of simulating AChems at multiple levels have been proposed (Hickinbotham et al., 2010) and mechanisms to automatically move between levels also postulated (Nellis and Stepney, 2010). While the emergence of coarser scales may occur within our simulations the explicit handling of multiple scales is beyond the scope of this work.

Summary

In conclusion, as we can see by the substantial investments being made and significant early results, among all of the approaches to simulate molecular interactions, MD has emerged as an early predictive tool in chemistry. The utility of MD has been established through innovative use of dedicated, parallel supercomputers and through distributed commodity computing grids. For our formulation of an experimental platform to support experiments in emergent phenomena in complexity systems using AChems, it is suggestive that we employ MD simulation in some kind of parallel or distributed computing environment. We will take up such environments next.

1.2.3 Past and Current Approaches to Parallel and Distributed Simulations of Artificial Chemistries and Molecular Dynamics

In the years since the launch of the Cray supercomputer in the mid 1970s the support of scientific simulation has undergone a transformation from dedicated machines like the Cray to large-scale networks of small computers. More recently we have seen the return of dedicated supercomputers. This section reviews a number of current hardware solutions and distributed computing topology solutions to the problems of chemical simulation.

A major challenge in simulating chemistry at the MD level is the sheer amount of computation required for very small populations of atoms and time durations (in nanoseconds up to microseconds). Therefore in this early phase of the maturing of the use of MD tools, parallel and distributed computing and optimization techniques will and are playing an important role to the early efficacy of these tools. From (Kalé et al., 1999) the authors argue for the case for parallel and distributed computing of MD because “the time scale of [ten thousand to one hundred thousand atoms interacting] requires that one simulate their behavior in time steps as small as 1 fs (10^{-15} s)… [and that] the number of computational steps required to complete relevant simulations is prohibitive on any single-processor computer” (p. 284).

Several projects including FASTRUN (Fine et al., 1991), MDGRAPE (Taiji et al., 2003), and MD Engine (Toyoda et al., 1999) each have produced special-purpose hardware to support the acceleration of the most computationally expensive stages of an MD simulation. The Anton supercomputer (Shaw, 2009) mentioned previously is producing the most dramatic performance improvements in MD simulations to date achieving from microseconds up to one millisecond of chemical simulation time of a virtual system of over ten thousand atoms making up a small protein enveloped by water molecules. The MD simulation was able to be carried out long enough to observe the protein folding and unfolding multiple times. The structure of the folded protein was then tested with real physical analogues *in vitro* and new properties were experimentally verified (Shaw et al., 2010). This work by Shaw et al. has established that the closing of the loop between computer simulation and chemical validation which we proposed in our definition of *cyberbiogenesis* is now practical, if expensive.

1.2.3.1 Commodity Cluster Simulation using NAMD

NAMD2 (Kalé et al., 1999) is a second release of the popular NAMD code. This release has been optimized for clusters of commodity computers and its previous version was reported (Phillips et al., 2005, p. 23) to have run “a 310,000 atom system on 256 Itanium processors of the NCSA TeraGrid system with an average production speed of 2.5 ns per day.” Proposals to simulate millions to a billion atoms, which would be required to model portions of a cell are being considered (Nakano et al., 2007). As we saw above from performance reported by Shaw et al., 2.5 ns per day is a full two orders of magnitude slower than the dedicated Anton supercomputer, however, Anton was simulating two orders of magnitude fewer atoms. In some sense this is not simply and apples and oranges comparison as the needs for simulation differ depending on the problem being addressed.

1.2.3.2 Distributed Network Simulation in Folding@Home

Distributed computing of seemingly intractable scientific problems has experienced a renaissance in the past decade. Beginning with the use of large collections of networked personal computers to break encryption mechanisms, searches through radio astronomy data in projects such as SETI@Home and the emergence of the BOINC network (Anderson, 2004) has all generated an enormous capacity to computationally solve classes of scientific problems. The networks supported by the BOINC network consist of home computers running screen saver software, and populations of university computers, often during off-hours. The combined distributed computing power of BOINC projects sometimes exceeds the computational capacity of many of the largest supercomputer clusters.

As we have seen, the digital simulation of atoms and molecules on networks of commodity computers is beginning to emerge as a tool for MD. A leading effort in this area is the Folding@home project based at Stanford University (Pande et al., 2003). Derived from the BOINC methodology Folding@home utilizes a fully distributed network of computing resources that includes college and home computers and even game consoles such as the Sony Playstation 3®. Folding@home employs the molecular dynamics code GROMACS. As is detailed later in this thesis, GROMACS was the platform adopted for this research. Folding@home has had significant success through the simulating of the interaction of the molecules of surrounding solvent with the ribbon like structure of protein nucleotides of predicting the folding geometries of proteins of up to several thousand atoms (Pande, 2011). In other words, Folding@home uses global search mechanisms to discover pathways to energy state minimization for these large molecules.

Figure 30 Early Folding@home network in 2002, source: (Jones, 2003a).

Folding@home's network topology (Figure 30) supports a large number of sequential simulations of a relatively small volume and is the best existing proof of concept that a system like the one we are proposing is viable.

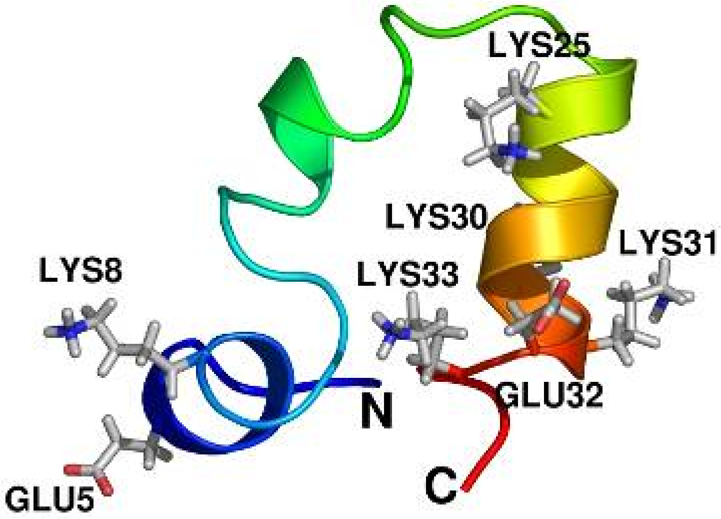


Figure 31 Villin headpiece protein (simulation model)

An example of a typical simulation target for Folding@home is illustrated in Figure 31. Michael Levitt and Erik Lindahl of the Department of Structural Biology, Stanford University School of Medicine describe the computational complexity involved (Jones, 2003b):

> The Villin headpiece is a very small protein consisting of about 600 atoms. However, the cell is always surrounded by water (red/white rods), which brings the atom count up to about 10,000. Every single atom interacts with the closest 100–200 neighbors; the interactions have to be calculated every single step, and then we repeat this for a half-billion steps to generate a microsecond of simulation data. The data for [Figure 31] was generated from a two-week run on ten of Iceberg's nodes.

1.2.3.3 The Challenging Path from In Silico to In Vitro

Due to the intractability of chemical simulation of more than a few tiny volumes and over time scales beyond a few microseconds, a significant roadblock exists to the witnessing of the emergence of complex structures of behaviors commonly found in nature or the laboratory. It is therefore suggestive that the only doorway into this world is to operate a large number of small simulations each for short periods of time. We propose that, despite their diminutive nature, such simulations can produce interesting results and inform larger future efforts. Such a computing challenge can be met by the innovative computational topologies of dedicated parallel supercomputers and by distributed networks of commodity machines.

The nascent MD simulation industry has already adopted some of the more widely used techniques for applying computational power to problems, notably parallel and distributed computing. However, for a class of problems in chemistry where insight into the *ab initio* emergence of structures and behaviours is sought, additional optimization techniques can be brought to bear. This is particularly applicable in the case of the investigation of models of life's origins focused on the emergence of structures (vesicles, informational molecules, and others) and behaviours (metabolic cycles, repair and replication and others) both independently and in combination.

The predicting and verification of emergent phenomena in the study of the chemical origins of life are both experimentally challenging to "design", control and observe in the "wet" or "bench" chemistry laboratory. As suggested earlier, since MD has proven itself as a tractable medium from which predictive chemical knowledge can be obtained it is possible that such tools could be adapted to meet the challenge of modelling *emergent phenomena in abiotic chemistry*.

However promising the application of MD techniques to origin of life problems may appear, the corresponding technological challenges are truly daunting. The presupposed building blocks of life range from very small molecules to very large supramolecular assemblages such as amphiphilic bilayers containing hundreds of thousands to millions of atoms. In addition, the minimal operational volumes occupied by both molecules of interest and surrounding solvents (predominantly water) which would have to be simulated would contain millions of atoms, easily outstripping available computing resources. Finally, the timescales required of simulations to model even the most preliminary biologically interesting phenomena that might lead to the formation of minimal protocells vastly outstrip the powers of the largest computational grids.

Therefore by necessity the problem of simulation of the prebiotic medium must be broken down into very small segments indeed, where one is looking for an emergent phenomenon within tightly bounded molecular contents and timescales. One could conceive of a chain of such small *in silico* experiments, replicated *in vitro,* forming a linkage that would shed light on a larger chemical pathway leading from non-life to life. This is the long view of an undertaking significantly beyond the scope of this thesis, but it is valuable to have more ambitious objectives in mind to guide the first few steps taken. Chapter Four will feature a treatment of this long-term view and the open questions that are posed to any effort that tackles simulation of life's origins.

1.3 A Working Map of the Cognate Fields

One valuable result from the above survey of the literature is that we can see a logical pattern emerging that determines an order of consideration of knowledge informed by and informing cognate fields in a kind of *cascade*. Through this cascade of considerations we hope will emerge a design for systems whose goal is to meet the challenge of cyberbiogenesis. To explain and explore this cascade let us define some monikers for the cognate and related fields:

Primary cognate fields:

Complex Systems – Complexity Systems Science
AChems – Artificial Chemistries
SimTech – Simulation Technology
COoL$_{ER}$ Computational Origin of Life with Experimental Realization

Related cognate fields:

OoL - Origin of Life
AL - Artificial Life
Search – Search Heuristics and Methods
Optimize – Optimization Techniques
TheoBio – Theoretical Biology
SynthBio – Synthetic Biology

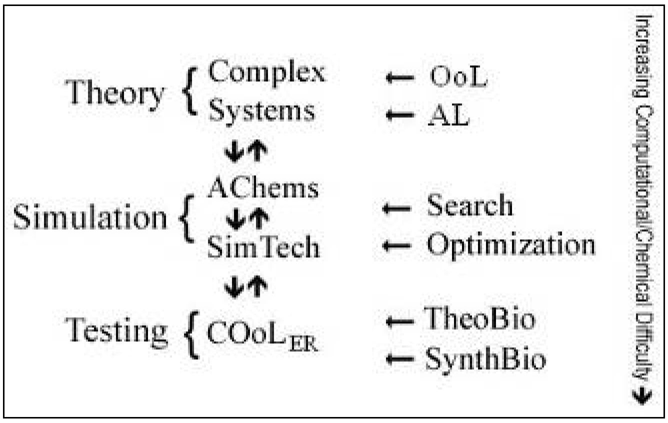


Figure 32 Map of cognate fields and how cyberbiogenesis systems may emerge from an interaction between these fields

Placing these major fields in a map (Figure 32) we start with *Theory* and the foundational cognate field of Complex Systems Science, which is informed both by the supporting fields of Origins of Life (OoL) and Artificial Life (ALife). Complex systems are the foundation field, however, as we are considering *ab initio* emergence of complex, self-organizing phenomena from a large field of interacting objects. The observation of self-assembly of macromolecular structures and gene-directed evolution in nature are a direct inspiration in the design of such Theory models. The Alife field of genetic algorithms mirrors natural behaviours in technology and provides some useful input to Theory models. Thus informed, our complex system theory will establish a concrete starting point for the next stage: the implementation of a prototype *Simulation*.

For the Simulation phase we place the Artificial Chemistries (AChems) and Simulation Technology (SimTech) together as peer cognate fields. As we saw in the above literature review, the expanding power yet persisting limitations of digital simulation technologies permit some progress to be made in artificial chemistries techniques such as CA, DPD and our focus: MD. There is a two-way dance between AChems and SimTech as each field advances. The supporting fields of search and optimization support the advancement of both AChems and SimTech. As Simulation problems can quickly enter computably intractable territory, these two supporting techniques will extend the range of their viability. As experimental systems are built in the Simulation stage, the Theory stage can also be revisited with what we hope might be the eventual development of an actual formal predictive model of understanding complex systems.

Simulations which can demonstrate the emergence of biologically relevant phenomena may next be carried on into the *Testing* stage. The next step would be the attempted validation of the computational origin of life (COoL) simulations through experimental realization (ER) in physical chemistry. This challenging phase would use input from theoretical biology (TheoBio) in the form of models of experiments which might include RNA-world, vesicle assembling or autocatalytic cycles per Kauffman. The technology for creating such laboratory bench experimental setups would be best informed by the field of synthetic biology (SynthBio). Clearly there would initially be challenges in duplicating the results in chemistry that were observed *in silico* so both the Simulation and Theory phases would have to be revisited and revamped.

Over time and through many years or decades of iteration this framework of interacting cognate fields would generate higher and higher fidelity pathways toward our vision of a complete cyberbiogenesis system expressed in our earlier thought experiment. Clearly, as indicated in Figure 32, the computational and chemical complexity increases as you go down through each of the three phases. The scope of this thesis effort lies therefore in the first two phases, some development of Theory leading to the implementation of a modest prototype in Simulation but not setting foot at all in Testing.

1.4 Objections to the Approach of Using Deterministic Digital Simulation in the Modeling of Natural Systems and Emergent Phenomena

University College London researcher Peter J. Bentley strongly argues that von Neumann architecture of most conventional computing systems is fundamentally unsuited to model and simulate living systems, stating that "natural systems and conventional computing [are] at opposite ends of a spectrum" (Bentley, 2009, p. 106, Bentley, 2007). Table 1 illustrates elements of Bentley's spectrum by listing the opposing properties of each environment. He goes on to propose a "a system of computation that has biological characteristics" and proposes a design architecture for *Systemic Computation* (p. 107) which would embody concepts of "parallelism, stochasticity and asynchrony".

Conventional	**Natural**
Deterministic	Stochastic
Synchronous	Asynchronous
Serial	Parallel
Heterostatic	Homeostatic
Batch	Continuous
Brittle	Robust
Fault intolerant	Fault tolerant
Human-reliant	Autonomous
Limited	Open-ended
Centralised	Distributed
Precise	Approximate
Isolated	Embodied
Linear causality	Circular causality
Simple	Complex

Table 1 Features of conventional vs. natural computation (courtesy Peter J. Bentley)

This is a highly promising approach which should be able to address the computational challenges of artificial chemistries. However, it remains in the research phase with no working hardware to assist in projects such as the EvoGrid. Computer systems such as Anton described previously are based on conventional von Neumann concepts but designed entirely around the task of molecular dynamics calculation using off the shelf microprocessor technology. It will be interesting to see how exotic, re-thought computing approaches fare against custom designed von Neumann solutions in the race to produce high fidelity simulations of natural systems.

Theoretical biologist David L. Abel argues that von Neumann computers are not just inefficient at simulating nature but are also inappropriate for this task, stating:

> Self-ordering phenomena arise spontaneously out of phase space, but we have no evidence whatsoever of formal organization arising spontaneously out of physical chaos or self-ordering phenomena. Chance and necessity has not been shown to generate the choice contingency required to program computational halting, algorithmic optimization, or sophisticated function. (Abel, 2009b, p. 273)

Abel builds on this to state that approaches to simulating abiogenesis or otherwise exploring living systems utilizing algorithmic techniques in computer software stand in stark contrast to the entirely *physicodynamic* properties of nature (i.e. chemistry). He states correctly that the chemical mechanisms of cellular metabolism and reproduction are driven by a clear form of symbolic program control, or “choice contingency”, through the programming selections of individual nucleotides (Abel, 2009a, Abel, 2010), codon sequencing, and gene regulation by peptides, polypeptides, proteins and mircroRNAs. (Abel, 2009b) He calls this divide between physicality and formalism the *Cybernetic Cut*, claiming that it is “perhaps the most fundamental divide of scientifically addressable reality” (p. 274). In Abel’s analysis all forms of Artificial Life programming (and any results gained from them) are highly suspect as naturalistic models of abiogenesis, since they sit solidly on the formal side of the Cybernetic Cut.

Abel would therefore object to the design and implementation of the EvoGrid prototype simply because it is based on computers and software and therefore is a formally-based, non-physical system. He would also hone in on and critique the algorithmically-expressed teleological mechanism of our search functions tracking and pruning branches in their quest to permit the emergence of phenomena in the simulation. In fact in a private communication with Abel on the primary optimization technique of this thesis he stated:

> No basis exists in quality science for blind belief in a relentless uphill climb towards the pinnacle of functional success. Physicodynamic determinism is blind and indifferent to formal function. An inanimate environment couldn't care less whether anything works, let alone works best. To purely physical interactions, non-function is just as good as function. Even if a pinnacle were ever stochastically reached, what motivation would exist to maintain an isolated function of any kind in a purely physicalistic [sic] reality. The only environmental preference is for a fully integrated, fully programmed, fully operational fittest phenotypic organism. How many hundreds of highly integrated metabolic pathways and cycles would be required for Mycoplasma, the simplest known organism, to come to life? (Abel, 2011)

It is difficult to argue with Abel's position. However, it is put forth that by adopting the tools of molecular dynamics simulation utilized by chemists, the EvoGrid takes one step through formal space toward the physicodynamic side of the Cybernetic Cut. Indeed, as molecular dynamics systems in software are already serving science as predictive tools for discovering behaviour in chemistry with gene folding being one prime example (Pande et al., 2003), it would seem to suggest that computer simulations can make predictive forays into emergent phenomena that may have played a role in the self-assembly of the first simple living systems. In summary, the EvoGrid seeks to travel some distance from the realm of programmatic, prescribed, and highly deterministic world of software and Alife systems by:

- Emulating physicodynamic natural processes not abstract, formulaic systems.
- Implementing an artificial teleological chemistry (AChem) based upon a system of a large number of encounters and associations of particles.
- Starting with an *ab initio* approach wherein there are no pre-built forms in order to study the properties of emergence within physicodynamic systems.
- Permitting a “loose” coupling between teleological search function and the state of the system by utilizing random selection.
- Employing molecular dynamics techniques which seek to converge on a high fidelity model of chemical systems, yielding results which may be physically verifiable or falsifiable by laboratory chemistry, permitting a cycle of fine-tuning of the virtual experiments.

1.5 Concluding Statement on the Scope of this Work

While this work is both *interdisciplinary* and *interstitial* lying between and drawing from the above cognate fields it is fundamentally a work of computer architecture. The author’s career background lies in the building of computer software systems and not in mathematics, chemistry or biology. No mathematical or chemical models will be proposed or treated, except to illustrate some basic concepts. Indeed, as the scope of the project is to show that the aforementioned techniques can be successfully applied to molecular dynamics and other artificial chemistry simulations, a highly naïve model of chemistry was utilized. The modelling of high fidelity chemical systems is currently the purview of large engineering teams equipped with sizeable hardware and budgets. However, we believe that we can show benefits from our optimization techniques and other advantages of our prototype architecture that could be adopted by future researchers to support full-scale chemical simulations.

To prevent confusion or misunderstanding, this preliminary work on a computational origin of life endeavour is *not*:

1. An origin of life simulation system;
2. a complete artificial chemistry or molecular dynamics system;
3. a viable test-bed for general complexity science;
4. or an artificial life system able to exhibit properties of artificial evolution.

Instead, the prototype to be designed and implemented *is*:

1. A software prototype employing open approaches and distributed computing utilizing a standard software component from the field of molecular dynamics;
2. a system designed to illustrate how optimization techniques including search and stochastic hill-climbing when applied to distributed computing can be applied to achieve the result of permitting emergent phenomena to arise within artificial chemistries while saving time and computational costs;
3. and an exemplar of the technological and epistemological conundrums facing origin of life researchers in building systems that could cross the chasm between abstract models and concrete chemical reality.

With the above introduction to the tools and challenges at hand we may now pare down the scope of the challenge undertaken by this work to a more reasonable level:

1. Design, construct and test a software simulation framework which deploys an industry standard molecular dynamics code engine to simulate small volumes of artificial chemistry over short time frames within a distributed computing framework.
2. Further, apply stochastic hill climbing optimization techniques to this framework which produces a flexible system to search for desired phenomena and build upon this to support branching of new volume simulations as well as back-tracking to restart from previously promising simulations.
3. Test and measure the performance of this system in a series of computational experiments. Compare a control case not using

optimization to the search-driven experiments. Estimate the reduction of computational time taken to reach a desired simulation state, and examine performance with respect to scaling the amount of computing resources available. The dual criteria for success are *the reduction of computational time together increased levels of observed complexity in the system over control experiments not employing these optimization techniques.*

4. Apply lessons learned from this work to guide future efforts to build systems to simulate scenarios in more realistic prebiotic chemical environments.

Summary

Now that the scope of work is sufficiently narrowed and the background cognate fields established, we can reiterate our belief in the intrinsic value of this exercise to science. We believe that the successful implementation of such a system may have the following value for complexity systems science: that the tools of chemical simulation may hold the keys to building a system capable of open ended complex emergence. Synthesizing the worlds of chemical simulation with their requisite richness in terms of simulated physics, with widely used distributed computational search and optimization techniques, may ultimately produce this result. In conclusion therefore we believe there is a value to proceeding to design, build and test the following system which exhibits the following computational cost-saving and observably emergent properties: *a distributed processing and global optimization through search coupled with stochastic hill climbing of emergent phenomena within small volume, short time frame molecular dynamics simulations*.

Chapter 2: Design for the EvoGrid Simulation Framework and Optimizations

Introduction

Given all the considerations from the cognate fields explored above, how would one design the computational aspect of a cyberbiogenesis system? In 2008 an initial design for a prototype of such a system was sketched out by the author. Over a series of months, this approach was presented for comment to members of the author's technical staff and to an international multidisciplinary group of advisors, who are all acknowledged in this thesis. The design underwent many improvements but the original insight held. This chapter will review the reasoning behind this design, describe it in detail, and compare it with similar systems built for other purposes.

To reiterate, we are aiming this design of a computer simulation implementation in the direction of endeavours regarding questions of the origin of life. The author realized that it would be valuable to engage in a second thought experiment and came up with a naïve model of the origins, operating principles and development of a simplified virtual universe in which life has arisen. While understanding the origins and evolution of the real universe is far beyond the scope of this work, the resulting naïve cosmology was found to be valuable underpinning for the philosophy of the simulation. We will therefore repeat it here, in abbreviated form.

In the exercise of the design of our computer simulation system we therefore take the liberty of *applying teleological goals that set the system up to find its way toward a state of increased associations*. These teleological goals are our *stand-in* for this observable effect in the universe. However, careful observation of the universe in the large and the small, as well as the process of evolution, is not able to detect any goal-setting (Dawkins, 1986).

Our teleological choices, which we call “search functions,” will be described later in this chapter. Their definition is not totally arbitrary even if we consider ourselves to be *artificial gods* and have free will to assign any whimsical property we would like the system to achieve. These search functions are informed by *prior knowledge* of the general properties of living systems and the likely steps and elements that life required to arise.

2.1 The Design of a Simulation System to Explore Origins of Life Questions

Any software prototype which purports to simulate the physics of chemistry is at some level itself highly naïve. After all, at some level, each of the virtual “atoms” or “molecules” is merely a list of numbers or symbols stored in some kind of abstract data structure and processes as though it were an object in reality. In molecular dynamics simulations many of the subtleties of the quantum mechanical level and behavior of electron orbitals are not represented. However, we can come up with a workable analog of the above naïve cosmology (and by extension and goal, of reality) by building a system to simulate random encounters between great numbers of virtual particles (notional atoms) in which a proportion of them form associations (virtual molecules). This is at least a good start and many competent artificial chemistries have been constructed around this assumption and have succeeded in guiding laboratory experiments (Pande, 2011, Shaw et al., 2010).

2.1.1 Design of a Generic Canonical Contiguous Volume In Silico Chemical Simulation

A second step in the design of our prototype framework requires us to make choices about what kind of volumes are being simulated. As we reviewed in Chapter 1, simulating a large single volume of molecular interaction poses both computational and organizational challenges. Boundary problems in distributing simulations of smaller parts of a larger contiguous volume across many computers tax these systems with extra communications overhead. This can be understood simply by picturing one atom traveling across the larger volume, all along the way it will transition between sub-volumes and a record of its location must be distributed rapidly and accurately between the computers handling these sub-volumes. Dedicated sub-system hardware was built into the Anton MD supercomputer to handle these boundary problems (Shaw and Dror, 2008).

Figure 33 depicts the conceptual form of a contiguous large volume simulation: perhaps a large contiguous space of millions (or more) virtual atoms driven by a large number of processors (in blue). Sitting outside are an equally (or greater) number of computers dedicated to analyzing parts of this simulation and applying the directed search technique. In this simplified scenario, interesting “areas” of the contiguous simulation space are selected for more computing resources.

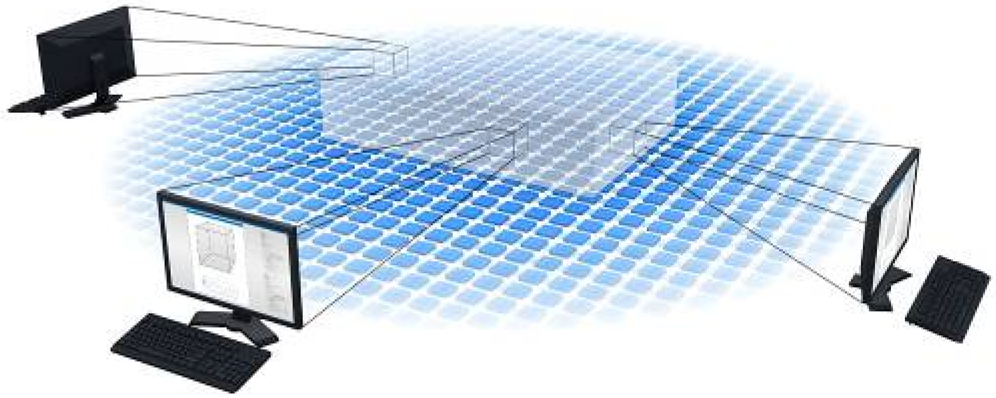


Figure 33 Conceptual view of artificial chemistry simulation grid

Figure 34 through Figure 37 depicts a high level conceptual view of the processing of molecular interactions within a single large volume. Figure 34 depicts how the volume would be broken up into cubic subvolumes and processed as a series of layered snapshots.

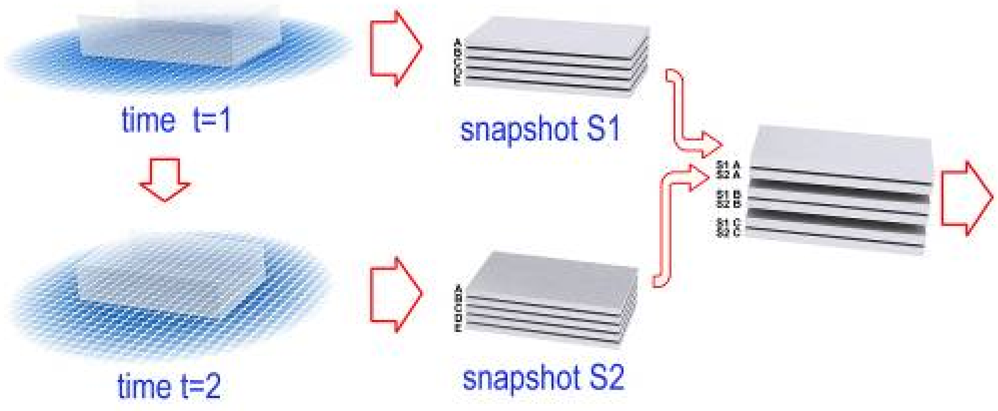


Figure 34 Conceptual view of layered approach to processing time snapshots

Figure 35 depicts how these cubic subvolumes we call frames might be distributed to analysis clients in a series of buckets representing queues. This concept of processing queues will be utilized later in the actual implementation of how the EvoGrid's simulation manager prioritizes and processes simulations (see section 2.3).

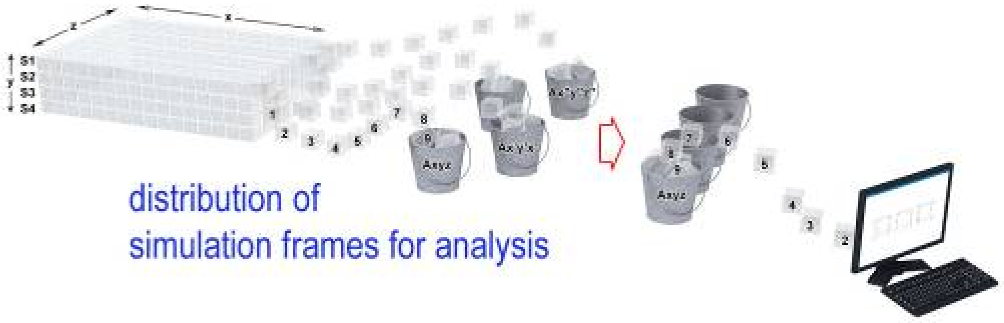


Figure 35 Conceptual view of distribution of snap shot sub-volumes to analysis grid

Figure 36 shows an overview of the complete conceptual cycle from analysis clients distributing results back to a simulation manager which then affects how the ongoing volume simulation is carried out.

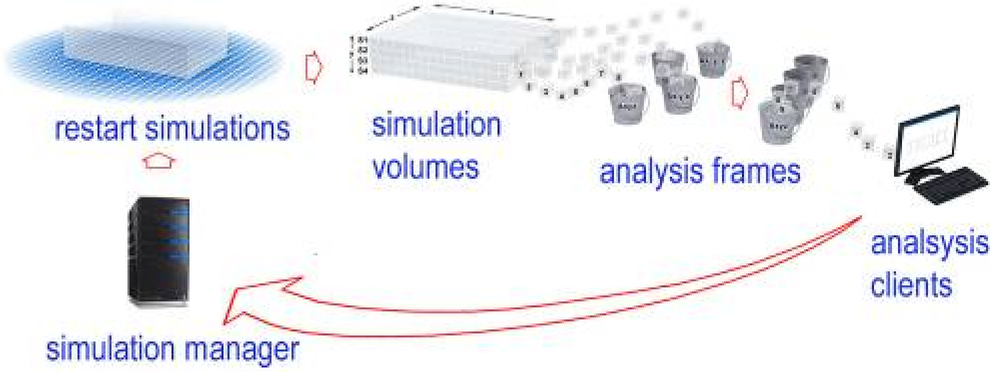


Figure 36 Conceptual view of feedback of distributed analyses to server

Figure 37 concludes our conceptual overview by illustrating the steps of analysis and search creating scores for individual frames that then inform the simulation manager.

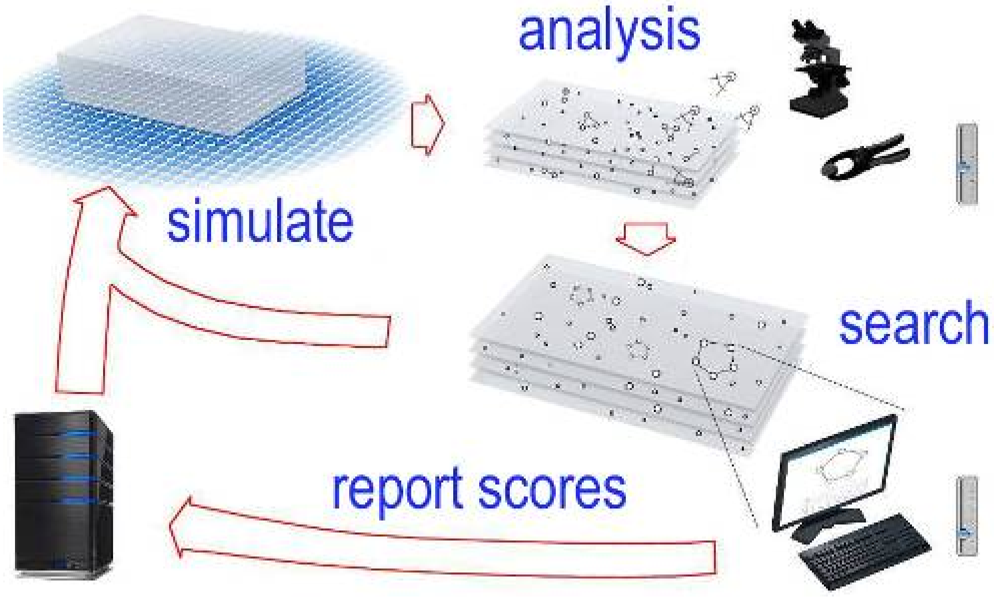


Figure 37 Conceptual view of weighted analyses based on observed phenomena

Due to a number of issues, not least among them the challenges of managing the communication between simulating grid machines assigned their own small subvolume this elegant high level view must rapidly give way to other architectural options.

We are therefore motivated to undertake the far more tractable challenge of simulating much smaller volumes, representing no more than one thousand atoms per unit volume. This design choice restricts the type and scale of molecular structures and processes that could arise within the small volumes. Even simple proteins found in living cells often are composed of several thousand atoms. Surrounding reactants and solvents would add several thousand atoms into the mix. As reported by (Shaw, 2009) meaningful simulations of such volumes are now possible and desirable for biochemistry. So there is promise with this approach, especially as computing power grows.

We must therefore give up all immediate hope of simulating, say, even a tiny fraction of the molecules making up a cell or any significant structure in a cell. For example, the derived estimation for number of carbon atoms alone in *E. coli*, a “relatively” simple prokaryotic cell, is ten raised to the tenth power, or 10,000,000,000 (ten billion) atoms as estimated in (Philips and Milo, 2009). By concentrating on small self contained volumes containing no more than a few thousand atoms we gain significant benefits. The key benefits are: the ability to more easily distribute computation of these volumes; and to apply search criteria to branch and backtrack processing of volumes whose full state may be stored for temporal and trend analysis.

Having chosen this design path we can apply searching and tree-branching processing of simulations to provide undirected (that is, directed only by random production and selection) formation of increasingly rich local environments. To return to our naïve model of the cosmos, we would design the simulation criteria to permit the emergence of enriched environments. This means an increasing propensity for small volumes of virtual chemistry to develop more associations or atom-atom bonds. In no way was it expected that the prototype considered here would come anywhere close to addressing the question of how the first plan-following mechanism or protocell might have arisen. This is left for future researchers and suggestions as to likely technical challenges are provided in the balance of this writing. What we hope to show with our first implementation of such a simulation platform is that it may be possible in principle to build such systems.

While there are a few well funded efforts to build special purpose computing hardware designed only to simulate molecular interaction (Shaw and Dror, 2008) our approach is to employ off-the-shelf von Neumann-type general purpose digital computers. Such grids of computers would be readily available for a modest effort like this. The use of distributed computers also fits well with the small volume approach.

2.1.2 Life in the World of Biomolecules Informing the Simulation Design

Beyond the simple simulation of the *content*, simply stated as the presence and movement of simulated atoms, the variety of atom-atom associations formed are the next consideration in design. Along with the traditional covalent and ionic bonds (where electrons are shared) there are number of other ways in which atoms and groups of atoms can associate, the so-called *molecular affinities* (Deamer, 2011, pp. 114-122). One such famous affinity is the hydrophobic interactions which loosely hold lipid together in the bi-layer of a cell membrane. These associations add an additional layer of computational complexity to each encounter between molecules and their member atoms.

The challenges of the representation of molecular shape and shape changing derive from molecular affinities. In recent years, significant progress has been achieved in creating simulations able to predict these shapes (Shaw et al., 2010). In addition and related to the shape that molecules take and then change when in solution is the challenge of modeling the surrounding solvents (in this case water, H_2O) which is arguably necessary (Luisi and Damer, 2009). Without the constant “bumping” of surrounding small molecules, molecules such as proteins, which are made up of a string of amino acids, would not take on their characteristic shapes. The shape of molecules is essential to their function as information containers or functional mechanical tools of molecular biology.

One last detail to point out is that within the relatively small, confined space of a single cell, there is a tremendous amount of dynamism. For example a sugar molecule situated at one end of the cell will be driven by thermodynamic forces into a state of constant motion and will travel within a few seconds all through the cytoplasm. During this frantic movement that molecule will encounter most other free-floating molecules and the boundaries of interior compartments which contain other organelles and functional parts of the cell (Goodsell, 2009). This dynamism guarantees that, eventually, reactions will happen where they are meant to happen without the individual molecules being carefully guided from one place to another. This point to point guiding as we would see in a human factory for instance, does not happen inside cells except in certain special cases such as the “walking” of large structures along microtubules of the cytoskeleton by motor proteins.

One could argue that despite the seeming complexity of these higher order behaviors, that faithfully simulating the atoms should permit these behaviors to simple emerge out of the low level interactions. However, as enumerated above, the sheer number of atoms and encounters and associations they are involve in, this kind of computation may be entirely intractable. Therefore many researchers argue for more coarse-grained simulation of molecular interactions in which, for example, a number of water molecules would be grouped together and simulated as one entity (Fellermann et al., 2007). We feel that the truth lies somewhere in between and that as is suggested in (Bedau et al., 2000), the useful simulation of biomolecules needs to flexibly employ low level fine grained representation for some purposes and switch seamlessly to coarse grained representations at other levels. This is often called the challenge of *multiscale multiphysics*.

We must also acknowledge that the *simulation of whole environments* is also important. For example, liquid water, whether it is in small surface ponds or at great depths near hydrothermal vents, is surrounded by complex substrates: the rock and sand of shorelines, the deposited minerals of vents, and the water-gas transition of liquid surfaces and bubbles of trapped gas. Molecules will be constantly interacting with other molecules in these phases, and making transitions between phases through dissolving, evaporating and condensing actions. To represent the solid and gaseous borders surrounding even the tiniest droplet in a pinhole in rock would require trillions of atoms. In addition there are energy gradients between sources and sinks. These gradients are key drivers in chemical reactions and in the motion of the medium and must somehow be simulated. Related to gradients are the vital influence of periodic effects that could arise from wave action, the day/night cycle or the action of geysers.

An additional simulation challenge is related to *simulation time scales*. As we saw above, a few seconds may be important to model events in the life of a cell. A bacterium such as *E. coli* may have taken all the steps to perform a replication of its DNA in just about twenty minutes. Ecosystems shift in terms of energy, the flux of populations in hours to years. And evolution itself takes place on a scale of a few days for some single celled organisms to millions of years for larger animals or plants. However, actions such as the folding of proteins, catalysis of reactants and other biologically important activities take place in nanoseconds. Nanoseconds of chemical time can be simulated in hours to days of computer time for small volumes containing no more than a few thousand atoms (Shaw, 2009). Thus at these limited scales of both contents and time, there is a doorway into the world of the bio-molecules.

As is apparent from this brief overview, the computational load of simulating even simple bio-molecular mixtures in small volumes and over tiny time sequences (nanoseconds) is still very large. The computational complexity of simulating a complete living cell is frankly stupefying and well beyond the means of modern computing grids (Philips and Milo, 2009). However, the challenge we are setting here is to produce informative simulations of important prebiotic molecular processes and structures. Life's origins are postulated to have occurred in a liquid medium containing a diverse range of small molecules (Deamer, 2011). Larger molecules emerged out of this milieu. So in fact there is a probable doorway into this world. One major cautionary note here is that the time scales for the emergence of large prebiotic molecules and reaction sequences is still intractably large for computer simulation. Nature possessed the “computer” of an entire planet, running every instance of every atom in parallel. Nature could afford many trial and error experiments in many environments. Therefore the challenge of simulating pathways from nonliving to living molecules must be broken down into a number of more tractable intermediate steps. These steps would involve a small number of small molecules over small time scales with a very limited representation of their environment (no solid of gas phases represented at all) and a simplistic energy input forming a trivial kind of source and sink, or at best, a functional gradient.

2.1.3 Arriving at a Doorway of Tractability

So we have arrived at the likely only tractable entry point to computational origins of life endeavours involving molecular simulation: the treatment of small volumes of bio-interesting small precursor molecules over small bio-relevant periods of time. What sort of virtual experiments can be carried out at this scale and which of these would be interesting to origin of life researchers? Deamer suggests that the formation of individual lipid molecules from the feedstock molecules which could have arrived from meteorite impacts (Deamer, 2011, pp. 7-22). The simulation of the formation of small micelles (Szostak et al., 2001) and their contribution to the growth of a lipid membrane is also interesting. The arising of catalysts and autocatalytic cycles (Kauffman, 1995) is also of interest and involves a tractable population of smaller molecules. Polymers and polymerases involved in the growth of replicable information molecules such as RNA may possibly be "shoehorned" into small simulation spaces (Deamer and Barchfeld, 1982). Examples of these experiments and their applicability for simulation will be treated in more detail in Chapter 4.

For the purposes of this thesis project of limited time and means we must take yet one more step down the ladder of tractability and recast our goals yet again, as:

> *A realistic goal for this work is to produce a proof-of-concept that simple molecules may be formed ab initio from simple tiny atomistic soups over small time frames. Those simple molecules could form the starting populations for later work.*

The use of an *ab initio* approach is a somewhat arbitrary choice on our behalf. The origin of life is a story of the origin of increasingly complex associations of atoms. While we could start our simulation with many small pre-formed molecules such as water, dissolved carbon dioxide or ammonia, amino acids or sugars, this would burden us with additional atoms to handle and the challenge of molecular affinities to compute. By starting with a simple soup of atoms we could observe the formation of primal virtual molecules without being concerned with the complex geometries of their shapes and interaction. Given such a simple starting point, we are forced to accept the few simple experiments that are possible. Two such experiments immediately present themselves, running tests where the "goal" is to measure the number of molecules formed, or to measure the size of molecules formed.

At this point the reader may be thinking: *what is the use of carrying out such simple experiments?* The answer is equally as simple: *to provide proof-of-concept testing for a simulation platform in preparation for more ambitious future efforts.* This proof-of-concept will make several contributions to knowledge for future efforts in simulating emergent biomolecular phenomena, notably:

1. Demonstrate that the simulation of an artificial chemistry modeled on molecular dynamics can be cast into the form of a distributed grid of small volume simulations running on a network of computers, all managed centrally.
2. Provide key tools which would provide optimizations, both saving computing resources and shortening the length of time needed to observe emergent phenomena of interest.
3. Create an open system which allows researchers to add extensions to other types of simulation, including supporting multiple scales with multiple physics.

2.2 The EvoGrid: Our Design Emerges

With the major design choices and desired contributions to knowledge derived from the previous sections we are now in a position to propose the final form of the EvoGrid prototype.

2.2.1 A Chemical Cameo Simulation

Returning to our design for the first prototype of what we are optimistically calling the EvoGrid. Let us adopt the term *cameo simulation* to represent simulations comprised of no more than a few hundred or thousand particles representing atoms and small molecules running over short time scales and in multiple instances. As was suggested in the design exercise above, existence of those instances is governed by a search tree function which permits variations of initial conditions and the branching of multiple, parallel simulations. Variation of parameters and branching are under control of an analytical step which looks for interesting structures or behaviors within each cameo simulation volume, which we call a *frame*. Frames deemed less interesting may be terminated so as to permit other branches to be explored to a greater extent. This approach is inspired by the class of genetic algorithms (GA) combined with hill climbing algorithms widely used in Artificial Intelligence (Russell and Norvig, 2003, pp. 111-114). It is a form of importance sampling (Kalos and Whitlock, 2008).

2.2.2 Search Functions and Hill Climbing within Cameo Simulations

With such a large number of small simulations running, it begs the question of which simulation is producing desired results and therefore should be propagated to permit the emergence of further variations. This question can most appropriately be addressed by some form of computational search. In recent years, search has become a highly developed branch of computer science. Search through large noisy data sets of signals from radio astronomy telescopes became the focus of one of the earliest massively distributed computing efforts in the 1990s: SETI@Home (Anderson, 2004). Recently search has become widely used in dealing with vast quantities of text, images and other media on the World Wide Web. Search through the noisy content environment of web sites and social network traffic has now become an active area of study (Champin et al., 2010).

Search within AChems is less developed and as we have seen in the review in Chapter 1 tends to focus on identifying an occurrence or shape of a particular molecular structure such as a protein. In the most general terms what can be searched for in cameo AChem simulations are relatively localized “patterns in space and time” as defined in the Alife field (Langton et al., 1992). In the case of our design for the EvoGrid, search functions “score” and select frames of simulated atoms providing a pool of seed frames for another “generation”. So, current searches affect the generation of future content to be searched in a feedback loop. Such a loop is analogous to a chemical reaction system which permits constructive growth or destructive degradation of sought after phenomena.

Search functions in the EvoGrid prototype have the ability to *branch* new simulations based on scoring criteria. We use the term *branch* to describe the creation of new, optional simulations which may or may not be executed, or *simulated*, depending on selection criteria. These search functions are also able to implement *back-tracking* through stored scores from previous simulations to permit re-starting at a previously promising branch if the current frames are scoring poorly. As we shall see in Chapter 3, this back-tracking is implemented through the use of a *degradation factor*. In between every simulation step, the atomistic content of frames is affected by random effects including: the drifting of parameters, the adjustment of heat, the replacement of atoms with other types of atoms, and the breaking of bonds. Thus the *objective function* which defines the fitness landscape is dynamically changing that landscape with each application of these effects. Therefore we expect that our landscape should be *rugged*, (Figure 23), dynamic and that local maxima may be challenging to locate and to leverage into higher global maxima.

Searching for optima within such a landscape can take advantage of a number of algorithmic methods belonging to hill climbing or simulated annealing methodologies. Hill climbing is well understood in computer science and comes in a variety of approaches including: “Simple Hill Climbing (first-best neighbor), Steepest-Ascent Hill Climbing (best neighbor), and a parent of approaches such as Parallel Hill Climbing and Random-Restart Hill Climbing” (Brownlee, 2011, p. 39). As we described earlier in the example of Darwin’s finches, hill climbing is a good metaphor for adaptation in complex natural systems (Kauffman, 1995, p. 154) yet hill climbing programs can be quite simple (compressed) while still achieving timely results.

Simulated annealing is a related method analogous to gradual cooling in which good local, or even global minima can be found (Kirkpatrick et al., 1983, Kauffman, 1995, pp. 249-252). However, simulated annealing techniques can be both computationally and time costly. One of the goals of the EvoGrid is computational cost and time savings. Therefore it is prudent to seek “good enough” but not necessarily “best overall” solutions which in AChems can quickly become *NP*-hard and un-computable within reasonable time frames.

Without going into an exhaustive review of hill climbing techniques, let us describe the method chosen for this work: a form of *stochastic hill climbing* (SHC). Stochastic hill climbing is a method of hill climbing for navigating fitness landscapes which uses a local iterative optimization involving the random selection of a neighbor for a candidate solution but only accepting it if the neighbor is equal to or improves upon the current or parent solution.

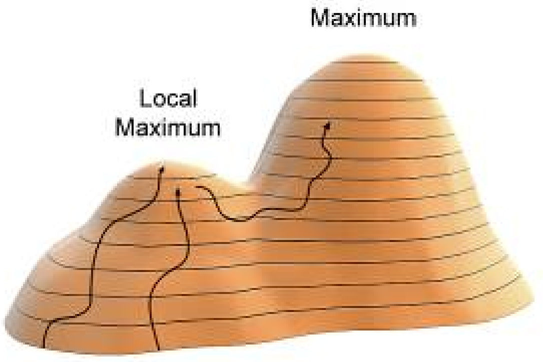


Figure 38 Illustration of stochastic hill climbing

Figure 38 depicts a cartoon representation of how a SHC might operate. There may be many pathways a traveler can take to a local maximum (the lower hill). The way up that hill looks somewhat like a random walk as the next steps are being examined at random but only taken if they are measured to provide equal or better upward placement (a gain in altitude) on the course. Once the maximum is reached the traveler will still be looking at random for a better next step. Not finding one the traveler may remain stuck on the local maximum unless this version of the SHC incorporates some kind of degradation factor or back-tracking. If the traveler becomes frustrated by not finding any steps that take her higher, she may opt to accept steps with lower “scores” and take those for a while. Finding her way into the more level “saddle” area between the local maximum and an even higher maximum, the traveler continues along as she is also programmed to accept steps that are equal in altitude to her current step. Soon the traveler finds her way to ascending values once again and travels up to the higher maximum.

What has been selected for the EvoGrid prototype could be termed “simple” stochastic hill climbing. Other more sophisticated implementations, including MIMIC by (De Bonet et al., 1997) discussed earlier, have explored more sophisticated hill-climbing techniques which involve correlating between paths. SHC was selected due to its relatively simple implementation and computational efficiency paired with a proven record of being able to scale multiple maxima on a rugged and changing fitness landscape. In the EvoGrid the fitness landscape will be changing with maxima growing and shrinking through time as the underlying objective function is changed. We felt that we would need an aggressive and fast technique to keep up with this dynamism. In addition, with the “state machine” being a volume of 1,000 rapidly interacting atoms the search space for the maxima is very large and not easily characterized. All of these conditioned argued for simple, speedy mechanisms.

We will now present a specific instance of search for a pattern in space within the EvoGrid’s implementation of cameo chemical simulations. To prove or disprove the hypothesis that search functions, which could be called a *directed search* can improve rates of emergence within cameo simulations, the EvoGrid prototype was designed to run in at least two modes, “control” (without search) and “test” (with search applied).

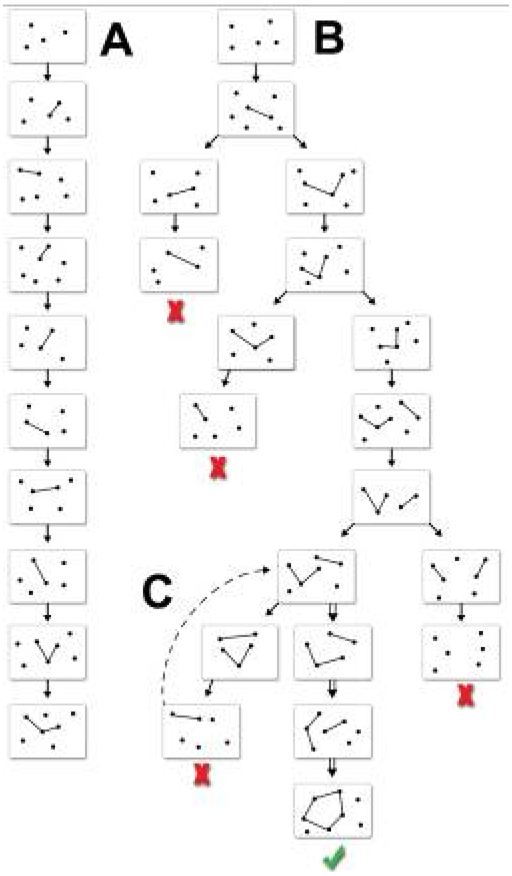


Figure 39 Illustration of the search tree method employed by the EvoGrid prototype

Figure 39 illustrates our implementation of an SHC within a simulation of frames of particles. There are two execution sequences, the first being a control (A) which depicts a typical linear time sequence simulation of frames without search and the SHC applied. The second is the test case (B) which depicts the arising of simulation branches due to search and selection for in this case the emerging phenomenon of more densely interconnected points. This illustration also depicts the other optimization we are terming a degradation factor, or temporal back-tracking in (C). If the simulation states of each frame can be stored through time, then a failed branch having low scores may be rolled back to the point at which "interesting" frames were still occurring. With this starting frame a new branch is started. This branch may therefore yield an interesting phenomenon forgone in the failed branch. In the example that phenomenon might be a ring structure, as shown in the frame with the check mark. In this way, *improbable occurrences may be guided across valleys of high improbability*. This technique for creating an "attractor" (Mitchell, 2009) for the formation of unlikely associations with higher likelihood is consistent with general ideas about the inexorable growth of complexity in physical and biological systems.

2.2.3 Examples from Biochemistry

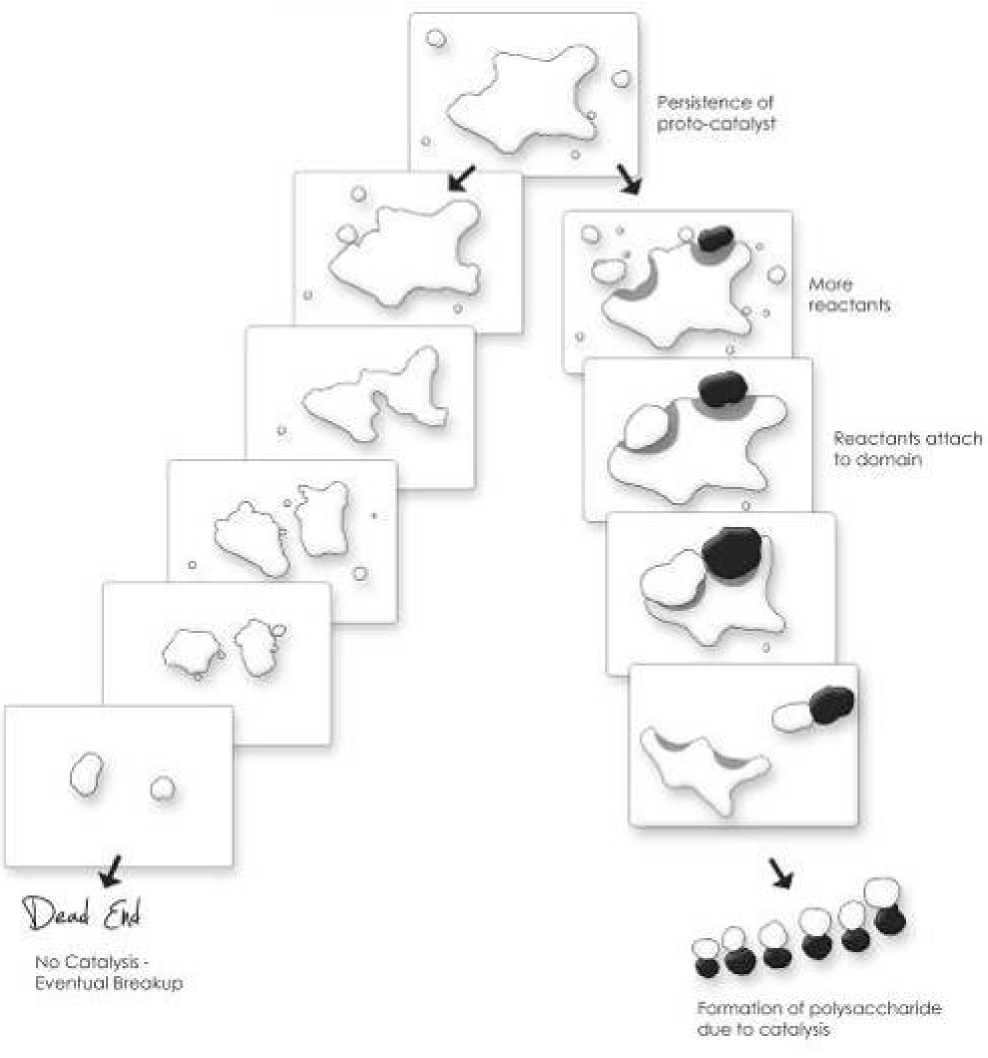

Persistence of proto-catalyst
More reactants
Reactants attach to domain
Dead End
No Catalysis - Eventual Breakup
Formation of polysaccharide due to catalysis

Figure 40 Example search tree whose nodes contain the formation of an artificial catalyst

Figure 40 depicts how this technique might be applied to a scenario in artificial chemistry, that of the emergence of a simulated catalyst. By knowing in advance about the geometry of active catalytic regions of molecules, the system might determine a given emergent molecular structure is developing the characteristics of a catalyst. A catalyst is a molecule that can act as a kind of “clamp” to perform (by bending) the formation of a bond between two simpler molecules that have attached themselves to the active sites on the catalyst. On the leftmost branch our simulated catalyst persisted but not long enough to actually catalyze anything before being broken up. On the rightmost branch, through our SHC directed search, the catalyst existed long enough to catalyze one or more reactions. If we did not know about the active sites in advance, a search function might still have detected the signature of the products of catalysis, in this case the formation of a polysaccharide out of the products of catalysis. The lack of *a priori* knowledge is a handicap but not necessarily a show-stopper if there is some unambiguous signal to search for. In the catalyst example knowing the template for the active sites that will make a molecule into a catalyst will assist the hill climbing. Not knowing this information would require the system to simulate many more pathways.

An extension to the detection of catalysts would be the detection of a whole network of reactions forming a so-called autocatalytic set (Kauffman, 1995) as shown in Figure 41 below.

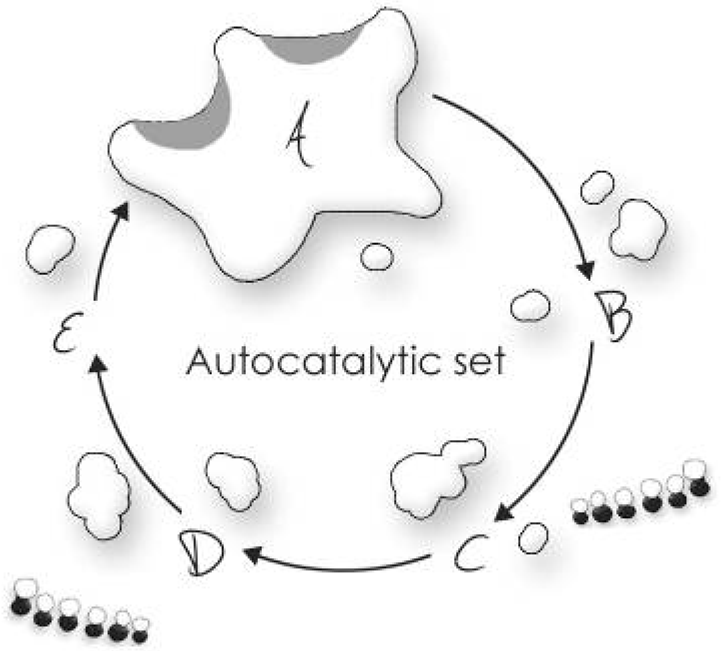


Figure 41 The autocatalytic set

In this kind of reaction, the catalyst is able to facilitate reactions that eventually lead to the creation of new copies of the catalyst itself, so the whole cycle can persist and cause the increase of population of the catalyst and a whole range of related molecules. It is thought that such autocatalytic sets or cycles were essential steps or *ratchets* in the origins of life. Observing the spontaneous emergence of such cycles in a molecular dynamics simulation would be a major result and be especially useful if they led to predictive models to develop these cycles in the laboratory. For this reason, this is one of the proposed future experiments presented in the road map in Chapter 4.

2.2.4 How Cameo Simulations Might Support OoL Simulations: The Genes of Emergence

Efforts to bridge nonliving and living matter and develop protocells from scratch (Rasmussen et al., 2003b) will rely on bottom-up self-assembly with commensurate self-organization of classes of molecules. The development of repeatable self assembly experiments *in silico* could serve as an important aid to *in vitro* protocell research. Self-assembly in computer simulations may be purposefully designed into the experiment or may be an emergent phenomenon discovered by a directed search through multiple trial simulations. The initial conditions for each simulation frame could be thought of as the coding sequences of a genetic algorithm (GA), and the simulation outputs seen therefore as its expressed phenotypes. The EvoGrid's search for self-assembly and other phenomena in cameo simulations is therefore a search for what we might term *genes of emergence* (GoE).

GoEs may be derived from within many different types of simulation, not just in the computationally intensive molecular dynamics (MD) world. More abstract simulation modalities may yield shorter pathways to the production of important emergent phenomena than through computationally complex artificial chemistries. The EvoGrid represents a "discovery system" operating on a continuum of techniques which might include: the execution of simulation modules that code for more tractable abstract universes yielding interesting results; to be then swapped out for a simple AChems within which we would hope to reproduce the results; and finally, carrying the GoEs one step further into high fidelity molecular dynamics which could make possible validation through full scale *in vitro* experimentation.

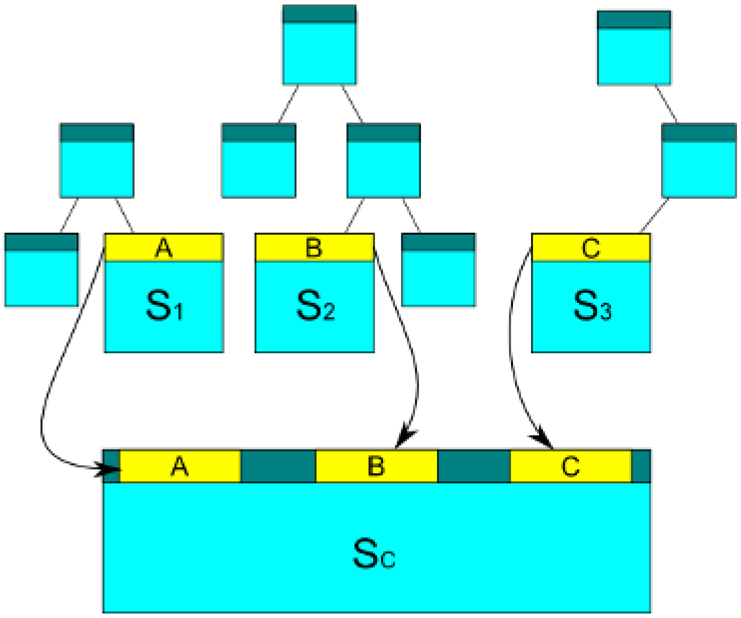


Figure 42 Illustration of the concept of cameo simulations feeding a larger composite simulation

Figure 42 graphically illustrates an extension of our original SHC tree. SHC search functions represented here as trees process through a number of small cameo AChem simulations. The end points of these simulations (shown here as S1, S2 and S3) each meet some criteria for generating a structure or behavior of relevance to a larger composite simulation, Sc. In the next stage, the large simulation Sc is assembled from a mixture of content from each of the "feeder" cameo simulations and is driven by an amalgamation of the individual cameo simulation experimental parameters A, B and C. The hope is that this amalgamation in simulation Sc, running with a much larger content store and over more biologically significant time scales, would generate a rich mixture of phenomena, such as the formation of membranes, emergence of replicators, or the observation of autocatalytic reactions. It is this enriched simulation environment which could be the basis for more ambitious computational origin of life endeavors. In another twist, an interesting phenomenon observed in Sc could be captured, its parameters and local contents extracted and cameo simulations run to characterize and fine tune the phenomenon more closely. Re-inserting the results of this “improved” sub-simulation may enable another ratchet in the emergent power of the larger simulation.

Not surprisingly this is a method commonly employed in bench chemistry when more efficient reactions are sought: an iteration of feeder solutions is developed separately and then tried repeatedly together. An example of this is the famous ribozyme experiment by (Bartel and Szostak, 1993) which is described in more detail in Chapter 4 of this thesis. We hope that this section has helped establish the value of search and hill climbing using small volume cameo simulations.

2.2.5 Global Optimization Problem Statement

Formally, the system is intended to solve a global optimization problem, which we will describe next. In summary and prior to describing the implementation of the EvoGrid prototype, let us summarize the overall properties of the simulation and optimization system we are seeking to build. The properties of the architecture and optimization system are as follows:

1. The *parameter space* of the system consists of a series of volumes of 1,000 simulated atoms of discrete types (2 in the first prototype and 6 in the second). These volumes are of a dimension of 10nm on the side, for a total volume of 1000 cubic nanometers containing 1,000 atoms whose density varies slightly throughout the volume. Each atom has an initial randomly selected velocity and a specific bond outer threshold, Lennard-Jones force parameter and covalent bonding force parameter (as specified in section 3.1).
2. *Constraints* applied to the parameter space include the limitation of heat (combined velocity of the particles) to the range of 300 degrees Kelvin. Adjustments to the level of heat (the velocity of all atoms) are continuously made throughout the execution of simulations.

3. *The objective (fitness) function* is computed from a simple addition of the bonds formed between atoms during the execution of the simulation of a particular volume. The larger the number of bonds formed the higher the "scoring" or fitness is of any given simulation. Beyond this there is a quality distinction which creates two "experiments": in one experiment fitness is measured by number of distinct molecules formed (bonds between two or more atoms) and in another the length of the longest bond within any molecule is the primary fitness measure (the molecular size). The utilization of these two variations on the objective function coupled with scoring and simulation prioritization is described concisely by the pseudocode listed in Table 4 in Chapter 3.
4. *The optimization method* is a single (non hybrid) global optimization method employing stochastic hill climbing which is enabled by the processing of scored, prioritized simulation branches. Other methods such as simulating annealing, as described earlier in this section were considered to be too computational intensive to employ while the application of a hybrid approach combining gradient methods, genetic algorithms or the searching of distinct molecular populations could be a direction for future research.

2.3 The Architecture of the EvoGrid Prototype

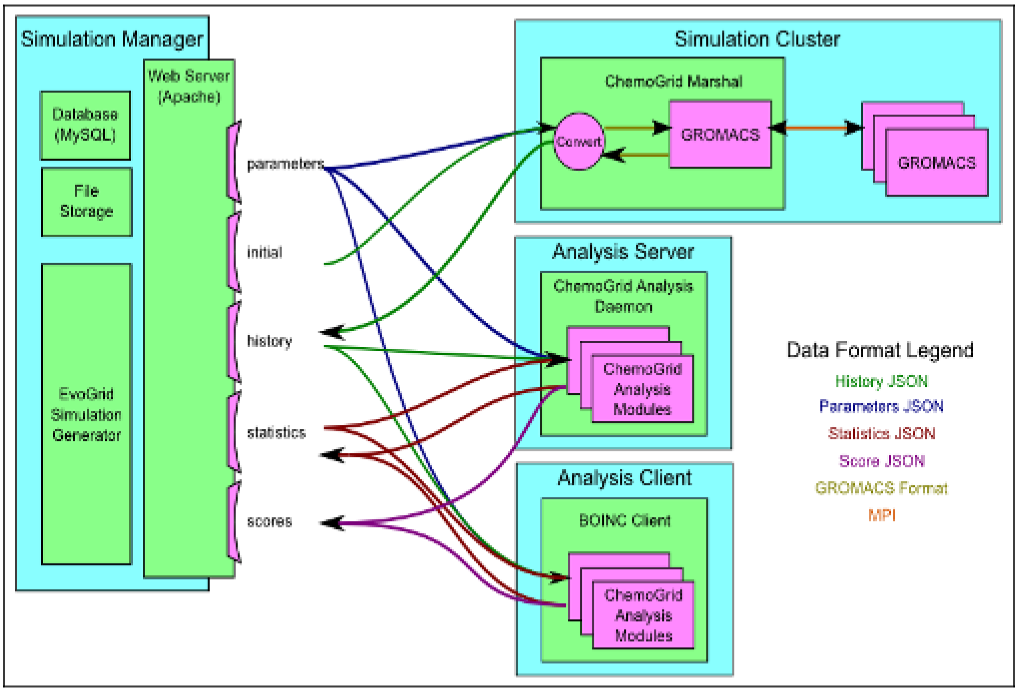


Figure 43 High level design and data flow of the EvoGrid shown in block diagram format

We have finally arrived at the point of having sufficient design principles and goals established for the first experimental prototype of the EvoGrid to be sketched out as software and hardware architecture. As depicted in Figure 43, the modular design of the EvoGrid encapsulates in its Simulation Cluster an MD simulation engine, in this case version 3.3 of GROMACS (van der Spoel et al., 2005), which we found to have good and stable performance on individual computers and was suitable to run as a plug-in component. It should be recalled that GROMACS was also used as the distributed MD engine on the Folding@home projects (Pande et al., 2003) described in Chapter 1 so comes with some pedigree. In their open questions in artificial life (Bedau et al., 2000) stated that combinations of different simulation approaches might be a pathway to significant progress. We therefore designed the EvoGrid framework such that GROMACS could always be swapped out for other suitable simulation systems or that the framework could support many simulation engines running in parallel on the same data sets. Other components depicted in the figure include an Analysis Server and an Analysis Client. Both of these components process inputs and outputs to the Simulation Cluster using the compact JSON format.

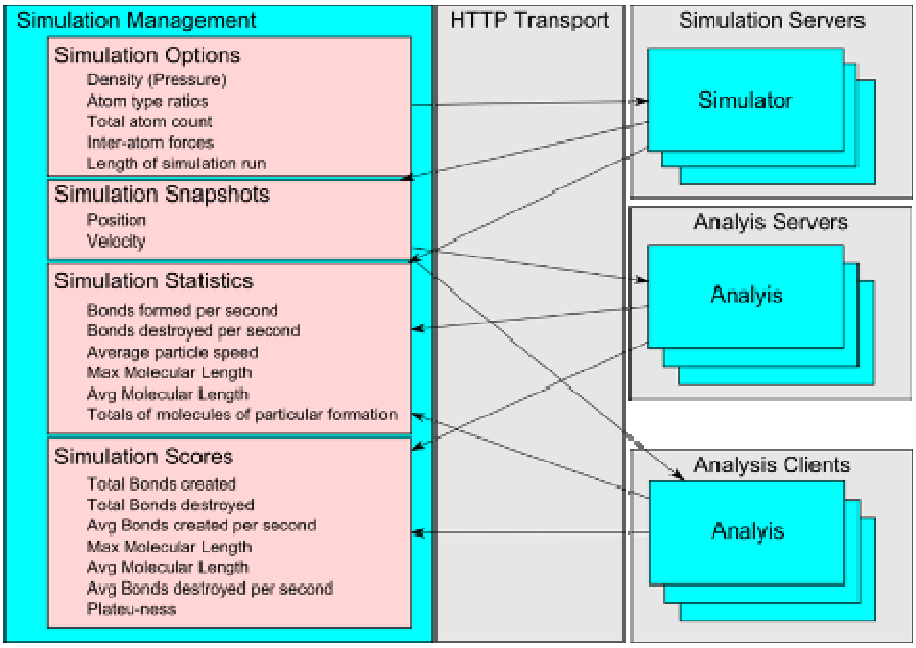


Figure 44 Lower level sequencing of data types through the EvoGrid with an emphasis on stepwise simulation

Moving to a view of the step-by-step flow of activities in Figure 44 we see that the Simulation Manager running via HTTP/Web services sequences the simulation analysis of individual frames. MD simulations typically have heavy compute loads in executing the time-steps for each force interaction of the artificial atoms. In the EvoGrid, hundreds of thousands of these time-steps are being executed and future computing frames are replicated through new branches. This process can generate terabytes of stored states for analysis. This could eventually call for a fully distributed simulation network, such as provided by the BOINC network (Anderson, 2004). BOINC supports many computationally intensive scientific applications, similar to Folding@home (Pande, 2011). However, at this time we are relying on the centralized analysis server.

2.3.1 Detailed Implementation

In the previous sections we detailed both a high level architecture and preliminary results from the EvoGrid prototype. In this section we will provide insight into the code-level design implementation of the prototype.

Simulation Manager

The simulation manager (SM) acts as the central data distribution point for the batch processing. The SM uses HTTP for communication, and provides either human readable XHTML or machine readable JSON. The SM shown in functional block format in Figure 44 accepts, stores and provides:

- Specification for pending simulation jobs
- Histories of completed simulations, for processing by analysis functions
- Statistics generation by analysis functions, for processing by searching functions
- Scores generated by both analysis functions and searching functions.

Due to the amount of data being stored and transmitted, the hardware requirements for the SM include large disk drive capacity for file storage, and database storage. The SM provides a method for daemons (software agents) to request “pending” variations on data to be processed. This allows the SM to select what order data should be processed in. To date, the selection method used computes the ordering by the “priority” property, followed by random selection from the pool of frames with the highest priority.

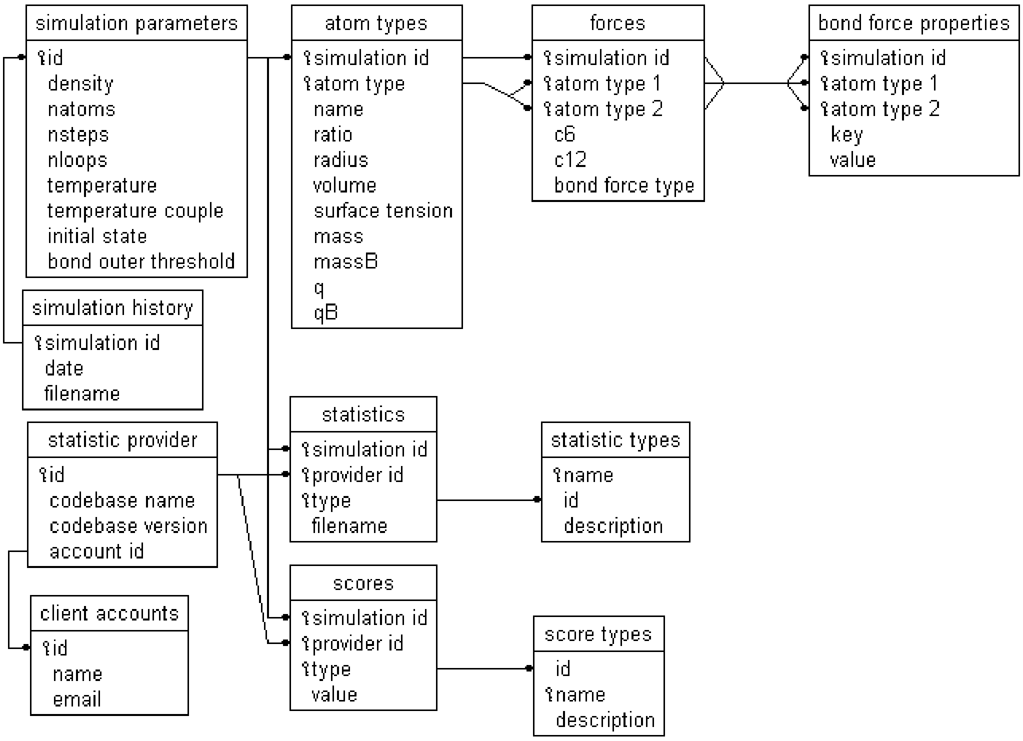


Figure 45 EvoGrid data type hierarchy

Statistics and scores are currently accepted in an open format in that any statistical score and naming convention can be used and this will be automatically added to the storage database. If there are no pending simulation specifications, then the SM generates new ones, by providing random parameters. The random parameters include the number of atom types present in the simulation. In the first prototype this seed generation was the only point capable of varying the number of atom types present.

Figure 45 illustrates the SM Database Schema and is useful to understand objects being managed and transacted. The simulator component retrieves pending simulation job specifications from the SM, performs these jobs and submits the history back to the SM. For a more exhaustive explanation of these data types see Appendix C.

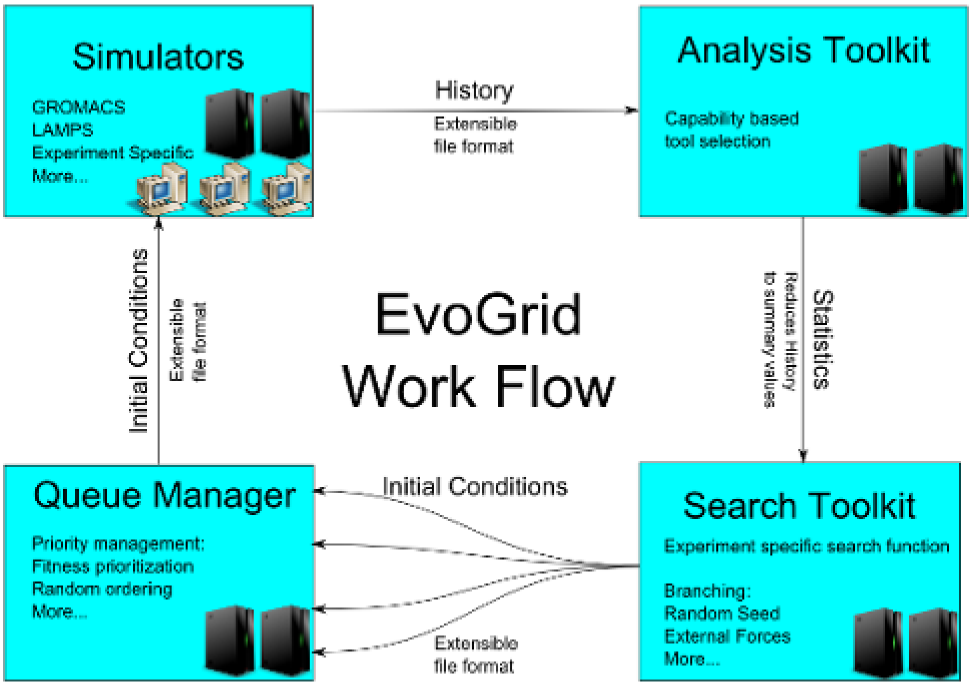


Figure 46 A view of the high level architecture of the EvoGrid

Figure 46 is a final look at the high level architectural components of the EvoGrid prototype. The EvoGrid is built upon an open source, published framework to allow a future community of researchers to extend it and the source code is published on the EvoGrid project web site (Damer et al., 2011a). *See Appendix C: Detailed Implementation and Source Code Examples* for detailed code and Application Programming Interface (API) level execution of the Simulation Manager and clients.

2.3.2 Target Chemical Model for the EvoGrid Prototype

One final step remains in our design exercise: setting a realistic target chemical model for the first EvoGrid prototype to simulate. Due to the severe constraints on this first prototype we will be well advised to limit ourselves to the most basic of chemical models: that of a highly diffuse atomistic gas, that is, a cubic volume composed entirely of unbonded atoms. David Deamer suggests that the EvoGrid prototype might represent a first step toward modeling the interstellar medium (Deamer and Damer, 2010b). Deamer describes this medium as “a mixture of elements in the gas phase, such as hydrogen, helium, nitrogen, oxygen, and sulfur; simple compounds like water and carbon dioxide; dust particles composed of silica… and metals like iron and nickel" (Deamer, 2011, p. 18). So it is suggestive that the prototype models the first items on this list, a mixture of elements in the gas phase. In terms of origins of life research it could be useful to implement a crude emulation of the behaviour of a diffuse atomistic gas similar to the interstellar milieu from which the first organic compounds formed. It is understood by astrophysicists and astrobiologists that this milieu is the source for the compounds for the formation of stars and planets (p. 19) including the feedstuffs for living systems.

2.4 Comparable Work to the EvoGrid

Other recent and active projects using techniques comparable to the EvoGrid are explored next. In the scientific and hobby communities there have been numerous simulation systems constructed using various levels of abstraction in artificial chemistry. Many of these systems use naïve chemistries and have pre-built objects designed to express some aspect of a living system. Some systems run on a single computer and others can be distributed. Many of these systems rely on the operator to detect "interesting phenomena" visually, while others implement some sort of automated search. Indeed, since Barricelli's Numerical Symbioorganisms in 1953 and Conway's Game of Life described in the Introduction, this class of software has become one of the most creative exploratory uses of computers. Winnowing down to a few projects in the serious research space Table 2 lists other projects using comparable techniques to study origins of life or emergent phenomena.

Project	**AChem model**	**Goal**	**Status**
Folding@home Pande et al. Stanford Univ.	MD-GROMACS Massively distributed	Folding of molecules	Active research tool
FLiNT nanocell/protocell Fellermann et al. Univ S. Denmark	Custom coarse grained DPD on single processor	Emergence of lipid nanocells to simulation of FLiNT protocell	Development to work with LAMMPS and possibly EvoGrid
GARD Lancet et al. Weizmann Inst.	Stochastic-MD Hybrid in MatLab on single processor	Monomers joining and leaving composomes	New version in development, possibly with EvoGrid
Avida-derived from Tierra Offria et al. MSU	Assembler Automata 2D Lattices distributed	Study of artificial evolution	Active development and use in research
Squirm3 Hutton	Abstract CA on single processor	Emergence of replicators from random soup	Active project as Organic Builder network app.

Table 2 Listing of projects comparable to the EvoGrid

Perhaps the closest work to this effort in terms of the tools employed is the Folding@home project extensively discussed in section 1.2.3. This project employs the GROMACS MD engine selected for the EvoGrid but not in an “emergent” mode of new molecules forming as in the EvoGrid. Instead, Folding@home is using a large number of distributed copies of GROMACS to compute how large molecules, specifically proteins, fold. Folding@home is a successful endeavour with a continuous stream of discoveries, network growth and usership and publications (Pande, 2011).

The FLiNT nanocell simulation developed by Harold Fellermann (Fellermann et al., 2007) is described in detail in section 4.2.1 in Chapter 4. This system is also close to our work on the EvoGrid and was in fact one of the inspirations of this effort. Fellermann developed a custom coarse grain dissipative particle dynamics (DPD) simulation in which the formation of small lipid droplets with attached surfactant could be automatically searched for and identified within frames of simulation running on a single processor. Active discussions with the FLiNT group are proceeding as to how to bring Fellermann's work into the EvoGrid framework. Where the nanocell simulation differs from the EvoGrid is that atom to atom reactions are not performed and the environment runs on a single computer for short periods of time under operator control.

The Graded Autocatalysis Replication Domain (GARD) statistical random chemistry framework (Shenhav and Lancet, 2004) described in Chapter 1 is also closely aligned with our current efforts with the EvoGrid. These authors have developed a hybrid scheme merging MD with stochastic chemistry and run simulations in the MatLab® package. GARD explores rates of monomers joining and leaving assemblages to study the evolutionary lifetimes of composomes. Exploration of methods to run GARD as a simulation within the EvoGrid framework has recently been undertaken.

Avida (Ofria and Wilke, 2004), currently being developed at the Digital Evolution Laboratory at Michigan State University, is an example of an "assembler automata" system. In such systems agents are defined by simple instruction sets acting on local and common memory. Any type of object can be defined as can rules of interaction such as emulations of inter-atomic forces and bonds. Avida has had some success modeling genes, gene expression and populations of simple agents that show some of the dynamism paralleling populations of *E. coli*.

Independent researcher Tim Hutton developed squirm3 as a CA-based environment to study minimal evolvable self-replicating virtual molecular structures roughly analogous to DNA or RNA. The project has recently been recast as the distributed open Organic Builder framework (Hutton, 2007).

Conclusion

In our search of the field we have found no exactly comparable project and implementation to our efforts, which are: *the distributed simulation of molecular dynamics volumes using search stochastic hill climbing to optimize the de novo emergence of bond formation*.

2.5 The EvoGrid: Answering Questions of Teleology and Intelligent Design

The creators of the EvoGrid cannot easily rebut any criticisms of “intelligent design” as with any software system it was indeed intelligently (or at least we hope competently) created by the hand of programmers. That said we strove to build a system in which, once the initial conditions were established, the complexity we hoped to observe would automatically emerge *without interfering hands*. This kind of emergence is a commonly sought after goal in work in many simpler artificial life systems (Sims, 1991, Ray, 1991) but has not often been attempted in chemical simulation.

The EvoGrid is a project to construct and test an exemplar emergent system and using common tools from the chemical simulation field of *molecular dynamics* and by extension other artificial chemistry systems. In this instance of our design of the EvoGrid prototype what will be driving complexity forward is *random chance*. Random chance will occasionally produce better results, and these better results are automatically “found” and “prioritized” using the search functions introduced above. For example, in a possible early prototype if our search scores are to be based entirely on, say, the formation of bonds we would simply select simulations where more bonds randomly form than are destroyed. What drives the formation of the simulation volumes themselves is therefore at least partly a random process. What permits those volumes to be selected for future processing would also be driven by random chance. To a great extent, this kind of experimental design parallels the day to day work in bench chemistry: experiments are carried out in small volumes in large numbers, sampled and analyzed and a few experiments selected to carry on, being replicated in large numbers for the next “batch”. This is sometimes termed “enrichment sampling” (Kalos and Whitlock, 2008) and is a common technique used in the discovery of new compounds.

A final point of design emerges from our quest to build a “black box” system to minimize operator bias. To do this our design should not code logic to *force* more bonds to form or directly influence the formation of more bonds. We would design the system to prioritize limited computer and simulation time to randomly generate conditions encouraging or presenting pathways to the behavior we were interested in. Each branch therefore retains equal likelihood of being selected, or "happening". In other words, the system should not create more bonds because we desire more bonds. Instead the design of the system would support the varying of the conditions that lead to fewere or more bonds forming. Along the way, random observations would determine whether the system is automatically following these conditions to produce higher scored frames. In no way should the system (or the user) know ahead of time which changes were going to result in higher scores or not, and thus the system should not be able to direct the occurrence of these higher scores. In other words, we believe that the EvoGrid prototype, even though it is a designed system, should permit the emergence of unpredictable phenomena which, rather than being completely random, should permit some learning about the nature of operating complex, chemically inspired *de novo* simulations. It is this class of simulation which will be of value to origins of life endeavours.

Summary

This chapter established the connection between the cognate fields, prior art and limitations of current computing systems and a viable architecture and approach to support the construction of a prototype EvoGrid. Specific mathematical approaches (the use of stochastic hill-climbing), molecular dynamics simulation engines (GROMACS), and distributed software architectures and open protocols (web/HTTP and JSON) were proposed. An approach utilizing search, scoring, back-tracking, and the random selection of frames was fit into a tractable computing model of “frames” of 1,000 virtual atoms and naïve bond formation. Lastly, concerns raised by the use of a teleological approach (a search function with end goals) were addressed. In actual fact, the real proof of the property of emergence within the EvoGrid will come with the running of actual experiments, which will be detailed in Chapter 3.

Chapter 3: The EvoGrid Prototypes: Implementation, Testing and Analysis

Introduction

In 2008 a small team at the author's company, DigitalSpace, working with the guidance of an energetic international group of advisors, undertook to build the first version of the EvoGrid which was called Prototype2009. Prototype2009 began preliminary trial operations at the end of 2009 and results were reported in a paper accepted for peer-reviewed publication in the proceedings of the Artificial Life XII conference (Damer et al., 2010). In early 2010 the next variant, EvoGrid Prototype2010 was produced and undertook several longer experimental runs in 2010 and again in 2011. We will present and analyze the results from both prototype variants in this section.

3.1 Implementation of EvoGrid Prototypes and GROMACS

The 2009 prototype of the EvoGrid was developed as a proof of concept of the architectural design detailed in Chapter 2. Many of the components described in Chapter 2 were implemented. What was not implemented was the storage of the state (atom and molecule position and velocity) of each simulated volume, or frame. State storage was to be implemented subsequently in Prototype2010. Without the ability to engage in re-starting simulations using prior states, the simulation frames began with an identical population of one thousand atoms of two kinds (Hydrogen, and Oxygen) which could all be simulated by GROMACS. The sole method to introduce variation into the simulation was the restarting of computing on a newly generated random frame with a drift of global parameters to GROMACS.

It is beyond the scope of this thesis to describe the internal operation of GROMACS in detail, as we are treating GROMACS as a "black box" in the development of our architecture and the testing of our optimization. However, for future reference it is still valuable to enumerate the parameters we set for GROMACS during each frame computation. Global parameters included:

- Density in particles per Angstrom^3 with a range of: 0.01 - 0.1.
- Temperature in Kelvin with a range of: 200 – 300 deg K, used for initial velocity and temperature bath.
- Bond outer threshold in Angstroms with a range of: 0.1 - 1.0 Angstrom distance, used for bond creation.

The minimum computational unit consisted of frames of 1,000 simulated atoms which were run for 1,000 time steps of one simulated nanosecond each within the GROMACS engine. Random starting contents of the two atom types were generated for each simulation frame. All their parameters (mass, charge, force interaction with other types, radius and volume) were selected from a uniformly distributed random range. For Prototype2009 we employed notional atoms which incorporated the properties of Oxygen and Hydrogen when in a diffuse environment.

Forces between atom types included pre-computed components of the Lennard-Jones force function which was described briefly in Figure 28 in Chapter 1:

- c6 0.0 - 0.1
- c12 0.0 - 0.00001

Covalently bonded atoms included the following parameter ranges:

- rA 0.0 - 2.0
- krA 0.0 - 2.0
- rB 0.0 - 2.0
- krB 0.0 - 2.0

An initial series of tests were carried out in late 2009 and early 2010 on a single running instance of GROMACS on one computer with the simulation manager on a second networked computer. When a bond was created, the Lennard-Jones forces would cease applying and no new forces were applied. This was done to minimize real world constraints prior to having access to a better technology supporting covalent bond computations. Our bond formation code, applied to post-processed histories after each simulation, was extremely naïve and led to the easy formation of frequent bonds. In both Prototype2009 and Prototype2010 atoms are bonded if they lie close to one another (that is within the van der Waals radius of each other) in each of the 1,000 state snapshots in an executed simulation. Atom placement results from GROMACS simulating the relevant interaction energies for each snapshot, done within its own internal 1,000 time slice execution which creates each stored state. In Prototype2009 if each frame was not reset back to an atomic soup between runs, the system would have rapidly generated a single linked super molecule of all 1,000 atoms akin to a crystal lattice. The main focus of both prototypes was to be able to test the architecture, not faithfully simulate any kind of accurate chemistry.

As mentioned previously the position and velocity data was dumped every 1000 cycles and a naïve bonding applied to all atoms or atom-molecule or molecule-molecule objects. After a thousand of these dumps, this collected history was processed by the analysis server. To determine if a bond should be formed, forces between each type of atom were extracted from the GROMACS export:

- c16 - Lennard-Jones force parameter
- c12 - Lennard-Jones force parameter
- rA - Bonded force parameter
- kRA - Bonded force parameter
- rB - Bonded force parameter
- kRB - Bonded force parameter

The proximity of the notional atoms sitting within their mutual bonding radii but also having a sufficient velocity to satisfy bonding force requirements determined if a bond was formed. This is a very naïve method of determining bonds akin to magnetic attraction between steel balls rolling about on a flat table, some repelling each other and some forming bonds given sufficient contact momentum. In this model there is no differentiation between types of bonds formed, whether they were covalent, ionic or more subtle affinities. This accuracy and subtlety could be achieved with the future application of quantum mechanical calculations, which was well beyond the scope of this work.

Measured values	**Final simulation scores**
Average molecular size	2.2303
Maximum average molecular size	4.47307
Average maximum molecular size	9.355
Maximum individual molecular size	17
Final maximum search score	33.0584

Table 3 Scoring produced by prototype analysis server for the final simulation frame

Once a bond-forming process was executed on a given frame, the score for that frame could be computed. Table 3 represents the scoring for frame number 144,204, the highest scored frame in our Prototype2009 “test” case trial run (see (B) from Figure 39). This was from 236 simulations processed between December 14, 2009 and February 1, 2010 with 179,636 abandoned. The analysis and search function were set up to select for the formation of “larger” virtual molecules, which in our simplistic implementation meant a simple count of the greatest number of bonds joining any two atoms. Employing stochastic hill climbing methodologies, the maximum search score reached in the trial was the number 33.0584, a simple sum of the entries in Table 3. Note that all average values are calculated over the 1,000 time steps in each simulation. The maximum individual molecule size achieved in this frame was 17, which suggests a long string of interconnected atoms. No attempt to visualize this structure was made because the molecule was considered so unrealistic as to have no correspondence with real world chemistry. Our bond formation was so naïve that many such large notional molecules were formed in many frames. We planned to improve this process for Prototype2010 to come closer to a more realistic chemistry.

If the analysis showed the generation of a sought-after property, in this case an improving frame score, then new simulation branches were created. Each time this happened, three new branches were made inheriting the same global GROMACS parameters (density, heat, bond outer threshold) except that for each branch one parameter was changed, drifting either plus or minus one percent of value. Next, for each branch the per-atom global parameters were not changed; instead one atomic mutation (change in type of atom) was made within the simulation volume. In addition, for each branch the ratio of the two atom types was also mutated with each atom's ratio value randomly changed by plus or minus ten percent then the ratio was normalized. Note that none of these branches were actually simulated by GROMACS until randomly selected.

3.1.1 A Key Operational Concept of the EvoGrid: Loose Coupling through Random Selection Yielding Emergent Phenomena

This creation of a large number of possible branches but *only randomly selecting one of them to simulate* is a core concept of the EvoGrid design and its use of stochastic hill climbing. This random selection permits the environment to support “innovation”, i.e.: the system has leeway to “find its own path” by virtue of being loosely coupled to the teleological end goal suggested by the search function definition. If the search function having selected a promising frame for the creation of future branches, also had a role in selecting which of the next branches to simulate, then the system would simply be driven deterministically toward the teleological end goal. If the search function or logic of any kind had “driving authority” over any frame, it is likely that the computational load would be vastly increased, as many more branches would have to be preemptively explored to determine their suitability. In a loosely coupled system we lose the efficiency of always finding the best branch to take in any situation but we also drum out a great deal of the potential rigid determinism (and computational cost) of the system. In a simulation supporting the *de novo* emergence of phenomena we always hope to be “surprised” by results. Another way to think of this is: we must *apply some portion of influence to set the stage for the observation of some amount of emergence*. One consequence of this design is that a large number of branches are created and stored on a “stack” or in a “queue” for later processing (simulating) but only a few of these branches are actually turned into new simulation frames and executed.

An example from the author’s previous experience working in computer networking could help make this concept clearer. In the 1980s IBM’s Token Ring network technology was competing with Xerox’ Ethernet. Token Ring employed a deterministic approach wherein a token was passed around the network giving a computer permission to write to the network channel. Computers would have to wait for possession of the token to be able to write and were also locked in to polling the network to look for traffic destined for them. Ethernet on the other hand relied on the simple rule that computers first attempt to write to the network channel when they need access and, if they find it busy, simply wait a short but random amount of time and then try again. Token Ring was a tightly coupled architecture and Ethernet a loosely coupled one. In all tests at the time, Ethernet came out ahead in terms of performance (efficient utilization of the network, and scalability) while Token Ring was faulted for inflexibility, network delays and extra overhead supporting the control of the network resource itself. Thus, the core simulation scheduling function of the EvoGrid implements an Ethernet-style loose coupling with a random selection of branches to simulate, and the ability to retry and back track and simulate unprocessed branches if need be. As we shall see from the results of Prototype2010, this “random selection and retry” gives us the benefits of the stochastic hill climbing that not only allow the simulation to climb to ever higher local maxima but we also opt out of having to convert most of the branches into computationally costly simulations. The above model is a crude approach to navigating the dynamic and chaotic nature of real chemical systems. We now turn our attention to a more detailed analysis of the results from Prototype2009.

3.2 Results from the EvoGrid Prototype2009

3.2.1 The "Control" Case of Prototype2009

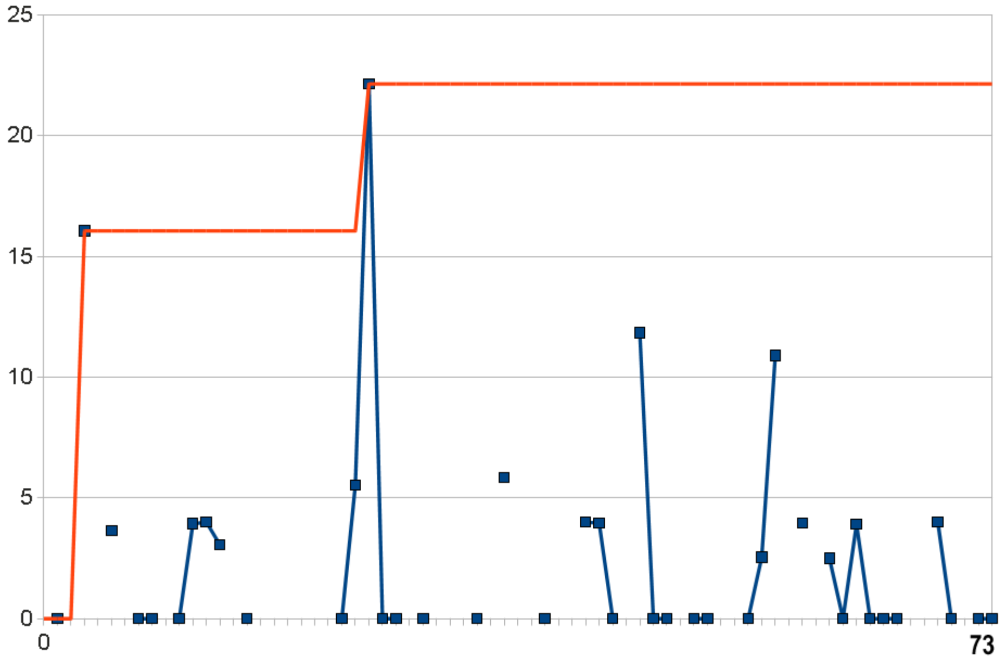


Figure 47 Scoring of experiments in "control" mode with no search tree function and random seeding

Figure 47 shows the "control" case (reference (A) from Figure 39) in which 73 frames were simulated with a randomly seeded restarting of GROMACS running for one thousand internal simulation steps to compute all atom-atom interactions and producing one thousand state dumps for analysis. As we can see, while there were some highly scored frames (red line), there is no maintained trend. Note that the missing lines indicate 31 simulation cases where our software generated impossible simulation configurations (atoms placed in the same position for example) and the execution was halted. This illustrated an area for improvement of how we were operating the GROMACS engine.

3.2.2 The "Test" Case of Prototype2009

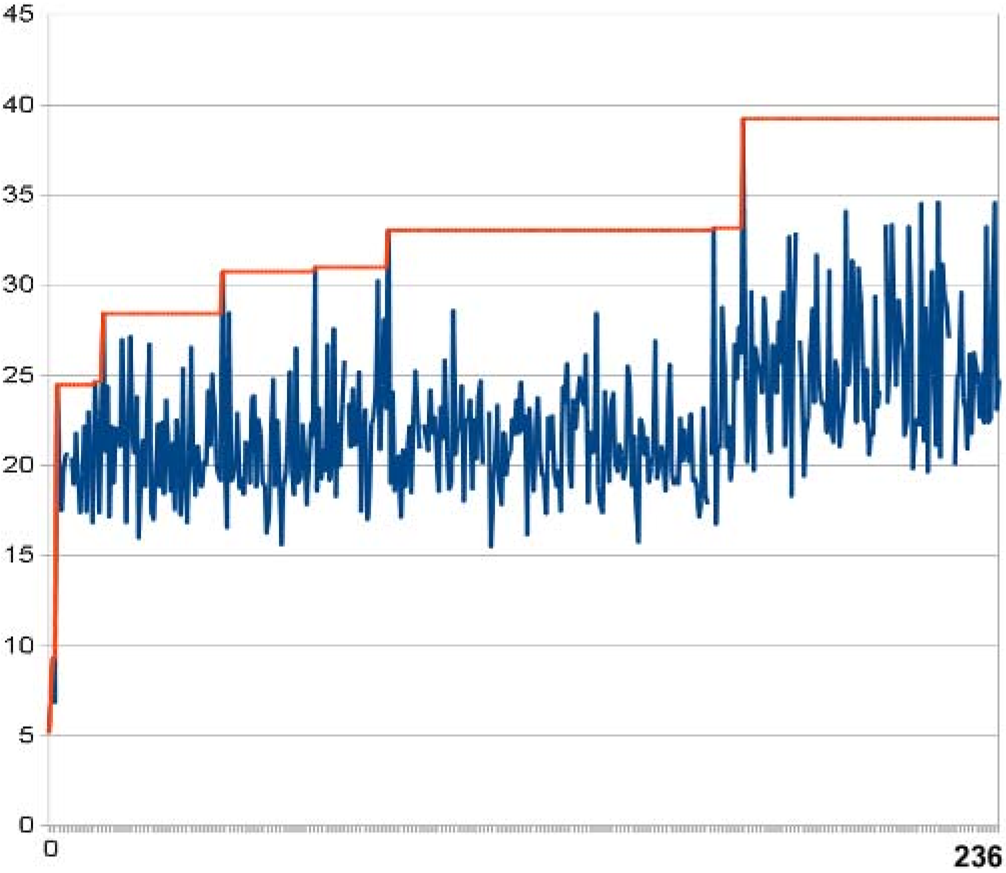


Figure 48 "Test" run showing trend toward higher "fitness" utilizing the search tree function

In Figure 48, the "test" case applied the search function over 236 completed simulations (with 10 halted and 179,636 abandoned, i.e. never simulated), and shows that the initially high value produced by the same starting frame generated for the control case is now improved upon over time. The strength of the search function and hill climbing is that subsequently generated frames eventually climb to a higher score-generating capacity, or fitness, over randomly generated control case frames. The search function will restart with lower performing simulations if all the potentially better options are exhausted. A record of these earlier simulations is stored as a linear stack of exported data. This is a crude implementation of back-tracking specified in our original design as shown at (C) in Figure 39. As seen in Figure 48, this back tracking allows the system to tolerate and bridge over a period where the evaluated simulation fitness (blue line) remains less than the best observed fitness (orange line). In this manner, the search function is operating as a stochastic hill climbing algorithm in that the system has the ability to find its way out of traps set by local maxima.

For further detail Figure 49 illustrates the underlying score of all components making up the summed scores in Figure 48.

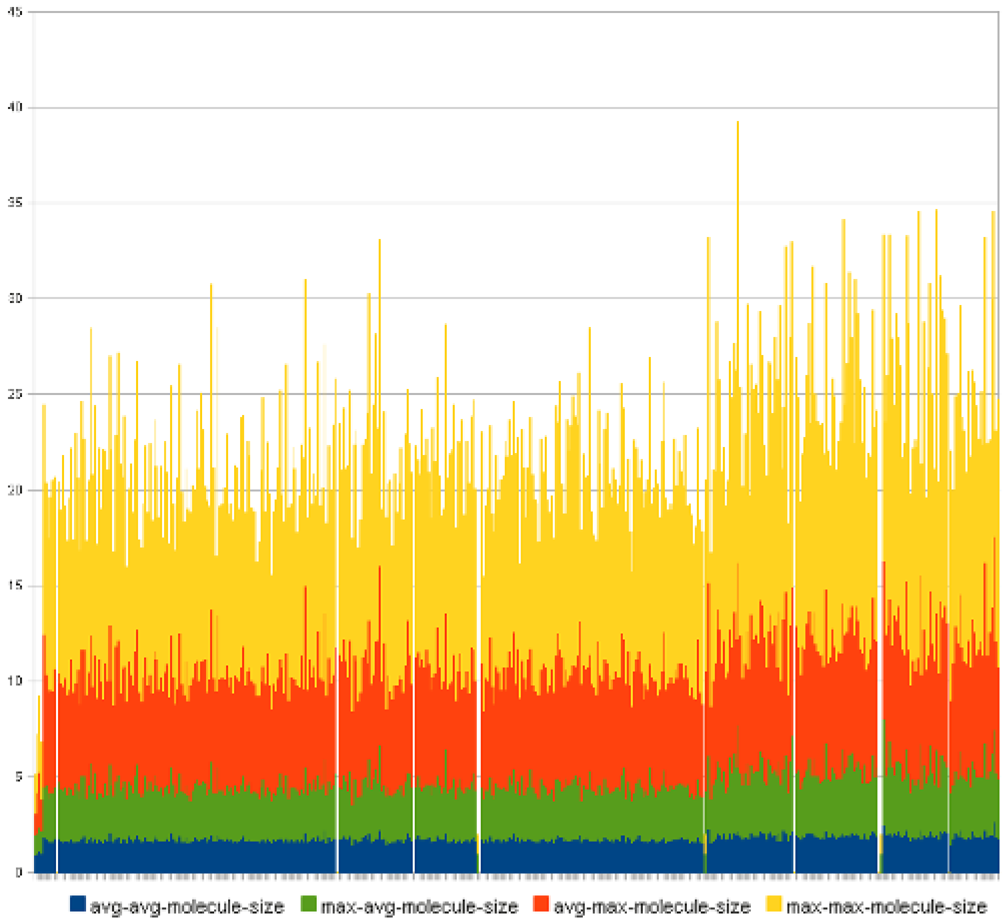


Figure 49 Components of the molecular size experiment simulation results

The seed sources of pseudorandom numbers generators in the system should be clarified. The random number seed used during the GROMACS simulation was generated using the mk_seed function provided by the GROMACS API. The simulation manager's random selection of which generated branch to simulate employed the MySQL RANDOM function, without specifying a manual seed.

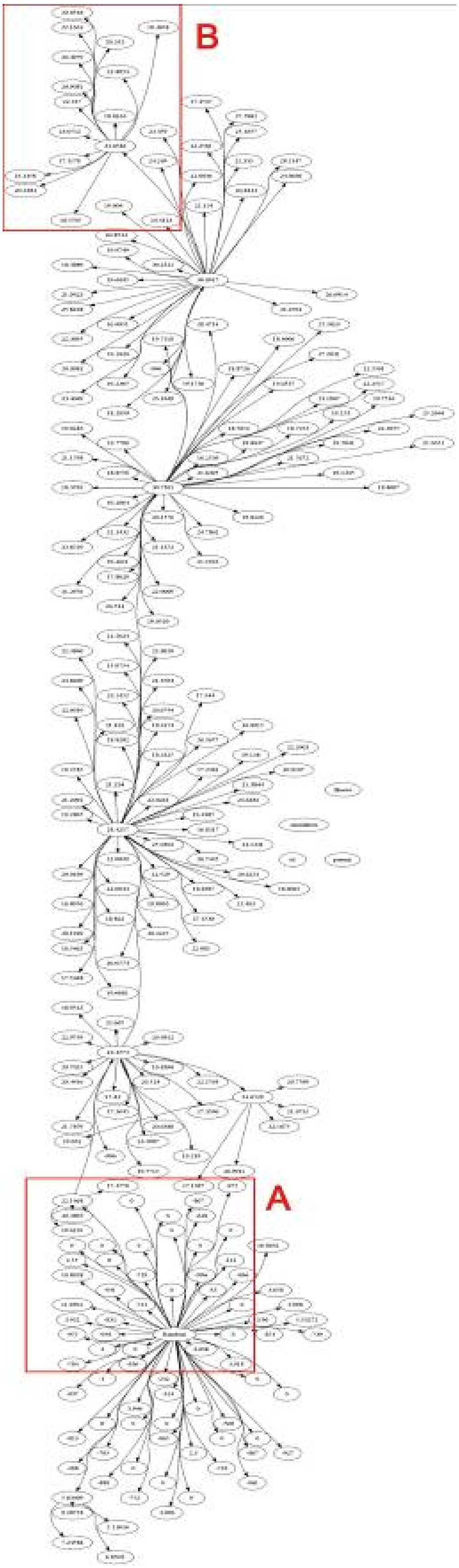


Figure 50 Inheritance hierarchy tree for Prototype2009 test case

Another view of the data from Prototype2009 is shown in Figure 50. This "inheritance hierarchy tree" (sometimes called a generation hierarchy) representation shows lines moving out from central "seeder" nodes representing simulation frames. The destination points of these lines are child frames that the system determined should be computed. The scores determined for frames are shown in each node. Nodes with zero or negative scores were the result of the failure of the simulation of the frame due to crashes or other anomalies that caused GROMACS to return null or unpredictable (negative) values. We simply discarded these frames.

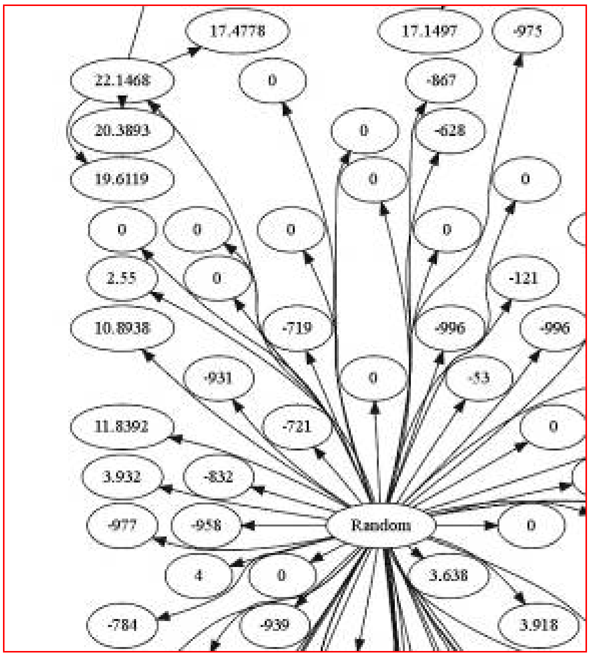


Figure 51 Starting seed simulation Random frame and immediately generated branches

Area (A) in the inheritance hierarchy of Figure 50 is extracted and presented in Figure 51. This is the area around the initial node, here labeled Random. Valid child nodes show that the scores are climbing toward a high of 22.1468 in the upper left corner. From that point, an arrow departs upward to a new cluster of nodes.

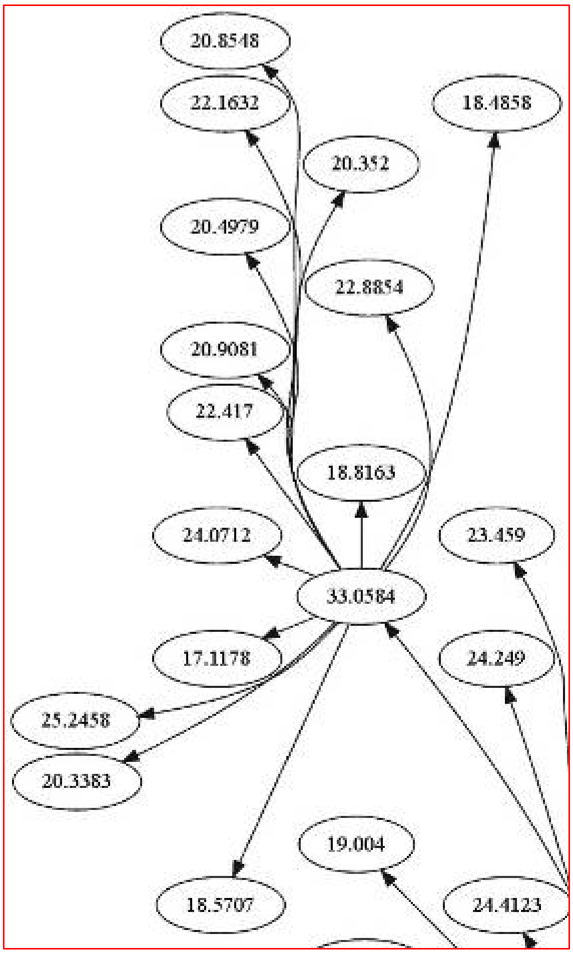


Figure 52 Final high score node in inheritance tree

Jumping to the topmost part of the tree, we can see in Figure 52 from the extracted region (B) shows that after wending its way through a number of hills and valleys of scores, the system has generated the frame with the top score of 33.0584 (center) and no higher scored successor frame is generated. We arbitrarily stopped the experiment just after this event to collate the data. It is important to note that many, many frames which were a part of declining score branches are not shown. Indeed, with almost one hundred and fifty thousand branches generated by the system, only a small fraction were selected by the search function for processing as simulations. This represents a gain in computing efficiency and reduction in time as a more deterministic system may well have sought to process more of these branches. Building an alternative system to carry out this more deterministic operation might be a good test, or additional control on these experiments, but it was beyond the scope of this work.

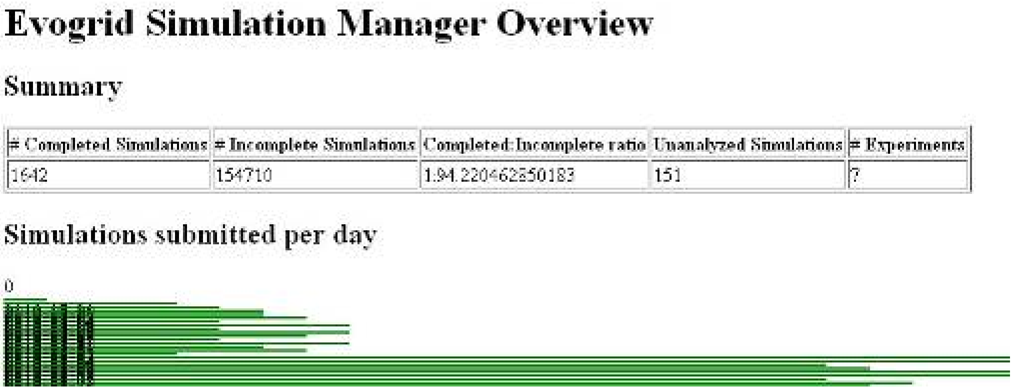
3.3 Design Implementation of the EvoGrid Prototype2010

The second EvoGrid prototype version, called Prototype2010, was put to the test in the summer of 2010. The major change from Prototype2009 was that the full state of each frame (atoms, molecules and their properties) were now being stored and able to be loaded back into GROMACS for continued execution. For the first experiments with Prototype2010 our simple simulations consisted of a random soup of six types (Hydrogen, Carbon, Oxygen, Nitrogen, Sulfur and Phosphorus) of one thousand notional atoms. These atoms were run through a hypothetical one thousand nanoseconds of simulation units using distributed copies of the GROMACS molecular dynamics engine running on a half dozen Intel dual core servers.

The six atom types were chosen for their key role in biology but this is, however, by no means a complete accounting of all bio-important molecules. From a definition of the term abiogenesis (Wikipedia, 2011) we note the proposed presence of our six chosen elements in the atmosphere of the early Earth:

> *Some theorists suggest that the atmosphere of the early Earth may have been chemically reducing in nature composed primarily of methane (CH_4), ammonia (NH_3), water (H_2O), hydrogen sulfide (H2S), carbon dioxide (CO2) or carbon monoxide (CO), and phosphate ($PO^4$3-)...*

Overseeing this process was a newly upgraded EvoGrid simulation manager with web-based administrative interface built into the Nagios environment. In addition, a publicly available simulation experiment web interface was produced enabling constant monitoring of EvoGrid experiments. See the simulation manager overview in Figure 53 and Figure 54, and a chart from an individual experiment in Figure 55 below.



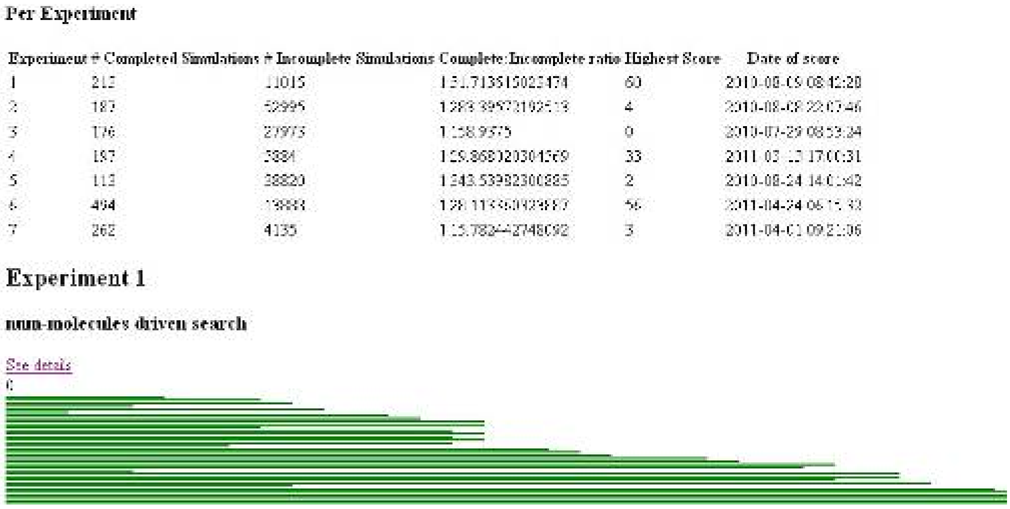
Evogrid Simulation Manager Overview

Summary

# Completed Simulations	# Incomplete Simulations	Completed:Incomplete ratio	Unanalyzed Simulations	# Experiments
1642	154710	1:94.220462850183	151	7

Simulations submitted per day

Figure 53 Simulation Manager web monitoring interface (top level)



Per Experiment

Experiment	# Completed Simulations	# Incomplete Simulations	Complete:Incomplete ratio	Highest Score	Date of score
1	213	11015	1:51.713615023474	60	2010-08-09 08:42:20
2	187	52995	1:283.39572192513	4	2010-08-08 22:07:46
3	176	27973	1:158.9375	0	2010-07-29 08:53:24
4	197	5884	1:29.868020304569	33	2011-03-13 17:00:31
5	113	38820	1:343.53982300885	2	2010-08-24 14:01:42
6	494	13888	1:28.113360323887	56	2011-04-24 06:15:33
7	262	4135	1:15.782442748092	3	2011-04-01 09:21:06

Experiment 1

num-molecules driven search

Figure 54 Simulation Manager monitoring interface showing processed and abandoned simulations per experiment along with aggregate scores

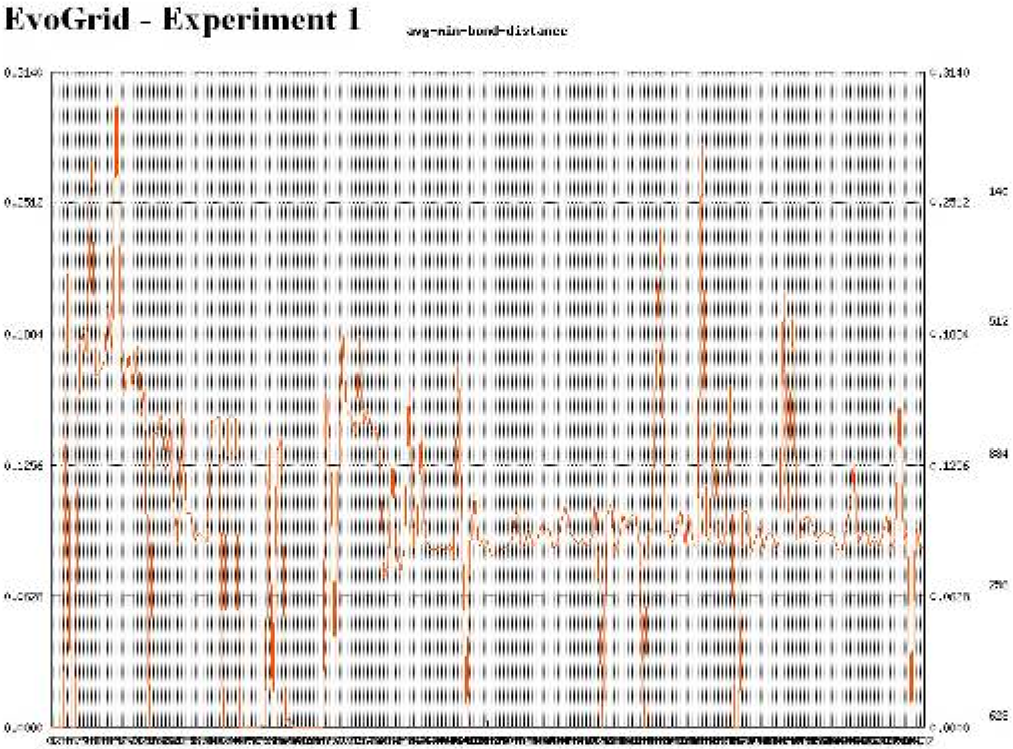


Figure 55 Simulation Manager monitoring interface showing individual performance graphs per experiment (in this case average minimum bond distance in Angstroms)

Other improvements in Prototype2010 included the implementation of two search functions, programmed to score and select the frames which had a superior outcome based on one of two predetermined criteria. These search functions therefore created two “experiments”. The first experiment sought to observe the increase of the richness of the simulation volumes by selecting frames having larger numbers of molecules, given that a “molecule” in this case is two or more bonded atoms. The second experiment searched and selected for frames which formed more large molecules, defined as molecules with the longer interconnections between member atoms.

This is a teleological approach in which desired goals are encoded by search, scoring, branch generation, and random selection mechanisms which are set up by human operators. The experiments are operated without operator interference with the hope of being able to eventually observe the emergence of a sought-after phenomenon. Searching and selecting is done by parsing periodic file “dumps” of the simulation state to minimize influence and therefore bias on the actual operation of the molecular dynamics simulation. It is hoped that the end goals will be met by the system, although this is by no means guaranteed. It is held that nature (excepting the human faculty of intelligence) does not operate in such a goal-setting fashion (Dawkins, 1986). Experimental bench chemistry as does much of science and engineering operates with goal setting and iterative testing and it is likely that any effective *de novo* chemical simulation system would have to be structured in this same fashion.

3.3.1 Description of the Frame Volume Simulation Experiments

The following steps summarize the implementation of Prototype2010. The starting conditions and actions surrounding the simulation “frame”, a three dimensional volume of six kinds of randomly placed atoms, are as follows:

1. The size of the simulation space is calculated based on the number of atoms to be simulated and an arbitrarily chosen density. To be able to simulate 1000 atoms with a density of 0.001, or 1 atom every cubic nanometer, we simulate a cubic volume of 10nm on the side.
2. A “heat bath” is applied, again through GROMACS, using an arbitrarily chosen initial temperature of 300K, roughly room temperature. We also set random initial velocities for the constituent atoms. The formulas for the calculation of these velocities are listed in Appendix C.2. Note that as with Prototype2009, the use of a thermostat for periodic “damping” of heat due to the over-excitement of the simulation resulting from our naïve bond formation was required. We maintain this “heat bath” at each iteration of a simulation and therefore bound the specified average atom velocity by scaling the velocities of all particles. In a simplistic way this introduction of heat and subsequent damping creates a crude source and sink of energy

(with no spatially defined gradient). The implementation of more sophisticated gradients is a goal for the future listed in Chapter 4.

3. We then generate the GROMACS-specific data file using the default values generated by the GROMACS API. The GROMACS engine then runs this instance of the space for 1,000 nanosecond “clock ticks” of chemical time.
4. We then apply searching and scoring with the evogrid-search-2010 functions which are described with source code listing in Appendices C.6 and C.7. The functions are: *priority_simple* which is used for Experiments #1 and #2; and *prriority_generation* which is used for Experiments #4-7.

3.3.2 Software in Operation – Core Pseudocode

To reiterate, the following pseudocode (Table 4) represents a high level description of the Prototype2010 operation of all test experiments.

```
Top:
  Calculate Fitness Score of completed simulation

  For exp 1,2, priority_simple function:
    Set Priority = Fitness Score

  For exp 4,5,6,7 priority_generation function:
    selection_daemon On
    If experiment 4,5
       Set selection_daemon Off

    Retrieve Parent Priority
    Retrieve Parent Fitness Score

    Set Priority = Parent Priority *
                   (0.9 * exp( Fitness Score -
                          Parent Fitness Score ) )

  For all simulations
     Simulate
     Branch
     If selection_daemon On
        Randomly select highest priority branch
        Simulate

Goto Top
```

Table 4 Core priority setting simulation code for the implementation for all test experiments.

The pseudocode for the “control” Experiment #3 follows below:

```
Top:
   Simulate
   Branch
   Simulate purely random chosen branch, giving no
   priority to earlier/later/complex/simple
   simulations.
Goto Top
```

3.3.3 Operation of Core Priority Setting Simulation Code

First, the simulation manager selects an un-simulated frame from the current experiment. The un-simulated state is determined by not having any associated history (stored frame state). This selected frame is randomly chosen from the branch members that have equal highest priority.

By following this process, any branch that increases in fitness will have increasing priority. When fitness ceases to increase, the priority will decrease over generations, as each branch is simulated. When the priority is lower than at a point "earlier" in the tree, that point will have the highest priority and is what will be simulated.

The score, or fitness, of a frame cannot be known before it is simulated. But the fitness of the preceding (parent) simulation is known. Eventually by simulating branches, a point will be reached that has a lower fitness (and priority) than a previously simulated simulation.

Each new branch is created when a simulation finishes and is placed in a queue for future processing. The simulation manager is continually picking from this queue and distributing simulation tasks to available client computers. At any point numerous branches of queued simulations may be followed. The simulation manager is keeping track of scores that result after each processed simulation completes. In practice a large inventory of unprocessed simulations accumulates.

At the point of creation of new branches we perform additional actions which emulate “natural” properties of chemical systems. These “branching actions” are described in more detail below. We are aware that we are falling far short of a complete model of real natural chemical settings, which would include energy gradients and a blend of solid, liquid and gaseous phases of matter among other properties. However, we believe that these branching actions, however naïve, implement a *crude emulation of the behaviour of a diffuse atomistic gas similar to the interstellar milieu from which the first organic compounds formed*. It is this environment to which the current EvoGrid prototype simulation is the closest analog.

Thus the selected frames are used as starting points for further “branches” in a simplistic inheritance hierarchy. A prioritization daemon (an automatous software agent) then selects at random from one of the available simulations from the generation that has the highest priority (score). Thus the initiating of new simulations for GROMACS is a stochastic process. It is possible that the stack of available simulations might contain generations which have identical priorities but this situation will simply result in a larger pool of valid potential starting points for new branches. The data for simulation frames that remain unprocessed when a branch is not pursued are added to a storage file stack. The system may return to that branch if it turns out to have been more promising (implementing the back-tracking example of Figure 39) so these frames are stored to the end of the simulation run.

The conclusion of our simulation experiments is determined not by any specific event but by the observation that frames were achieving consistently higher scores. At this point the stored frame database is frozen and indexed for statistical treatment by several tools. The proportion of processed to abandoned frames could be considered a measure of the overall efficiency of the EvoGrid prototype optimizations and the specific search functions.

3.3.4 Variations for Experiments #4-7

For Experiment #4 and #5 the *selection_daemon* was turned off. This forced the exploration of every generated simulation resulting from a given branching action. So, instead of a random selection of one of the branched simulations of high priority, all simulations were executed. It was expected that this might have the effect of increasing the probability of finding higher maxima but at a higher price of additional simulation. This strategy harkens back to Kauffmann's warnings about *NP*-hardness and the high cost of exploring every pathway in a fitness landscape (Kauffman, 1995, p. 155).

Experiments #6 and #7 were run with the same parameters as #4 and #5 but with the *selection_daemon* turned on so that only randomly selected branched simulations would be executed. It was hoped that the random picking would also find higher maxima but at a lower computing cost. Thus, experiments #4 and #5 act as a control for #6 and #7 in terms of a test of the full SHC optimization.

3.3.5 Explanation of Branching

A key activity in the EvoGrid simulations is the branching action. Once the GROMACS simulation has iterated through 1,000 time steps the state is saved to disk. Following the calculation of the score from that saved state and based on the experimental criteria, new branches are then created. These branches are prospective simulations in that they may or may not be computed. Branched simulations are filed into a stack data structure for later selection for execution. As we have seen, the population of branched but un-computed simulations grows much faster than the processed simulations.

The final, saved frame state of the simulation block of 1,000 frames is the starting point for branching. We are working with a naïve model of atomistic simulation somewhat reminiscent of interstellar chemistry (Deamer and Damer, 2010a): a diffuse atomistic reaction network. Within this naïve model we came up with an equally naïve if well intentioned model of the stochastic and physicodynamic activities that might occur in a diffuse atomistic reaction network. This model is implemented during the branching process. The goals of this model were to "emulate" the natural processes of changes in temperature, the breaking of molecular bonds, the changing of charge states of some of the atoms due to the effects of radiation, and the mixing of atoms flowing into and out of the area. The branching types and their action, shown in Table 5, implement these effects.

Branch Type	**Branch Actions within Frame State**	**Branches Created**
No-Op	Simple duplication of final state	1
Temperature	+1% or -1% adjustment to each new branch	2
Bond Breakages	New branch created with each bond broken which in turn depends on the number of molecules of size 2 or greater in the frame	Based on the number of bonds broken
Charge Changes	2 random atoms selected for +10% and -10% charge modification	4
Wash Half Random	Half the atoms replaced with a single new atom type	1
Wash Half All	Half the atoms replaced with new type, for each atom type	5, based on the number of atom types excluding Phosporus

Table 5 Classification of branches and their actions

How branching works is that the saved frame state is taken as a "seed" and then cloned with "mutations" (the branch actions). Each branch type action is applied to its own copy of the seed frame state and new branched simulations created. Therefore, for a starting frame state having ten molecules present (associations of two or more atoms) the total number of branched simulations created would be:

1 + 2 + 10 + 4 + 1 + 5 = 23.

Let us now examine the branch actions in more detail by enumerating the pseudocode of each action.

No-Op

No mutations are made to this branch; simply a single copy of the seed frame state is produced. The pseudocode for this action is:

```
Duplicate final state
Submit
```

Temperature

This branching action simulates gradual external temperature increases and decreases emulating heat infusing or departing the area in a diffuse way (no gradient or source or sink is modeled). A thermostat is used in a separate step to modify the heat bath target temperature to account for excess heat introduced to the simulation due to our naïve bond formation (when atoms bonds are formed too closely GROMACS introduces heat). The pseudocode for this action is:

```
Duplicate final state
Bath temperature set to +1%
Submit
Bath temperature set to -1%
Submit
```

Bond Breakages

Bond breakage is performed to simulate the effects of collisions or radiation on molecules. We perform bond breakage outside of GROMACS in each separate branch of the simulation. The number of branches formed is equal to the number of greater than two sized atoms molecules in the simulation frame. Bonds are created following the GROMACS simulation phase of 1,000 time steps. Bonds are broken in a separate branch. This all occurs outside GROMACS. No bonds are broken during the 1,000 frame simulation, only created (or replaced if a better bond comes along and the atom is at its bonding limit). The pseudocode for this action is:

```
For every bond in the simulation
     Duplicate final state
     Break bond
     Submit
```

Charge Changes

Charge changes are applied (+10% or -10%) are applied each to two randomly selected atoms, creating four simulation branches. Charge changes emulate the external influences of radiation. The pseudocode for this action is:

```
For 0 to less then 2 (loops twice)
     Duplicate final state
     Select random atom
     Increase charge by 10% from final state
     Submit
     Decrease charge by 10% from final state
     Submit
```

Wash Half Random

This branching action replaces half the atoms with a random type, emulating chemical mixing from adjacent sources. This action uses the Wash Generic function with a call-back that selects random types for each replacement molecule.

Wash Half All

This branching action replaces half the atoms with the same type for each type of atom. This also simulates chemical mixing from an external source. The pseudocode for this action is:

```
For each type of atom
     Use Wash Generic with callback that specifies
the current atom type for replacement
```

Wash Generic

This function is used by other wash implementations and the pseudocode for this action is:

```
Duplicate final state
Group atoms into molecules
Randomise molecule order
Select molecules until total number of atoms is
larger then required percentage
For each selected atom
     Change atom properties/type to type selected
by callback
     Remove all bonds the atom is involved in
     Try up to 10 times:
          Give random position
          Test for proximity to other atoms
          If not in bonding range to nearest atom,
stop loop
Submit
```

Note that washes are performed for each completed frame producing a new simulation branch for each, separate from the branches created for the other methods. We replace randomly selected atoms in already formed molecules, as well as single unbonded atoms.

3.4 Results from the EvoGrid Prototype2010

Hardware Implementation and Performance of the EvoGrid

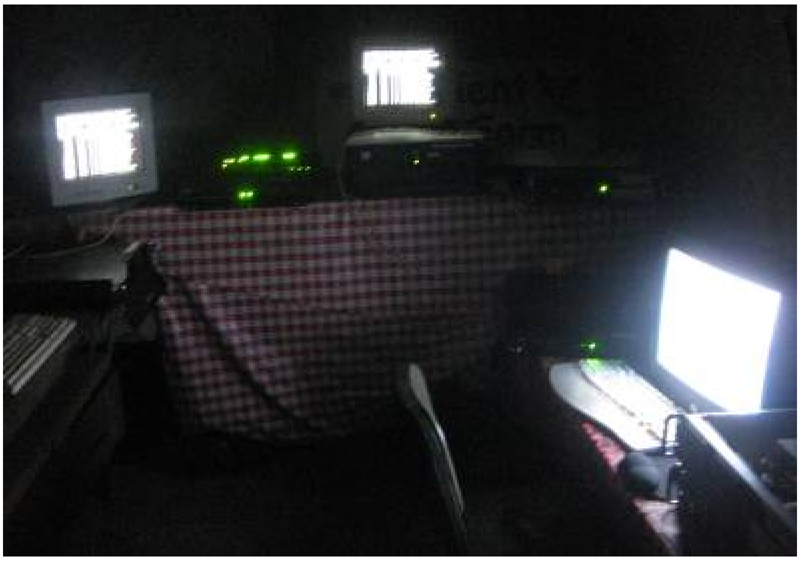


Figure 56 First full configuration of the EvoGrid in a room in the author's barn, in August of 2010

With an explanation of the software architecture and core algorithms behind us let us briefly look at the first two realizations of the hardware of the EvoGrid prototype. Figure 56 depicts three of the half dozen five to ten year old dual core Intel servers configured in the first full EvoGrid prototype network. This network was set up by the author in August

2010 in a room in his barn in northern California. This network ran fairly reliably for eight weeks but experienced the non-recoverable failure of three client machines half way through. The load was taken up by the remaining systems.

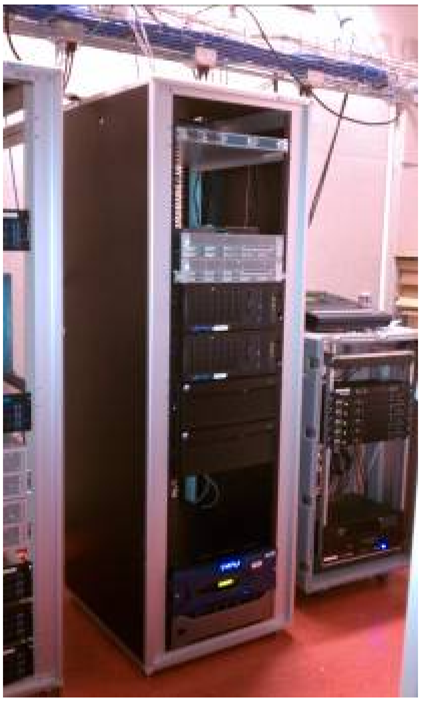


Figure 57 Second full configuration of the EvoGrid running at Calit2 on the campus of the University of California at San Diego

In the late fall of 2010 the staff and management at Calit2, the California Institute for Telecommunications and Information Technology, located at the campus of the University of California at San Diego, agreed to host the 2010 EvoGrid prototype for its next set of trials. The network was set up by January of 2011 and the core system, a Sun Intel i7-based 24 core processor (shown in Figure 57) was set up with seven "slaves" or client systems running as Debian Linux virtual machines on computers within the Calit2 cluster. The first task of the system was to run the complete analysis of the control (Experiment #3) data. This data was analyzed in a matter of days when it would have taken weeks on the home-built first generation EvoGrid. This system was then utilized to run the 4th, 5th, 6th and 7th experiments through to our project completion in May, 2011.

3.4.1 Scaling of the Hardware and Software Services

Prototype 2010 of the EvoGrid ran experiments on three distinct networks, the last two being variations of the software setup on the same installed servers.

Experiments	Clients: function	Cores	SM cores
Exp #1, #2, #3 - DigiBarn	4 (average): 1 sim	1 - native	1 core - analysis
Exp #3, #4, #5 – Calit2	7: 2 sim	2 - VMs	24 VM - analysis
Exp #6, #7 – Calit2	7: 1-2 analysis 2 sim	4 - VMs	24 VM - analysis

Table 6 Hardware configurations and services running for each of the three phases of experimental operations.

Table 6 illustrates the hardware configurations for the seven experiments. The first three experiments were done on the DigiBarn network on an average of four client machines running GROMACS simulations and one dedicated machine to run the simulation manager. The SM was also running the analysis (scoring) daemon service. Each was running with dedicated hardware, using a single core and running natively (with no virtualization).

The UC San Diego Calit2 network was initially configured to run Experiments #4 and #5 on seven clients running two cores each with two simulation services (one per core). The SM was running on a 24 core server but only using a single core for the SM functions and analysis. The machines on this network were configured with faster CPUs and more memory than the DigiBarn systems. However, they were running the EvoGrid through virtual machines (VMs), which brings overall throughput down into the range of running natively on the older hardware. The statistical analysis phase of Experiment #3 (searching for the number and size of molecules) was run on this network.

Experiments #6 and #7 were also run on the Calit2 network but with the seven clients configured to run EvoGrid services in four cores. We took the opportunity to distribute analysis daemons to each of these seven machines. In some cases the client machines were running two analysis daemons each.

In the initial network having four clients running against three experiments there was a *linear* relationship between the size of the network and the number of frames being processed per day. With the addition of more analysis daemons each generation of the experiment's inheritance hierarchy trees was being explored in parallel by a number of analysis daemons before the next generation of branches were made. This permitted a branch to be explored more thoroughly and increased the possibility of finding maxima. On the higher performance Calit2 network the numbers of processed and unprocessed frames for the longest running Experiment (#6) is far higher than for previous configurations. This was attributed to more dedicated computing hardware but also the more sophisticated distribution of simulation and analysis daemons. Future development of systems like the EvoGrid would do well to establish some kind of formal understanding of an optimal balance between simulation and analysis services. When analysis falls behind simulation, branching does not occur, and inheritance hierarchy trees get explored wider, rather then deeper.

In conclusion, the computationally intensive step of simulation scales linearly with the number of physical client computers employed. The use of separate CPU cores improves this by another linear factor. There is no limit in the number of processors for linear scaling save the capacity of the simulation manager to handle their coordination. The use of virtualization carries a cost but enables the use of many more network computing resources. The distribution of analysis daemons, which supports the exploration of simulation results and generation of the inheritance hierarchies, has a large impact on the pace and type of exploration done. The above-described summary of the network and services topology should be kept in mind when considering the results of experiments later in this section.

3.4.2 High Level Analysis of the Optimization Methods at Work

The high level performance of the DigiBarn EvoGrid network can be determined from the data in Table 7. For the first experiment (search criteria: number of molecules formed) 213 frames were processed but of those only 187 fully completed processing (due to crashing of the simulator part way through the frame simulation step) and 11,015 branches were abandoned as incomplete, i.e. without being processed. This yielded a ratio of one frame fully or partially processed for approximately 52 abandoned (i.e. never processed). One argument we are making here for the desirability of applying this kind of methodology to study emergent phenomena in chemical simulations is the saving of computing resources, and therefore realizing a substantial time savings. As the large number of abandoned frames need not have been computed as preset goals were already met, the first EvoGrid experiments seemed to be providing such savings.

Another argument is that without these teleological (goal setting and attaining) methods, desired ends may simply never be reached, or reached in such a long period that research grants (or even the lifetimes of researchers) may by then have run out.

Experiment	#Processed (#Failed) Simulations	#Incomplete Simulations	Complete:Incomplete Ratio	Highest Score
1- num mol	187 (26)	11015	1:51.713615023474	60
2- mol size	148 (39)	40654	1:217.40106951872	4
3 -control	155 (21)	16178	1:91.920454545455	N/A

Table 7 Tabulation of two months of execution of the EvoGrid first prototype network run between August and October of 2010

Continuing to look a the high level results from Table 7 we see that for the other two experiments we measured ratios of 1:217, and 1:91 processed to abandoned frames, respectively. Back of envelope calculations suggests that with the first hardware network, each frame took the following time to compute:

Fully or partially processed frames:
213 + 187 + 176 = 576

Days of operation of first simulation (approximately) =
First full day: August 8, 2010
Last full day of simulation: October 6, 2010
for a total of 64 days or 1,536 hours.

So if 1,536 hours is divided by the total of 576 frames processed we get an approximate time of 2.66 hours or two hours and forty minutes per frame for processing. Of course many of these frames only achieved partial processing so this number is very approximate. Processing of these frames involves simulating 1,000 time steps with 1,000 atoms executed within the GROMACS version 3.3 engine. For the 1,000 time steps GROMACS carried out a diffuse atomistic molecular dynamics simulation involving encounters of atoms which then experienced attractive or repulsive forces. As described previously on the final time step the state of the field of atoms was dumped to disk and a form of naïve bond formation applied and the frame was scored. Other global effects including the above-described branching actions were applied at this time and the resulting new frame states placed onto a queue to be selected for future simulation in one of the distributed GROMACS clients. All of this activity was under the control of the Simulation Manager (SM).

The time taken to do the above steps for each frame on a single processor in a distributed fashion is a guage of the computational load involved in even simplistic prototype simulations. It should be noted that during execution of the molecular dynamics simulations the client system's monitors showed the CPU activity close to 100% committed to this one task. The Simulation Manager ran on a dedicated Debian Linux server and was not so computationally burdened.

3.5 Detailed Low Level Analysis of the Optimization Approaches

We will now take a close look at the results of all seven experiments, beginning with the three initial simulation test runs (two experiments and one control).

3.5.1 Experiment #1: Number of Molecules

EvoGrid prototype Experiment #1 had pre-scripted search criteria designed to optimize the simulation to pursue a vector toward generating frames having the largest number of molecules formed. In the final processed frame (the 187th in the queue to be analyzed) the end products were tabulated after 213 frames were processed (less 26 failures during processing) and over eleven thousand branches abandoned without being turned into processed frames. The "best" frame identified contained nine kinds of molecules in a total population of sixty molecules overall. This constituted the most populated frame.

Figure 58 shows the plot of results with the simulation-sequential processed frames on the x-axis plotted against the number of molecules observed per frame on the y-axis. Note that not shown are the 26 discarded frames due to incomplete computation caused by crashes in the simulation engine. We can see two effects here: one is a consistent climb in the number of molecules per frame to plateaux in the 20s and 30s to a more distinct plateau at around 57 molecules, a "pause" and then resumption to a level of 60 molecules where the simulation stayed until we terminated the computation at the 187th frame. There is a second effect: the occurrence of lower scored frames (the low value "spikes" below the main trend graph). These represented frames whose score retreated due to the breaking of more bonds than new ones that were formed. These frames represented the stock of hierarchical simulation tree branches that were ultimately abandoned.

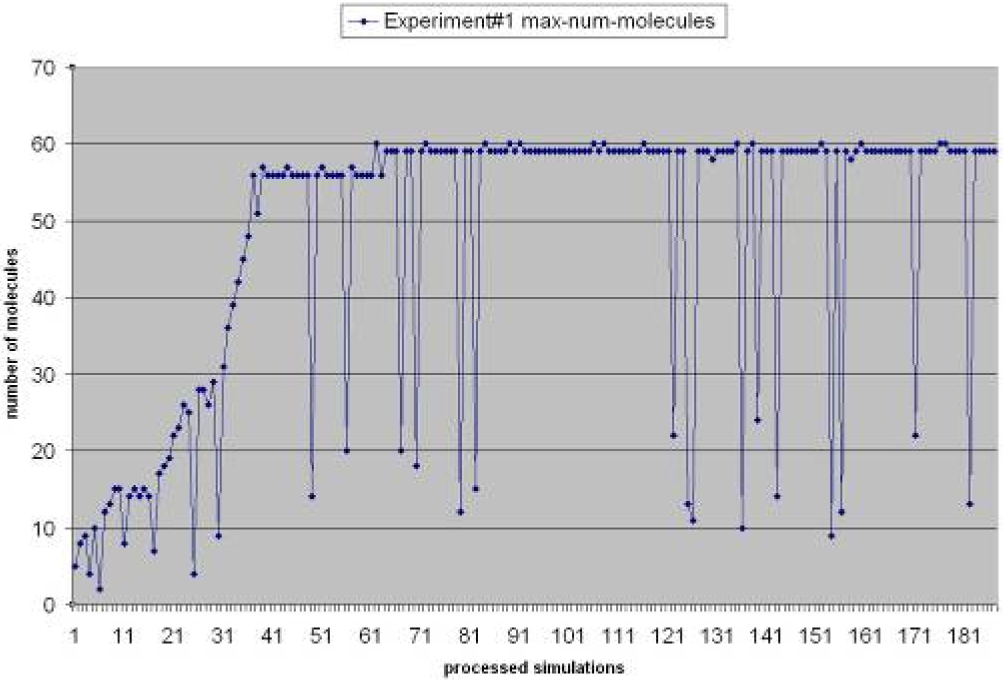


Figure 58 EvoGrid Experiment #1 showing 187 successfully processed simulation frames plotted along the x-axis with the number of molecules observed on the y-axis

Note that Figure 58 represents an aggregate of the frame scores, not a strictly time-linear output of the whole simulation. Figure 59 below presents a view we call the hierarchical inheritance tree, which represents another view of processed simulation frames within the actual order of execution. The topology of this tree, which must be seen from left to right as it is rotated ninety degrees counterclockwise on the page, shows the rapid climb in score (number of molecules) from the earliest simulations followed by the two plateaus. We have extracted a portion of this tree graph to be better viewed in Figure 60.

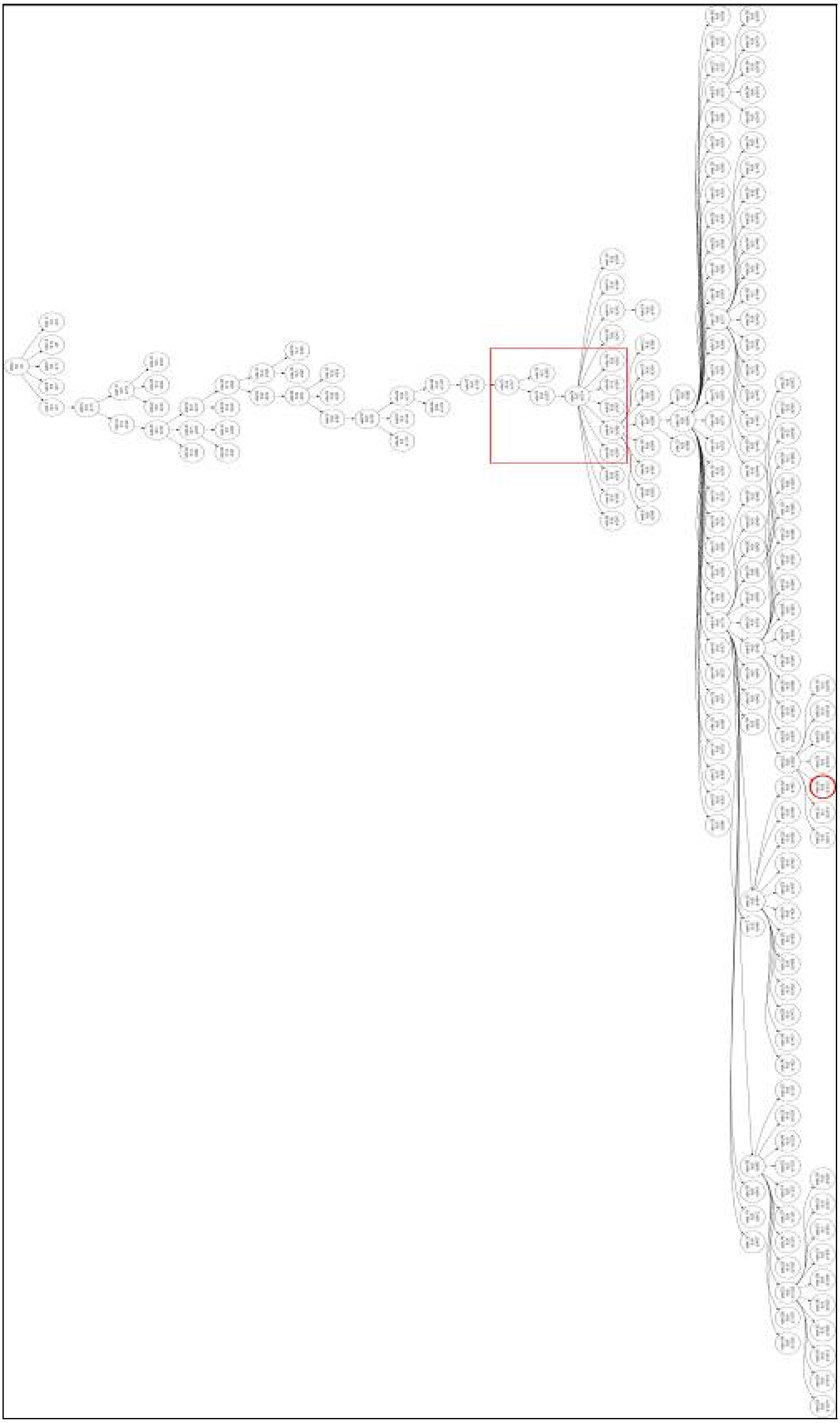


Figure 59 The hierarchical inheritance tree from Experiment #1 showing a section to be extracted for better viewing and a “terminal node” for examination

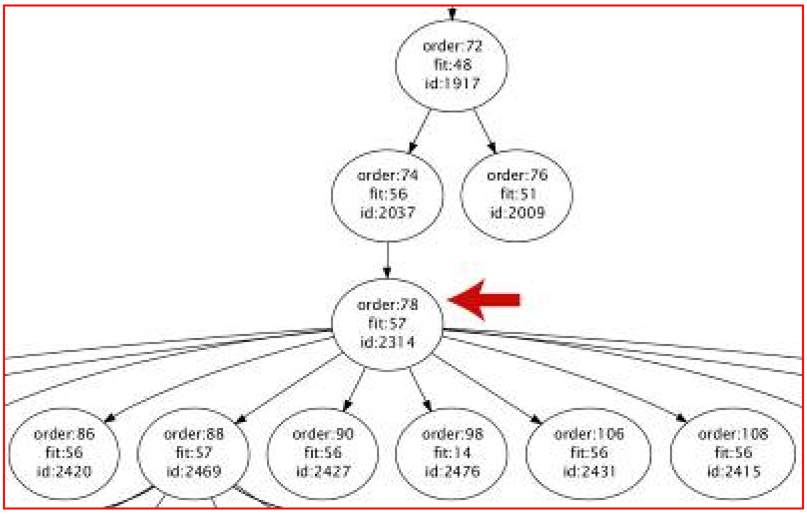


Figure 60 Extracted view of a portion of the hierarchical inheritance tree surrounding simulation ID 2314

Here we can see the critical juncture at the "node" representing the simulation of ID 2314, which is the 2,314th simulation frame generated by the system but only the 78th to be processed (the "order"), the rest having been placed in queues from which they may never be processed. Note that the order number of 78 also includes the 26 failed simulations and 13 simulations which were discarded due to server issues, so this corresponds to the 39th successfully processed simulation and the corresponding point on the chart indicated by the arrow in Figure 61. Note that once on that first plateau one frame experienced a drop to a fitness of 14 creating the first large downward "spike". This branch is soon abandoned, allowing the "better performing" branches to continue.

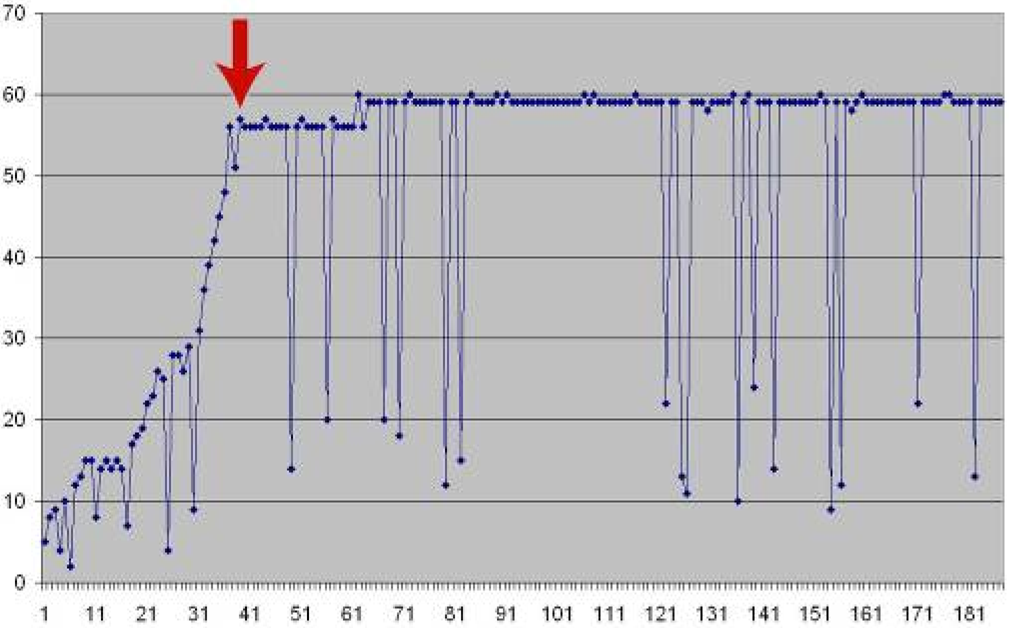
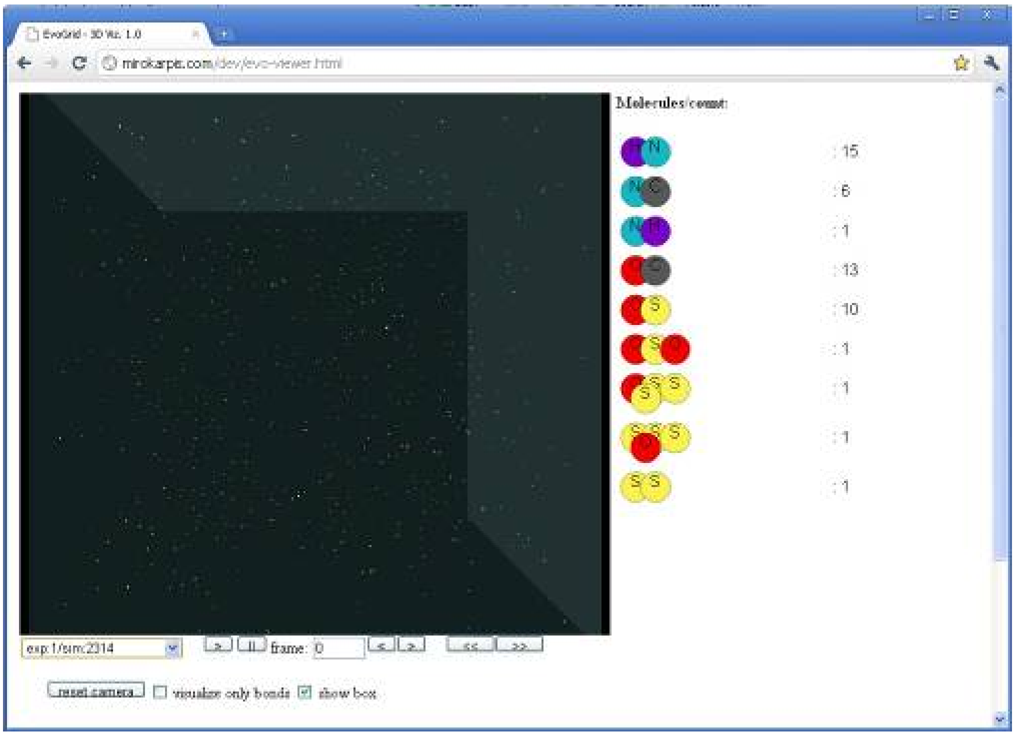


Figure 61 Indication of the presence of the point of simulation ID 2314



Molecules/count:

N	: 15
N C	: 6
N	: 1
C	: 13
S	: 10
S	: 1
S S S	: 1
S S S	: 1
S S	: 1

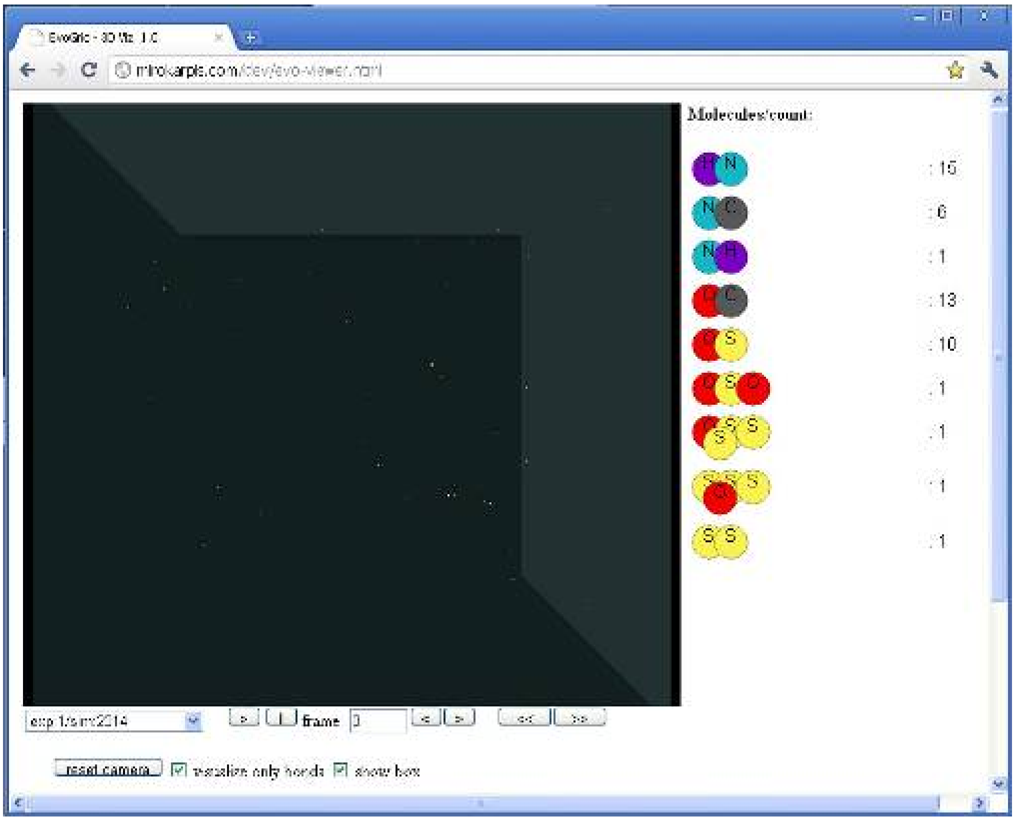
exp 1/sim 2314

frame: 0

☐ visualize only bonds ☑ show box

Figure 62 3D visualization of Experiment #1, simulation ID 2314

A third view of this data was created by the programming of a WebGL-based 3D visualization system shown in Figure 62. This is shown running in the Google Chrome (beta) web browser. On the left is represented a full 3D placement of all 1,000 atoms and the molecules formed by the time that frame is processed. On the right is a table enumerating the type and population of molecules formed using our naïve bonding mechanism for the particular simulation of experiment/simID# indicated in the lower left. Also below the main 3D window are controls to permit the loading of alternative simulation data from the simulation manager, the playing of a “movie” representation of the action in subsequent frames and an interface to position the viewer’s camera near the notional molecules. There are additional controls to turn on or off the view of the bounding box of the simulation and to show only bonded molecules.


EvoGrid - 3D Viz 1.0
mirokarpis.com/dev/evo-viewer.html
exp 1/sim2314
frame 3
reset camera
☑ visualize only bonds ☑ show box

Molecules/count:

H N	: 15
N C	: 6
N H	: 1
O C	: 13
O S	: 10
O S O	: 1
O S S S	: 1
S S S O	: 1
S S	: 1

Figure 63 3D visualization of Experiment #1, simulation ID 2314 with only bonds shown

Figure 63 shows data for the simulation of ID number 2314 with only bonds (molecules) displayed. We note that by this simulation branch there are nine kinds of molecules with forty-nine being the total population of all molecules (having two or more atoms as members). Figure 64 and Figure 65 depict how the user can move an observer camera to view a particular molecule within the 3D scene of simulation ID 2314. In this case we are viewing a notional hydrogen-nitrogen association.

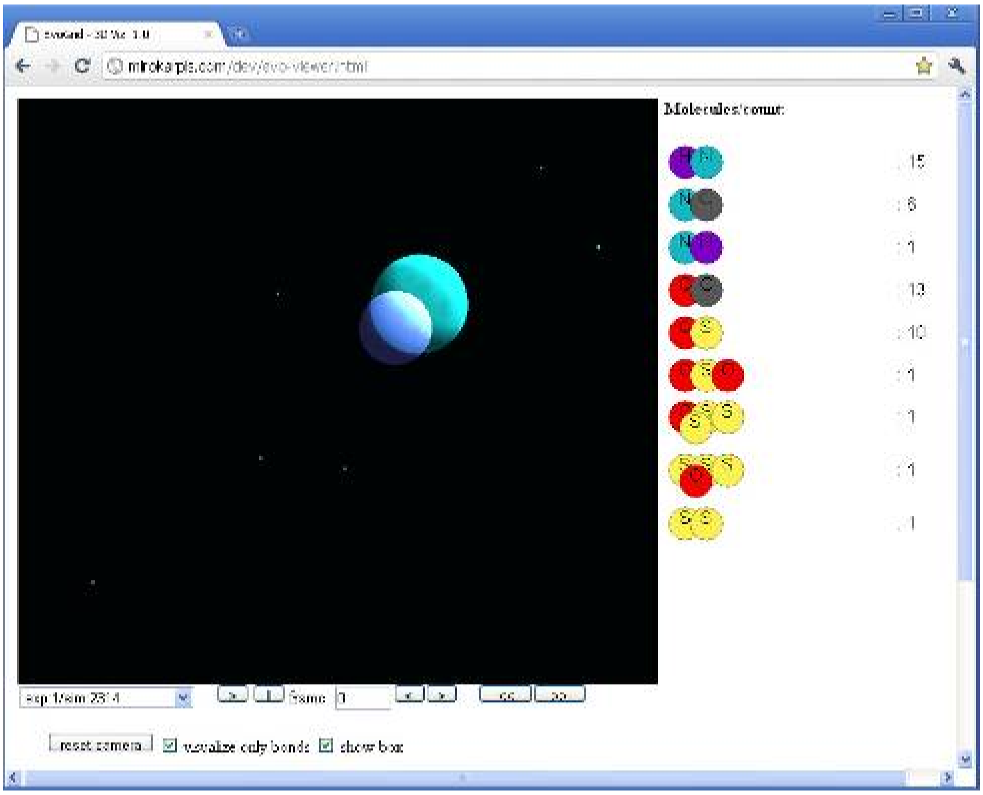


Figure 64 3D visualization of simulation ID 2314 with the observer camera zoomed to show one of the molecular products

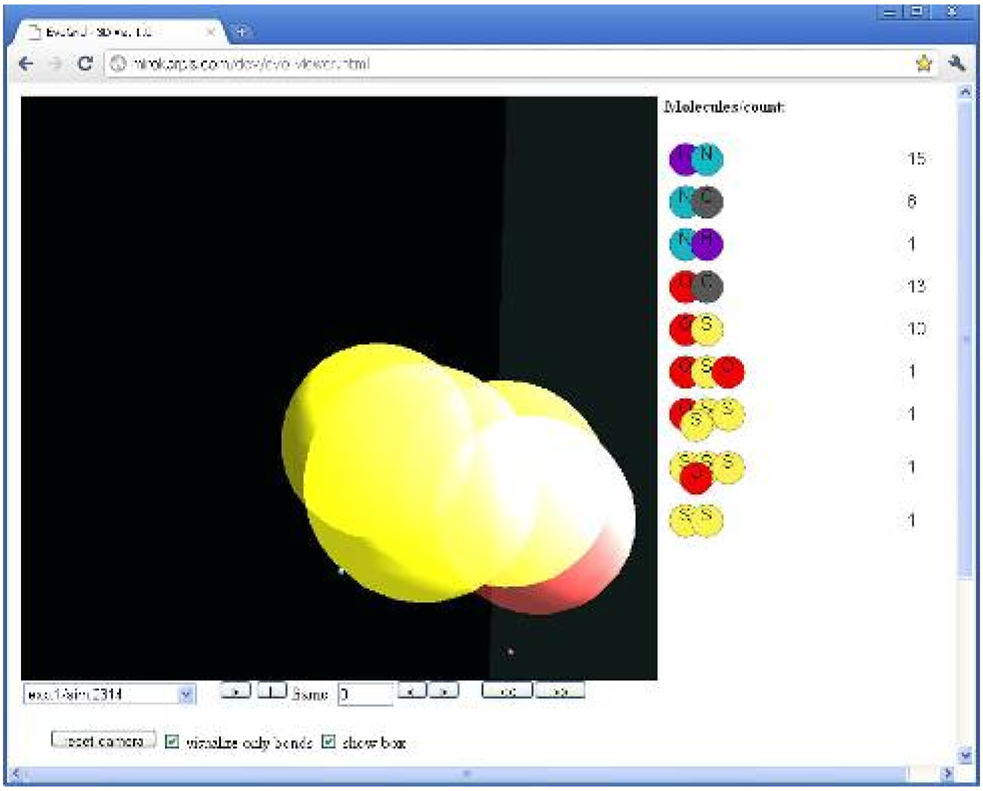
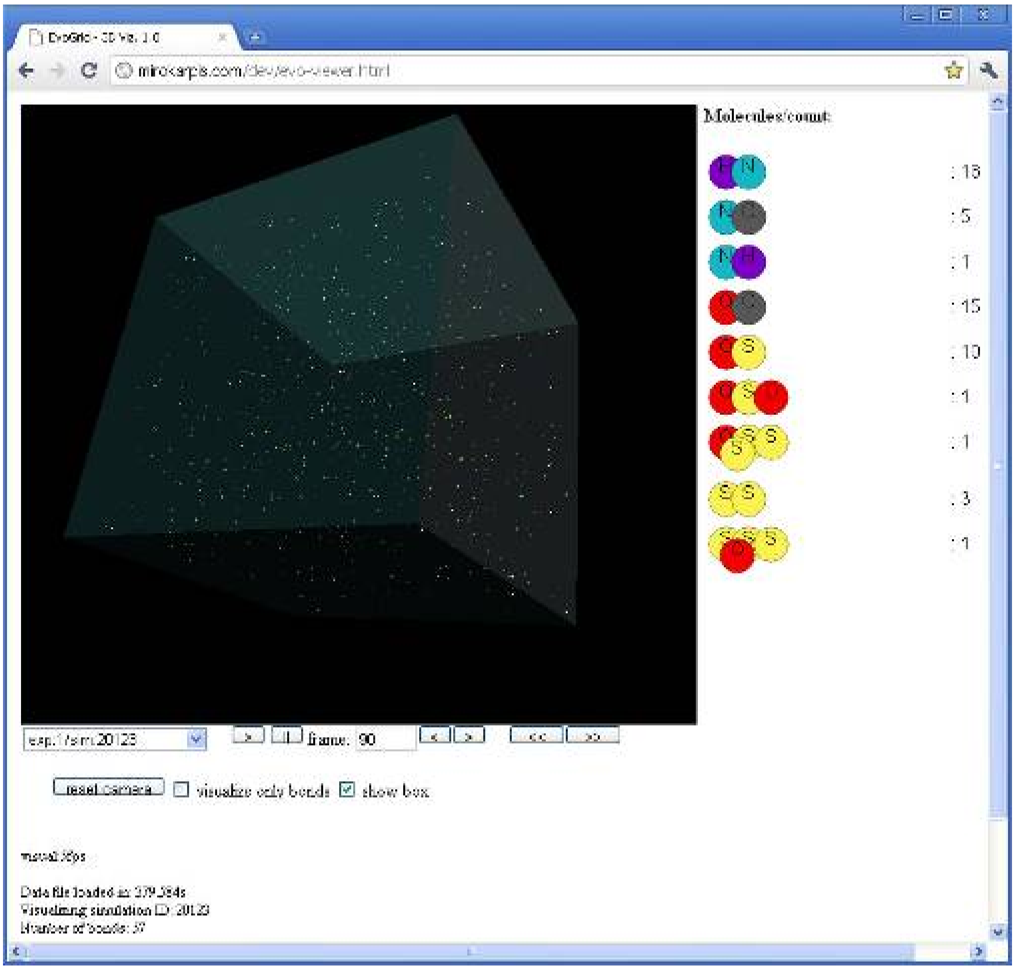


Figure 65 3D visualization of simulation ID 2314 with the observer camera zoomed to show one of the larger molecular products

Figure 66 depicts an aggregate view of processed simulation of ID 20123 that occurs toward the bottom of the inheritance hierarchy tree for Experiment #1. This is the node circled in red in Figure 59.



Molecules/count:

H N	: 16
N C	: 5
N H	: 1
O C	: 15
O S	: 10
O S O	: 1
O S S S	: 1
S S	: 3
S S S O	: 1

frame: 90

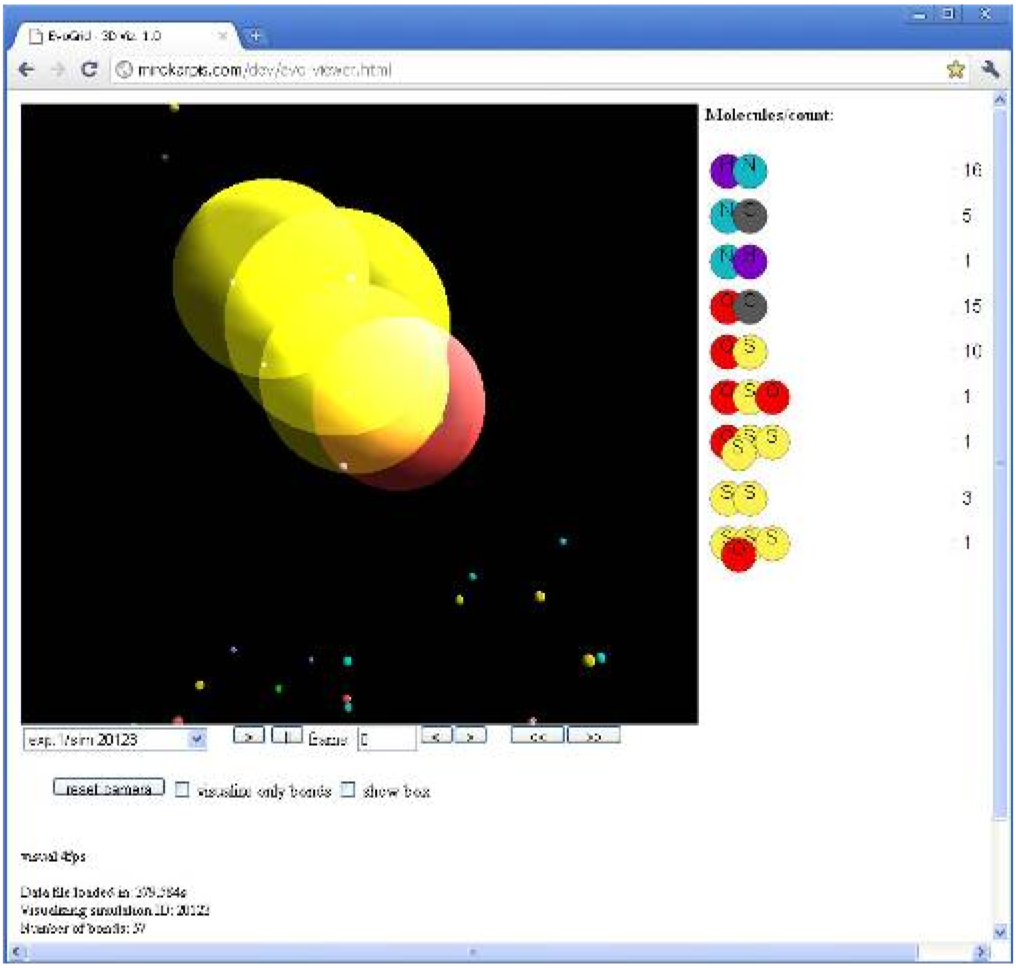
reset camera □ visualize only bonds ☑ show box

visual: 30fps

Data file loaded in: 379.384s
Visualizing simulation ID: 20123
Number of bonds: 57

Figure 66 3D visualization of simulation ID 20123

The larger molecular product shown in Figure 67 is a sulfur-sulfur-sulfur and oxygen association. It should be noted that by this simulation there are still nine type of molecules but now fifty-two in total with three molecules having four members, versus two for simulation ID 2314.



visual 4fps

Data file loaded in 379.384s
Visualizing simulation ID: 20123
Number of bonds: 57

Figure 67 3D visualization of simulation ID 20123 with the observer camera zoomed to show one of the larger molecular products

It may have been noted by the reader that there is an absence of phosphorus in any formed molecule. Phosphorus contributes to the Lennard-Jones forces (when non bonded), but there is no atom that can bond with it in our simulations.

3.5.2 Experiment #2: Size of Molecules

EvoGrid Prototype2010 Experiment #2 had pre-scripted search criteria designed to optimize the simulation to pursue a vector toward generating frames having the maximum molecular size. Our definition of “molecular size” is the count of atoms within the series of bonds that connect the longest string of atoms in a molecule. Table 8 illustrates this for several cases.

Molecule Shape	Max #Bonds in longest string / Molecular Size
O – O	1 / 2
O – O – O	2 / 3
O – O – O | O	2 / 3
O – O – O | O	3 / 4

Table 8 Definition of “molecular size”

In the final processed frame (the 148th in the queue to be analyzed) the end products were tabulated after 187 frames were processed (less 39 failures during processing) and over forty thousand branches abandoned without being processed as frames. The “best” frame identified contained molecules of measured size 4 in a total population of 119 molecules overall. This constituted the frame with the highest number of the largest molecular assemblages. The number of molecules was not specifically stored for this experiment, only observed in the database during execution. Later experiments stored both the number or molecules and molecular size for all experiments.

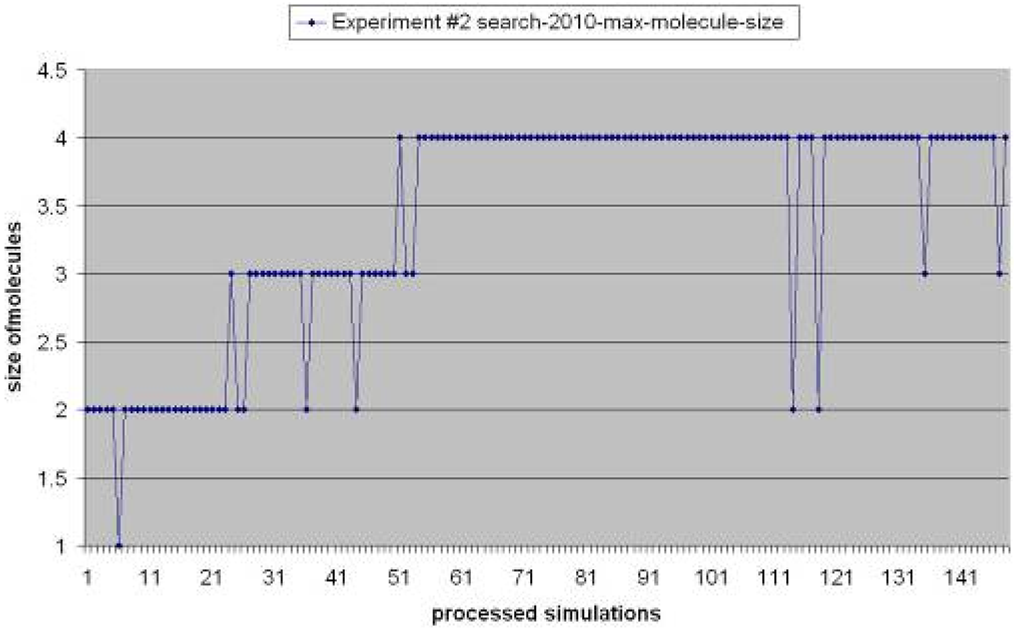


Figure 68 EvoGrid Experiment #2 plotting 148 successfully processed simulation frames along the x-axis with the size of molecules observed on the y-axis

Figure 68 illustrates the time sequence of the occurrence of frames with larger and larger molecules present. As we can see there is an immediate move within the simulated frames to form at least one bond (molecular size of 2) and then some time passes before the plateaus of molecules of 3 and then 4 size ratings occur. There is a long plateau where molecules of size 4 abound but no further growth occurs.

Figure 69 shows the entire hierarchical inheritance tree for Experiment #2 with a section to be extracted for easier viewing. Viewed left to right as this graph is shown rotated on its side, we can see that the topology of this experiment is “bushier” than that of Experiment #1. This indicates more “exploration” of options earlier on before reaching a stable plateau.

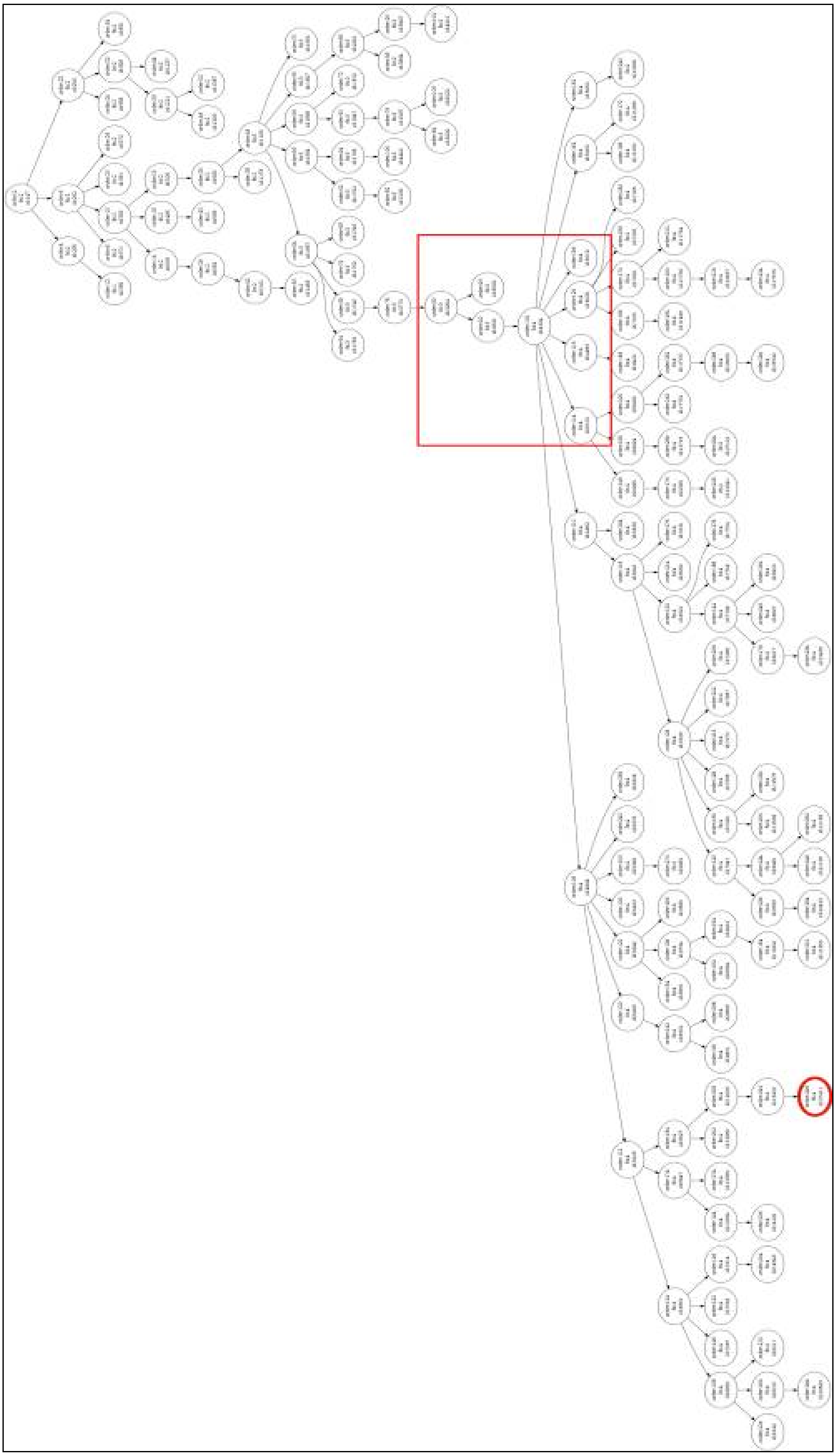


Figure 69 The hierarchical inheritance tree from Experiment #2 showing a section to be extracted for better viewing and a “terminal node” for examination

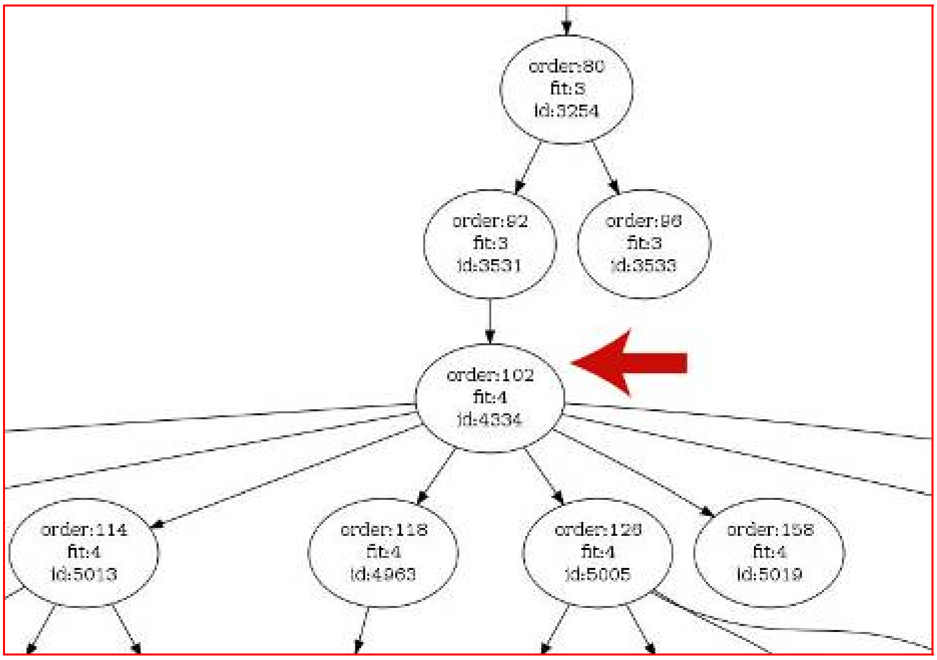


Figure 70 Experiment #2, frame ID 4334 where transition to fitness 4 (molecular size of 4) occurs

Figure 70 depicts the critical juncture at the “node” representing the simulation of ID 4334, which is the 4,334th simulation branch generated by the system but only the 102nd to be processed (the “order”). Note that as in experiment #1 the order number does not factor in failed frames. Figure 71 shows this “inflection point” which is point number 51 where the final plateau starts. This view illustrates the climb from the “fitness” score of a molecular size of 3 to the final plateau of molecules of size 4. On that final plateau several frames still experience declines of maximum molecular size (to 3 and 2 in some instances, not shown in this view) but these branches are abandoned, allowing the “better performing” branches to continue. In this way the “trend” toward larger emergent molecular structures is preserved.

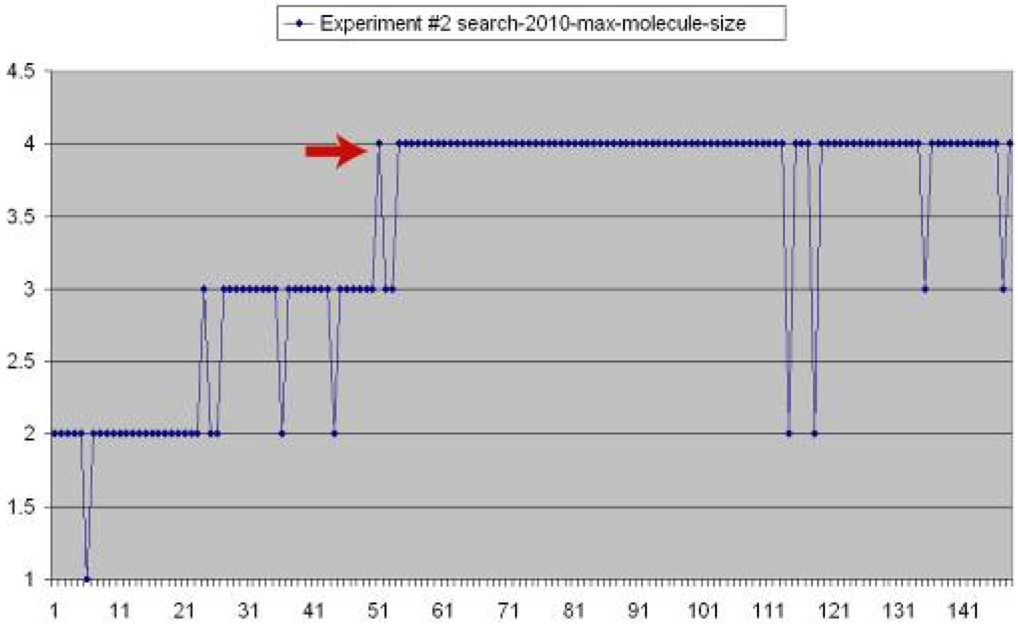
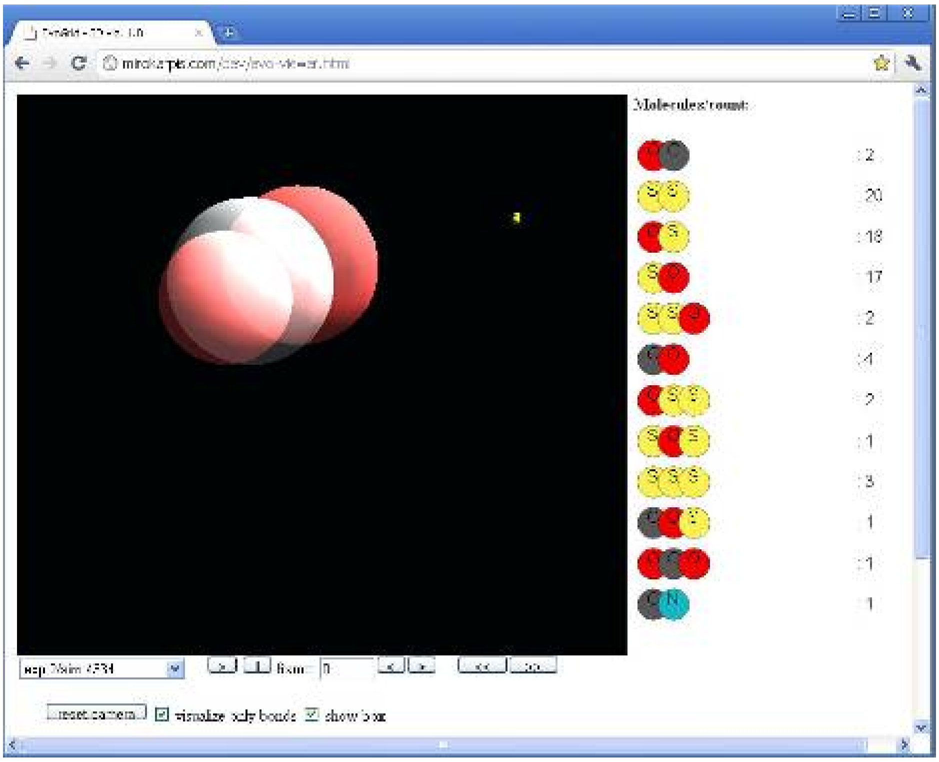


Figure 71 Experiment #2, frame ID 4334 at the inflection point where the maximum molecule size jumps to 4 and stays on that plateau with a few subsequent (abandoned) branches trending to lower scores

Figure 72 and Figure 73 show how the data for Experiment #2 are visualized in the 3D interface. In this case Figure 72 shows twelve types of molecules and 71 molecules total. At that particular simulation ID there are as yet no molecules formed above a size of three.

Figure 73 is a 3D view of the “terminal” node at the bottom of the tree in Figure 69 at simulation ID 27411. This is one of the richest frames in the experiment with sixteen types of molecules and ninety nine molecules in total.



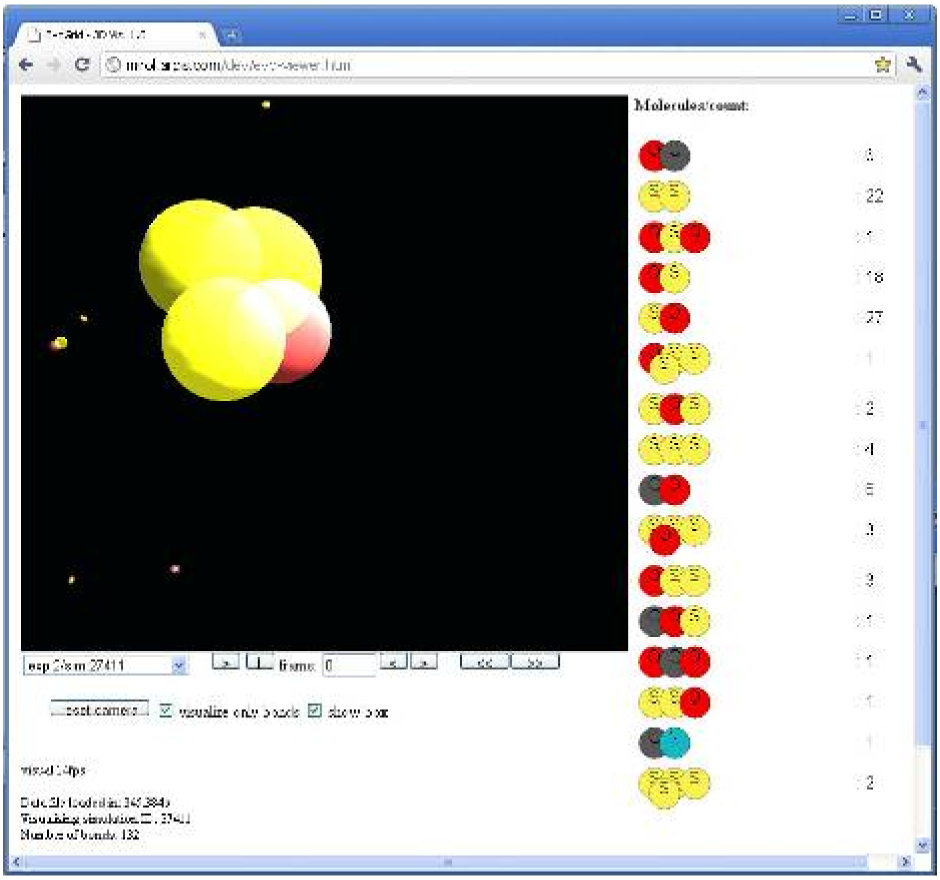
Molecules/count:

O C	: 2
S S	: 20
O S	: 18
S O	: 17
S S O	: 2
C O	: 4
O S S	: 2
S O S	: 1
S S S	: 3
C O S	: 1
O C O	: 1
C N	: 1

exp 2/sim 4334 frame: 0

reset camera ☑ visualize only bonds ☑ show box

Figure 72 Experiment #2, 3D view of frame ID 4334 which does not yet show molecules with a size of 4



Molecules/count:

O C	: 3
S S	: 22
O S O	: 1
O S	: 18
S O	: 27
O S S S	: 1
S O S	: 2
S S S	: 4
C O	: 5
S O S	: 3
O S S	: 3
C O S	: 1
O C O	: 1
S S O	: 1
C N	: 1
S S S S	: 2

exp 2/sim 27411 frame: 0

reset camera ☑ visualize only bonds ☑ show box

visual 14fps

Data file loaded in: 343384s
Visualizing simulation ID: 27411
Number of bonds: 132

Figure 73 Experiment #2, 3D view of frame ID 27411 which now shows molecules with a size of 4

3.5.3 Comparing the Experiments and a Significant Early Result

It is also valuable to compare the inheritance hierarchy trees for both experiments. Experiment #1 hit a local maximum by chance (by random selection) and explored it thoroughly (see left side of Figure 74). Also by chance, Experiment #2 hit a better local maximum and was able to reach a richer end point in fewer steps (see right side of Figure 74). The vertical depth of these trees is indicative of early successes in finding maxima and in not having to explore widely before subsequent maxima are found. Later on (lower in the tree) it becomes more challenging to find new maxima, so we get the effect of a plateau of complexity in the system. We will see differences in the shapes of these trees in subsequent experiments.

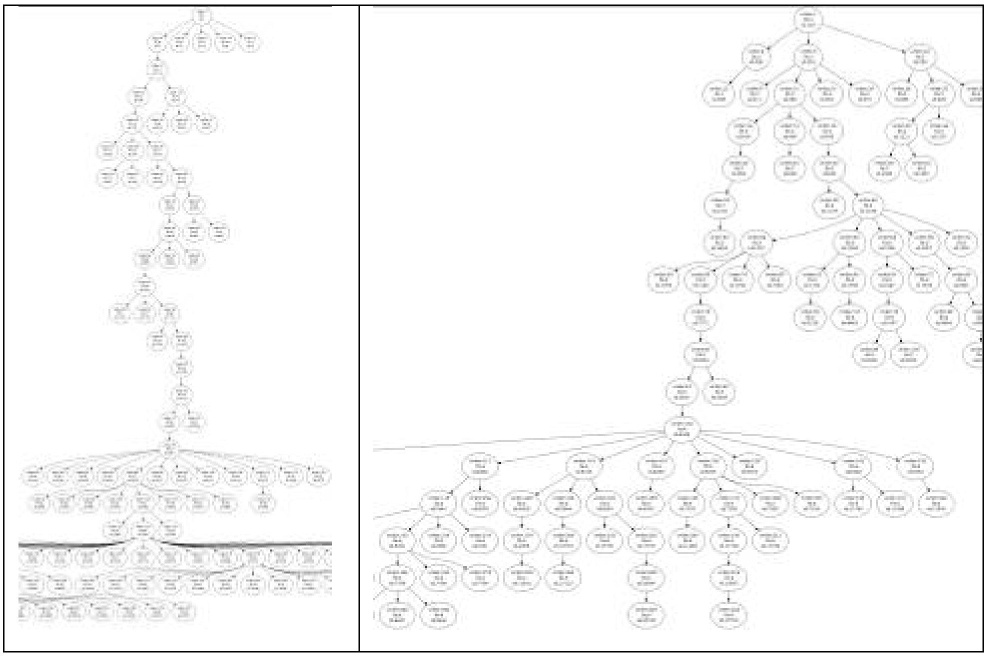


Figure 74 Inheritance Hierarchy topology of Experiment #1 and Experiment #2

One surprise that was encountered is that while Experiment #2 was designed to create the *largest* molecules in fact created *more* molecules. There was one frame with 126 molecules generated by Experiment #2 versus the final plateau of around 60 molecules for Experiment #1. This result was discovered by individually sampling the simulation database as at this early stage in our prototype we were not storing the data for all the scores. This unexpected result has a logical explanation: by setting the teleological goal of seeking larger molecules, *we conditioned the system to generate more types of molecules, and, as it turned out, more molecules overall as it "attempted" to seek ever higher local maxima containing larger molecules*.

This is a significant result which showed that even though we as designers had the best intentions of creating sought-after goals, the dynamism of a system modeled crudely upon the highly dynamic environments one might encounter in nature created another effect. This result suggests that the results intended by our "intelligent design" were confounded by unpredictable emergent phenomena. As explored in Chapter 1, this is a desired property of any system whose goal is to become a useful experimental tool in complexity systems or in biology.

Experiment #3: Control (No Directed Search)

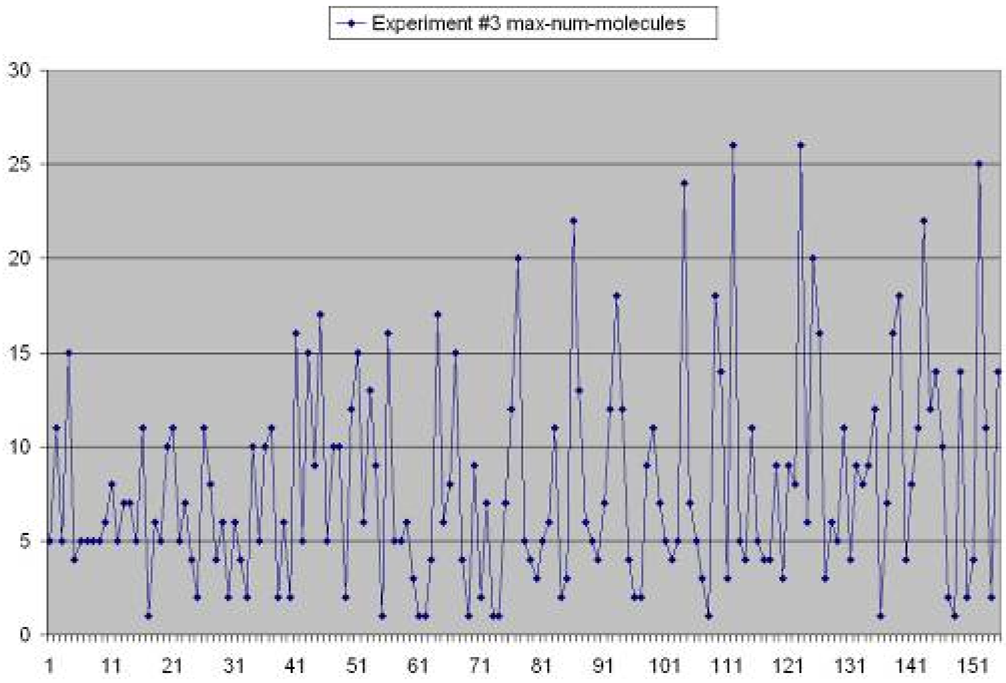


Figure 75 Experiment #3 control data showing the number of molecules observed in simulation frames

Experiment #3 was a parallel *control* experiment which simply processed frames chosen at random from randomly generated offspring in a “linear” fashion with no search branching. This experiment produced some frames with a significant number of molecules (see frames containing over twenty molecules in Figure 75) but no molecules with more than a single bond. Indeed, due to the fact that bonds are broken as well as being formed, without the search driving the process of stochastic hill climbing the frames of the control experiment often shifted to lower states of complexity without a rapid return to increasing complexity as was seen in Experiment #1. There was no identifiable level or ratchet visible in the control results but there is an obvious slight increase in the average number of molecules through time. As Experiment #3 included time-consecutive simulations where a simulation was started from the end state of a previous simulation, subsequent frames will eventually develop more complexity than the first parent simulation. However, the rate of increase of this is lower then that of our directed search. Since the control experiment used the same branching functions, it has the same chance of complexity increasing, if all possibilities were simulated.

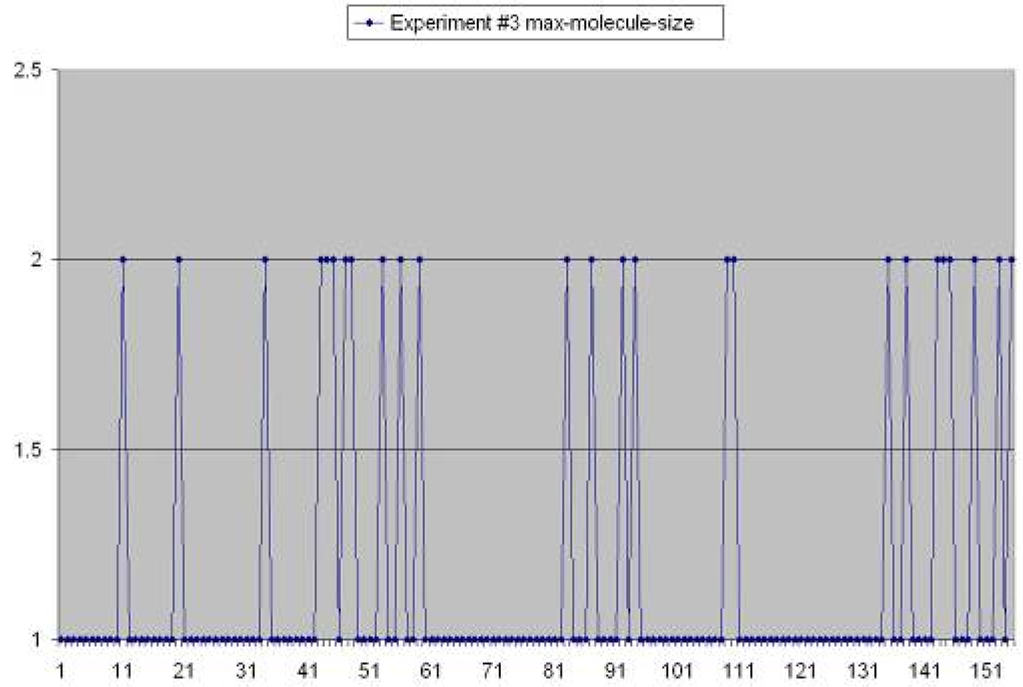


Figure 76 Experiment #3 control data showing size of molecules observed in simulation frames

Figure 76 plots the molecular size data which showed the occasional formation of molecules of size two (one bond and two member atoms) but no overall trend in this data despite the upward trend in the number of molecules.

Figure 77 shows the inheritance hierarchy tree for Experiment #3 turned counter-clockwise on its side. A general observation is that this tree is both broad, and deep, as the space is being explored at random, with no search directing selection of prioritized simulation branches. So while the space is explored there is no predisposition to climb local maxima.

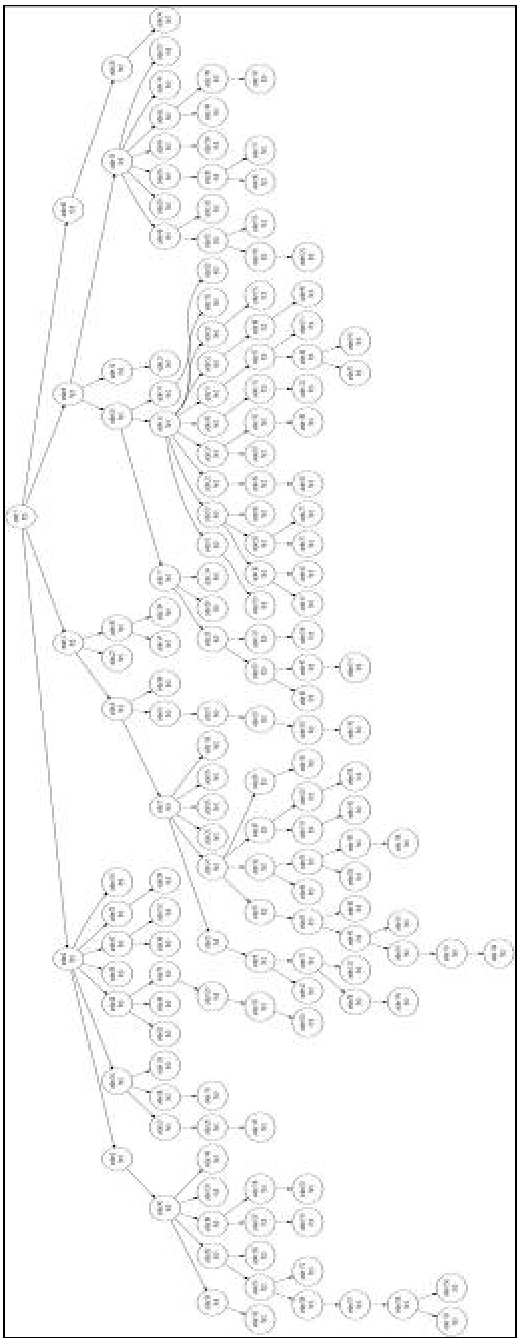


Figure 77 Inheritance hierarchy tree for Experiment #3

3.5.4 Experiments #4 and #5

Having established that directed search clearly is generating superior results over the control experiment the author determined that it was time to instance two new experiments, with a modification to the simulation search and branching criteria. Working with the project team the logic was developed to include a degradation factor of 90% to be included in the score calculation (see pseudocode in Table 4). The pseudocode for this factor is included again below:

```
Set Priority = Parent Priority *
               (0.9 * exp( Fitness Score -
                      Parent Fitness Score ) )
```

This formulation will degrade the priority by the above factor of the difference between the newly calculated score and the parent score. This modification was made to produce the effect in theory, that once achieving a local maximum, the system would more quickly exit that maximum and thereby traverse adjacent "valleys" and be able to achieve a higher maximum in another part of the fitness landscape.

As indicated in the pseudocode in Table 4 another change to this pair of experiments was to turn the *selection daemon* off. This is the facility that randomly selects from available high priority simulations which one to actually simulate. With the selection daemon off, the system would therefore process every simulation. While computationally much more intensive, it was felt that with the degradation factor in place and that by exploring the entire space, there might be a better chance of reaching higher maxima.

Experiment	**#Processed (#Failed) Simulations**	**#Incomplete Simulations**	**Complete: Incomplete Ratio**	**Highest Score**
4-num mol	158 (39)	5884	1:29.868020304569	33
5-mol size	71 (42)	38820	1:343.53982300885	2

Table 9 Tabulation of EvoGrid Experiments #4 and #5

As Table 9 indicates a significant number of frames were simulated but not as high a number as for the first experiments. In addition there were a higher number of failed simulations. This was in part due to failures in the virtual machine setup at the U.C. San Diego Calit2 facility. The highest scores for both experiments were approximately half the values as for the first two experiments. This experiment was run for approximately the same time (eight weeks) on a hardware grid of similar power to the dedicated network of Experiments #1 and #2. Table 6 lists this hardware configuration which, while more clients were committed, the simulations were run via virtual machines running on host operating systems, which imparts some performance cost. Let us now analyze the output of both of these experiments.

3.5.5 Experiment #4: Number of Molecules

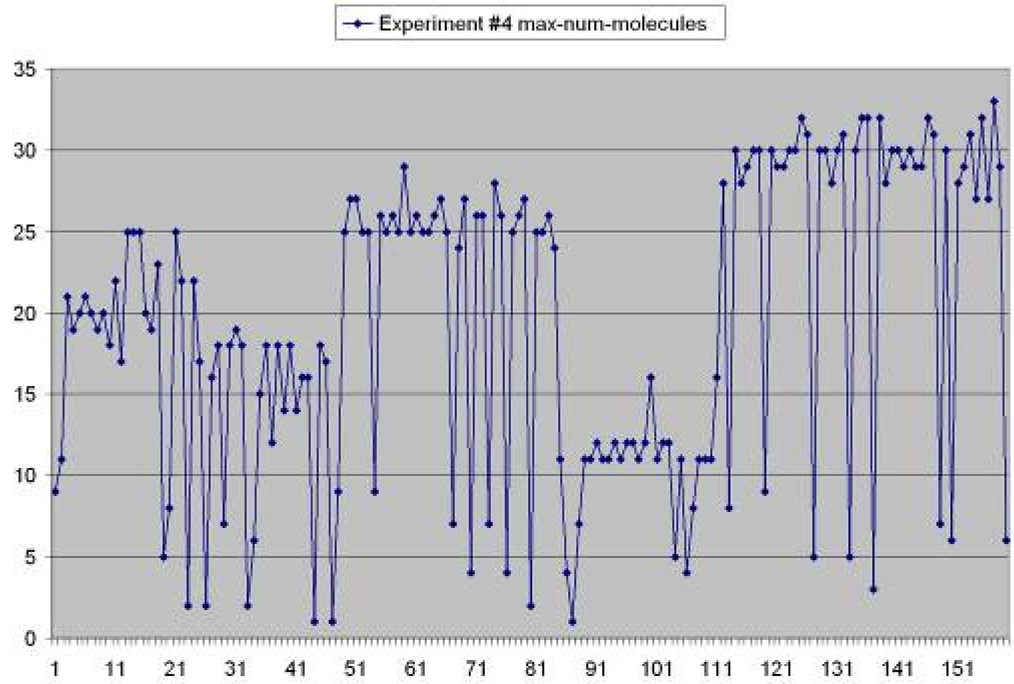


Figure 78 Experiment #4 plot of number of molecules occurring over processed simulations

Figure 78 shows the observation of populations of molecules in simulation frames, with a relatively high population (25) achieved early, followed by a fall-off and then subsequent attainment of higher plateaux. There seems to be a trend toward higher maxima punctuated by significant times in minima. The experiment might have showed a longer trend toward higher maxima but we determined that the run time of two months was the maximum time we could commit to it. What is clear here is that there is no sustained upward trend with only short term reversals as was seen in Experiment #1 for instance. Reversals are deep and longer lasting.

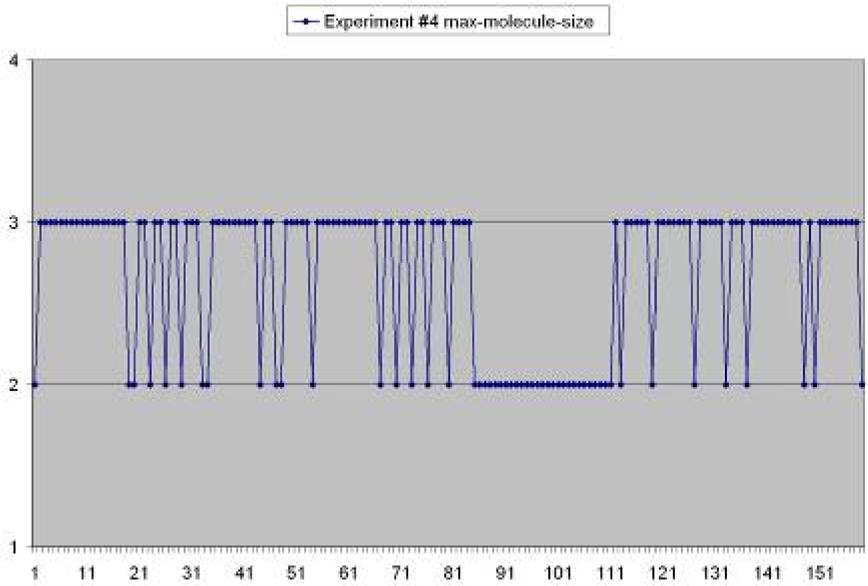
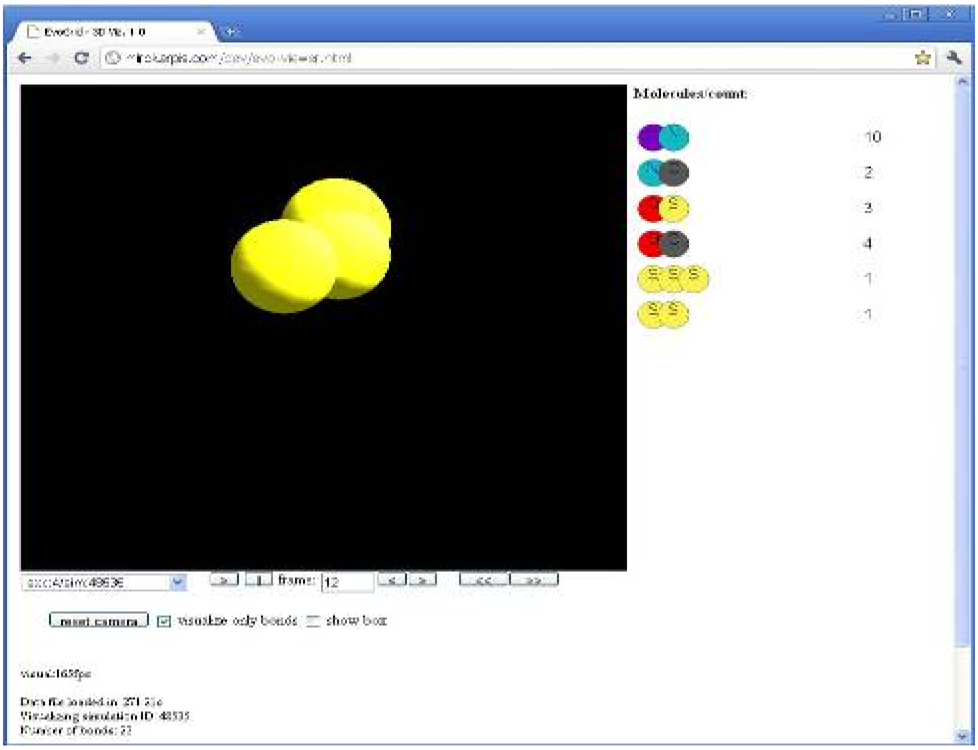


Figure 79 Experiment #4 maximum molecular size

Looking at Figure 79 for maximum molecular size we see that molecules of three members were frequently observed as is shown in Figure 80 below. There does not seem to be any sustained upward trend, however.



visual:163fps

Data file loaded in: 271.21s
Visualizing simulation ID: 48535
Number of bonds: 22

Figure 80 Experiment #4, 3D view of molecule consisting of three sulfur atoms together with a population of 21 molecules formed at simulation ID 48535

3.5.6 Experiment #5: Size of Molecules

Ironically the experiment set up to search for and process simulations containing larger molecules produced only populations of molecular size two (see Figure 81).

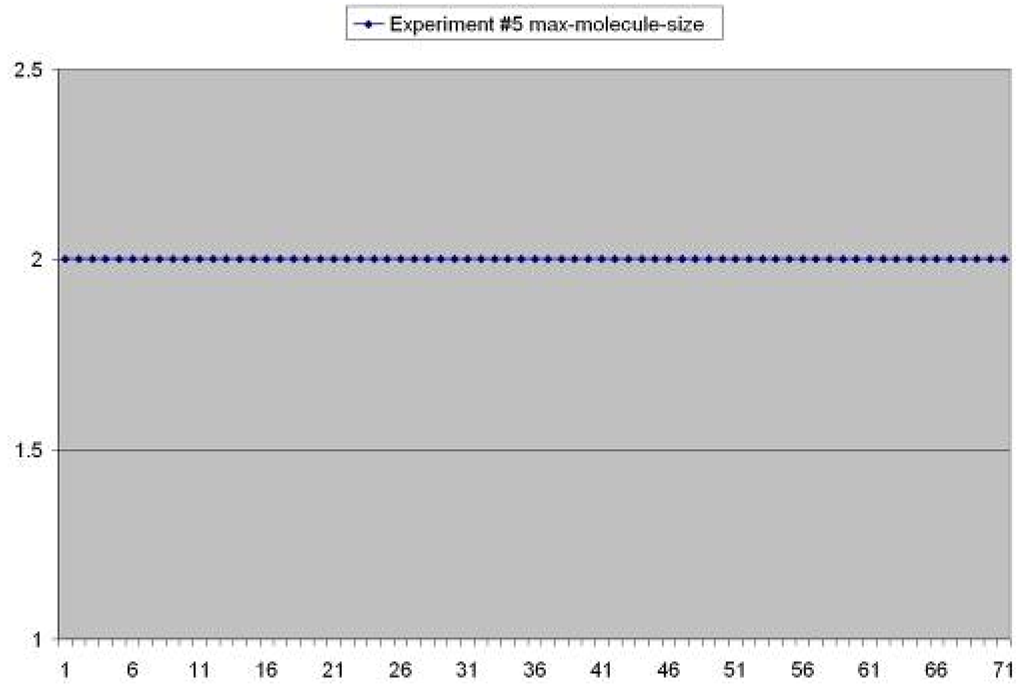


Figure 81 Experiment #5 size of molecules observed over all processed simulations

When studying the population of molecules produced in this experiment (Figure 82) we see absolutely no trend, except perhaps degradation in the simulation's ability to find frames with much molecular population at all.

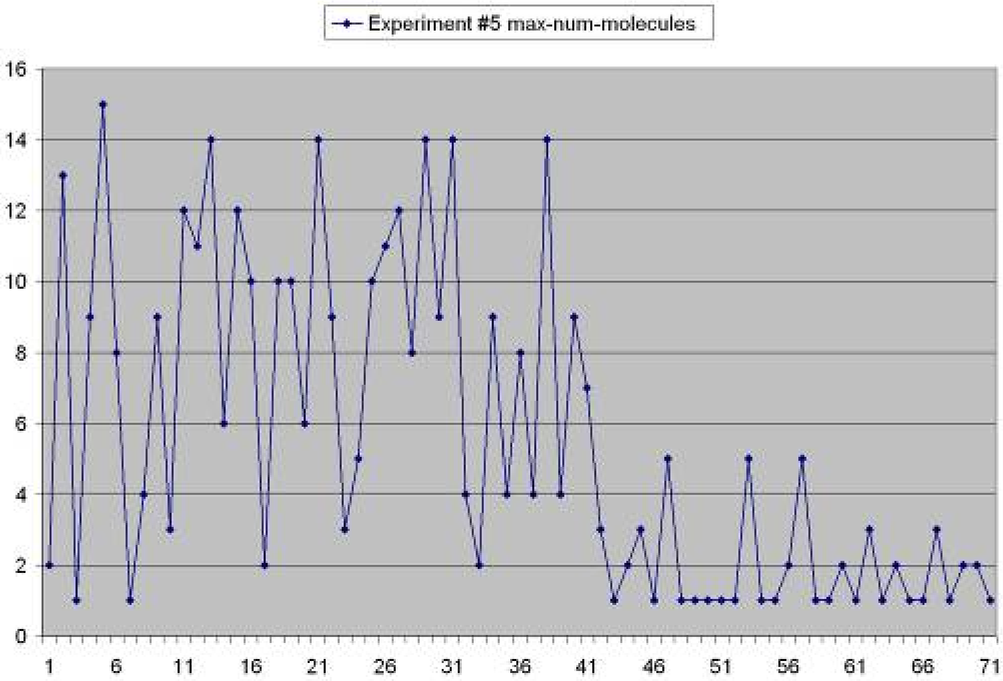
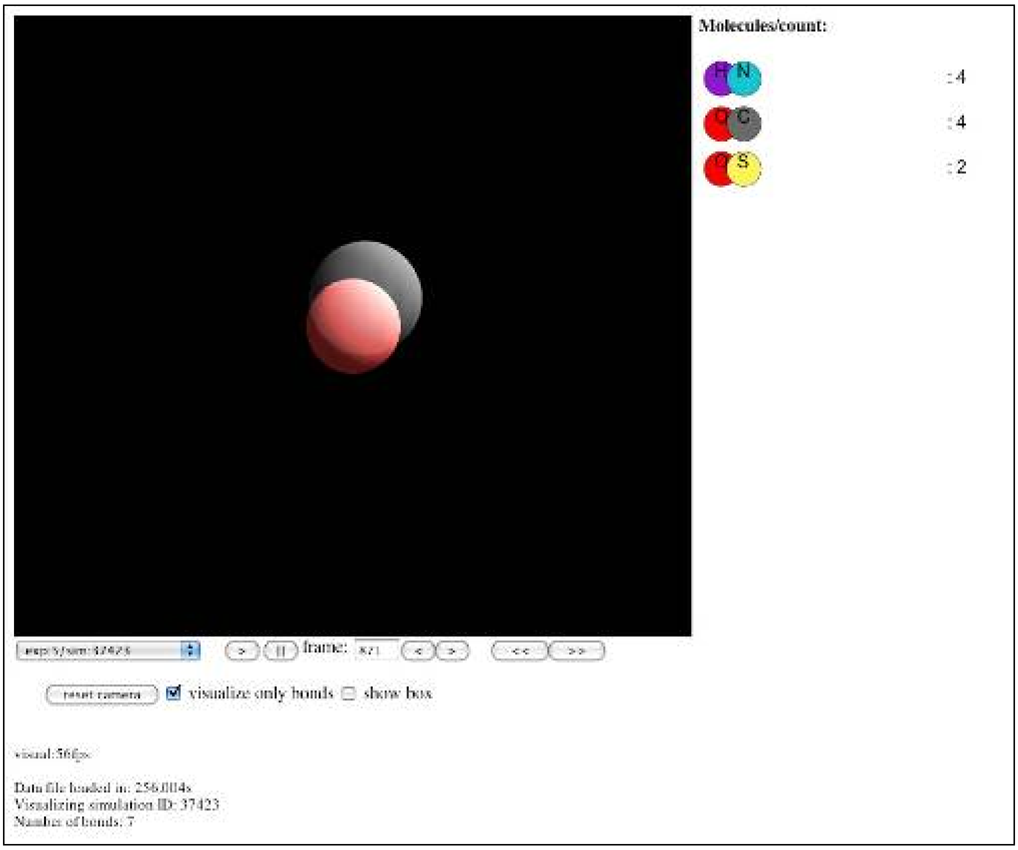


Figure 82 Experiment #5 number of molecules observed over the simulation period

Figure 83 shows one of the highest scored frames in Experiment #5 with ten molecules each of size two.



Molecules/count:

H N :4
O C :4
O S :2

☑ visualize only bonds □ show box

visual:56fps

Data file loaded in: 256.004s
Visualizing simulation ID: 37423
Number of bonds: 7

Figure 83 One of the highest scored simulations in Experiment #5

Figure 84 may provide further insight. The generation hierarchies for Experiments #4 and #5 are rotated on their sides and shown slightly cropped with the first processed simulation on the left side. When comparing these with the hierarchy trees for Experiment #1 (Figure 59) and #2 (Figure 69) what stands out is how broadly distributed the frames are. The earlier experiments show a vertical race to the maxima while these trees tell a tail of broad and costly searching. Given that the selection daemon was inactive, this would result in a large number of branches produced at the “same level” of relative score, hence the breadth of the tree branches.

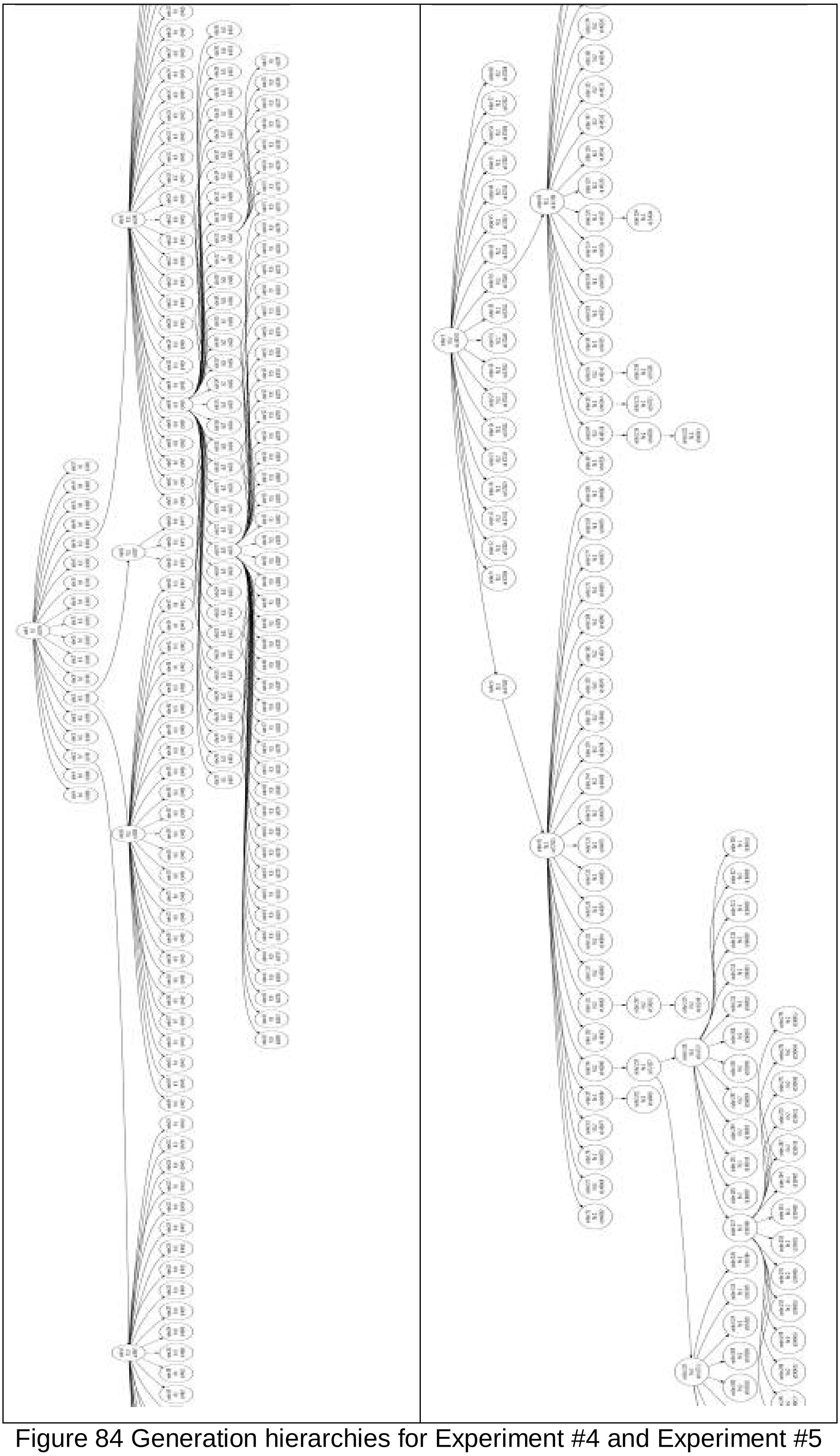


Figure 84 Generation hierarchies for Experiment #4 and Experiment #5

In concluding this round of experiments we can only surmise that the strategy of processing every frame, rather than choosing a smaller proportion of available high priority frames at random, failed to allow the simulation to consistently attain greater maxima and may have led to being trapped in minima for long periods. In addition the degradation factor may have contributed to faster falling off of maxima. In summary, this configuration was unable to explore as much space and what maxima were found were soon lost and therefore not able to be leveraged as a minimum starting point for the assault on higher peaks.

3.5.7 Experiments #6 and #7

It was determined that the restarting of the selection daemon to permit a sparser selection of random high priority frames might overcome some of the challenges faced in Experiments #4 and #5. The degradation factor was left in place. The hardware network, as described in Table 6, shows that we utilized the same number of physical computers (7 clients and one server) but we utilized the additional processor cores on these machines and roughly doubled the number of processes (VMs) running and also distributed a series of additional simulation and analysis daemons. The first simulation of these experiments was completed on 20 March, 2011. On April 21, 2011 Experiment #7 was shut down to allow all resources to be allocated to Experiment #6. A number of server outages or failures of daemons occurred during this period, necessitating the constructing of a “watch dog” function which could restart daemons or regulate the rate of the production of simulations to process. Despite these issues the performance of the EvoGrid prototype during this period was superior, with some days averaging forty processed simulations or one processed simulation in substantially less than one hour.

Experiment	#Processed (#Failed) Simulations	#Incomplete Simulations	Complete: Incomplete Ratio	Highest Score
6-num mol	966 (134)	61415	1:50.422824302135	141
7-mol size	212 (45)	4391	1:16.759541984733	3

Table 10 Tabulation of EvoGrid Experiments #6 and #7

Table 10 depicts the total simulations processed by the time of the completion of the finalization of this thesis in early May, 2011. Experiment #7 was shut down as it was showing no clear trend. Experiment #6 was showing a clear trend as will be discussed next.

3.5.8 Experiment #6: Number of Molecules

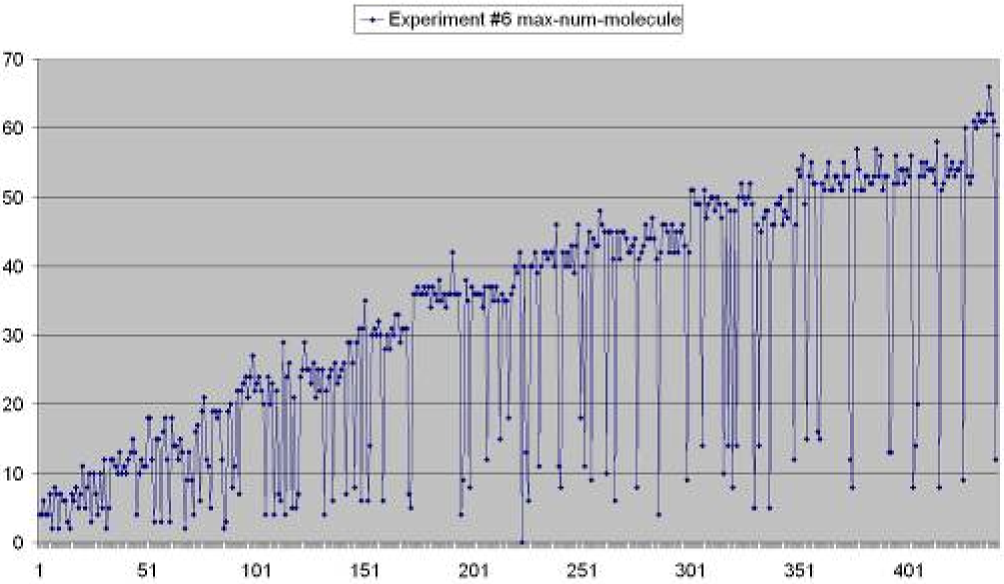


Figure 85 Experiment #6 number of molecules through the full processing period ending April 28, 2011

Figure 85 clearly shows a sustained trend in the growth of the number of molecules observed within frames for the data collection period ending April 11, 2011. The degradation factor is clearly not affecting the search function’s ability to find and sustain new maxima. At a number of points on this graph relative plateaux are achieved only to be exceeded later. A number of poor scores are obtained but these branches are quickly discarded. As can be seen in the chart, the previous record of a maxima of 60 molecules attained in Experiment #1 has been exceeded and the trend is continuing upward.

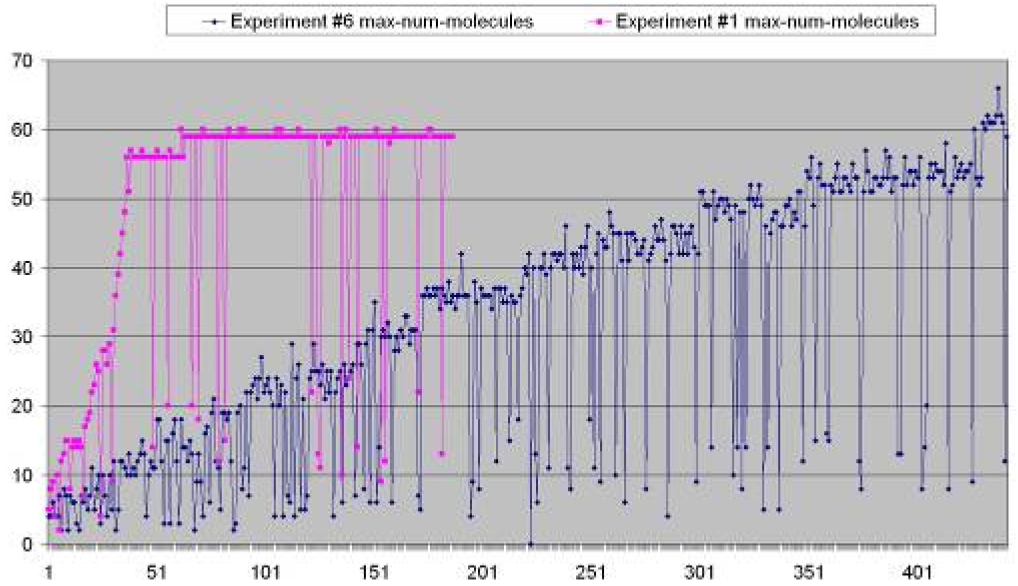


Figure 86 Experiments #1 and #6 compared

Figure 86 plots the results of Experiment #1 (in pink) and Experiment #6 (in blue) clearly shows a sustained trend in the growth of the number of molecules within frames. Experiment #1 was terminated at 187 processed simulations after two months of computation when it was felt that the maximum of 60 may represent a long term plateau. Clearly this experiment was “holding on” to the maximum of 60 and with no back-tracking degradation factor in place it was unable to climb down from that maximum.

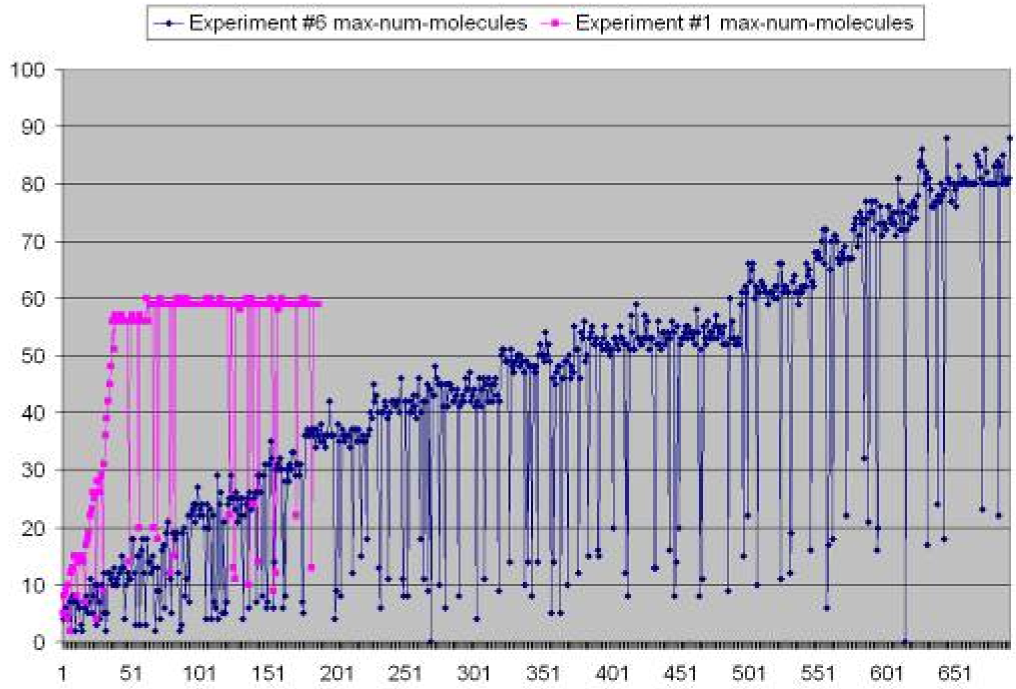


Figure 87 Experiments #1 and #6 compared following four days of additional processing

Figure 87 illustrates the continued rise of the number of molecules observed in Experiment #6 with an additional four days of processing time and plotted on April 15th. In this period two hundred frames were computed and a maximum score of 88 was observed. This increase in performance occurred when the queue of simulations for the terminated Experiment #7 was finally cleared and Experiment #6 was permitted to run on all the cores and virtual machines.

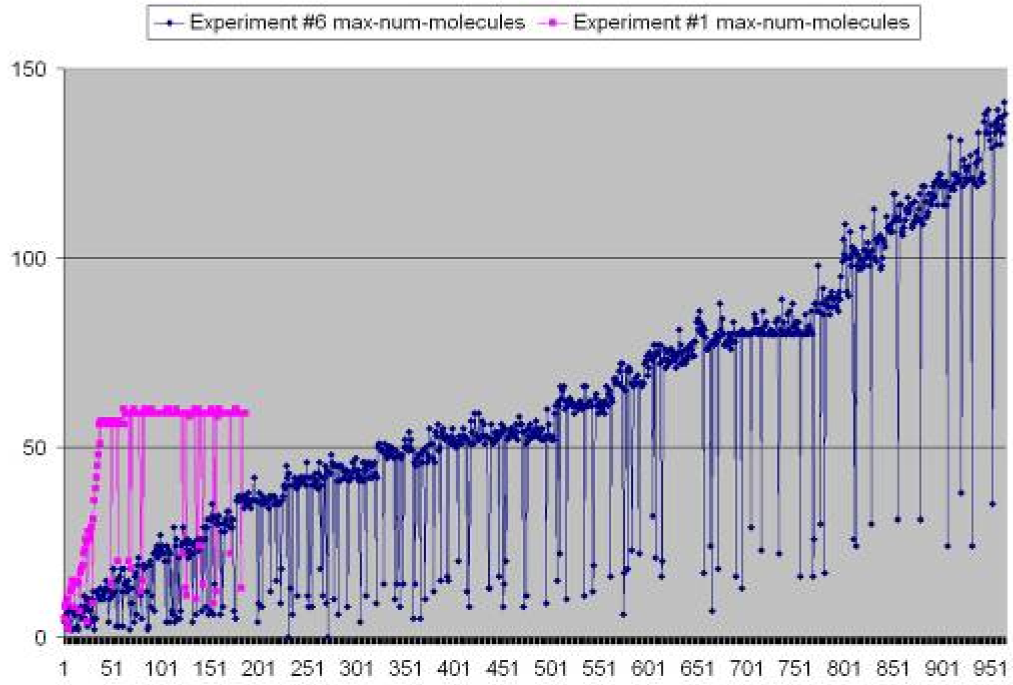


Figure 88 Experiments #1 and #6 compared following seven days of additional processing

Experiment #6 data was again collected on May 4, 2011. As Figure 88 shows that after processing 966 simulations Experiment #6 posted a maximum score of 141 molecules observed, with a size of three being the largest molecule. Note that the number of molecules, not size, was the target of this objective function. Figure 89 below presents a visual analysis of local maxima reached for Experiment #1 (a single maximum at 57-60 molecules, circled) and for Experiment #2 (a series of maxima reached up to the high of 141 molecules observed, all circled). The time to depart a given local maximum and find a path to a new higher maximum varies, and seems to generally increase, although with only one data sample, any characterization at this stage should be considered highly informal.

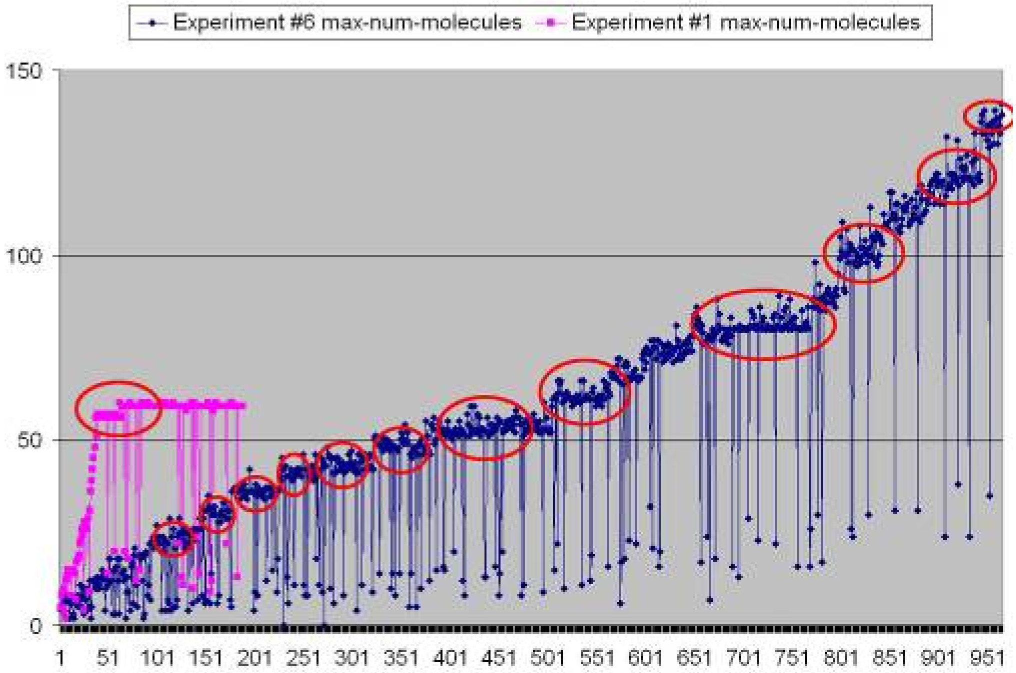


Figure 89 Analysis of discrete local maxima reached for Experiments #1 and #6

Figure 90 presents an interesting result following a further six weeks of processing, up to the point of termination of Experiment #6 in mid June of 2011: that of the reaching of what appears to be a long term maximum of 167 and then another at 189 molecules once past the 1655th simulation.

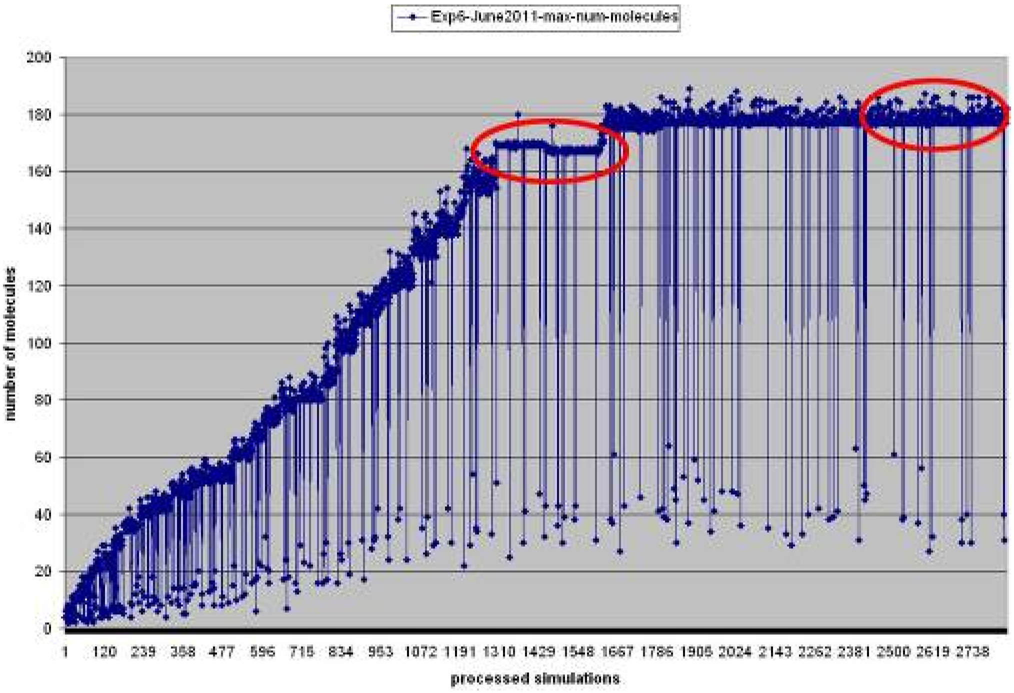


Figure 90 Experiment #6 with an additional six weeks of computation

3.5.9 Characterizing the Fitness Landscape

The question we may now be prepared to ask is: what are the properties of our underlying fitness landscape? In his book *At Home in the Universe* (Kauffman, 1995) Stuart Kauffman introduces random and correlated landscapes. For hill climbing to be possible Kauffman talks about the need for landscapes to be “still smooth enough to provide clues in ways uphill to distant peaks” (p. 154). Random landscapes are so randomly rugged that there are “no clues about uphill trends [and] the only way to find the highest pinnacle is to search the whole space” (p. 155). On the other hand, Kauffman describes other classes of landscapes which embody these clues and are conducive to adaptive searching and, as it turns out, climbing by evolution through Darwinian natural selection. Kauffman refers to these smoother, clue-filled landscapes as correlated, stating that in such landscapes “nearby points tend to have similar heights [and] the high points are easier to find” (p. 169). From our success in climbing up sequential maxima in Experiment #6, we are clearly dealing with a rugged, yet correlated landscape. In addition, our landscape is not static. As can be seen by the final results of Experiment #6, almost one fifth of the atoms available in each 1,000 atom volume have been captured within molecular bonds. Thus, as discussed by Rangarajan in Chapter 1, the *objective function* that underlies the landscape is itself changing. In the simplest terms the number and kind of interaction between atoms is changing through time, although we make no attempt here to characterize this function. Kauffman also observes that “climbing toward the peaks rapidly becomes harder the higher one climbs… [as] the higher one is, the harder it is to find a path that continues upward” (p. 168). It turns out that even in modestly rugged correlated landscapes the “waiting time or number of tries to find that way uphill increases by a constant fraction… [and it becomes] exponentially harder to find further directions uphill” (p. 178). From a comparison of experiments, in Figure 90 and we can see that the event of the reaching of a significant plateau (which might an regional but perhaps not the highest maximum) took over 1600 processed simulations for Experiment #6 versus around 40 for Experiment #1 (Figure 61). These results are suggestive of Kauffman's constant factor and that if an 8th experiment should be carried out it would take a much larger number of simulations to best the score and reach another regional maximum. The final consideration here is Kauffman's observation that in nature, increasing rates of mutation within a living species can overcome the natural tendency to find ways to higher fitness and cause the "population [to drift] down from the peak and [begin] to spread along ridges of nearly equal fitness" (p. 184). Our degradation factor employed in Experiments #4-#7 might be considered an analog to this mutation rate, in that as we can see from the results of Experiment #6 we have retreated gradually from a number of local maxima only to find ways to higher peaks. Kauffman points out that "if the mutation rate increases still further, the population drifts ever lower off the ridges into the lowlands of poor fitness" (p. 184). Such an "error catastrophe" may be evidenced in our experiments by the too-powerful effect of the degradation factor in Experiments #5 and #7. As a final point, therefore, these experiments, and the hill climbing fitness they implement, are *highly sensitive to the conditions set up within the search and scoring, even more so than the underlying properties of the fitness landscape itself*. Further analysis of these experiments, which is beyond the scope of this thesis but valuable for future work, might include:

1. Generation of a histogram of the occurrence of the types of molecules through time.
2. A mathematical treatment of the formulae and derived rate constants of creation and destruction of molecules for several of the experiments based on the Lotka–Volterra equations.
3. An examination of the simulation to extract the chemical reaction graph (molecules formed and broken through time) which could serve as a valuable input to the hypopopulated reaction graph experiment promoted by Kauffman (Chapter 1).

3.5.10 Overall Analysis and Key Results

The rapid initial rise and stair-casing of Experiment #1's performance indicates how the *priority_simple* function in the pseudocode in Table 4 operates and conserves its gains. In priority_simple there is no active mechanism to force the search back down off a maximum, so if no new, higher maxima are available within the pool of branches created from that point in the generation tree, no further progress is possible. The initial good fortune of the system in finding a pathway to early maxima may turn into bad luck as the vector becomes trapped on a long term global plateau. This corresponds to the case (C) in Figure 39 where without back-tracking no further exploration of opportune branches in the tree would occur.

The degradation factor in operation in *priority_generation* is clearly working its magic in Experiment #6. While maxima are found, the system does not tarry long within them. As a result the trend is to back off out of maxima but the system is still able to aggressively search for further maxima. Given sufficient computational resources, or luck, or both, the selection daemon is able to find pathways to new maxima, despite a large number of intermediate poorly scored simulations. Figure 87 clearly illustrates that the approach taken by Experiment #6 is far more computationally costly, reaching par (60 molecules) with the best results of Experiment #1 at the 428th successfully processed simulation versus the 39th successfully processed frame for Experiment #1. This is a computational cost factor of 11:1. If, however, the goal is the increased complexity, the total run time of six weeks on a very modest array of computers is not a high cost differential when Experiment #1 achieved its plateau in approximately half of that time on a computing grid of less than one half of the computing power. Exact measurement of the scaling of algorithm performance over the addition of computing resources was not attempted in this work but clearly would be an admirable goal for future development and testing.

In conclusion, we put forth that the chart in Figure 88 embodies the major result of this entire work, which we summarize as follows:

> *As a result of operating and analyzing these experiments for six months of continuous run time we have determined that this class of distributed small volume, molecular dynamics simulation is highly sensitive to small changes in the criteria used in the search function, to computing resources available, and to the presence or absence of the actions of selection and analysis daemons. Within this design underlying systems must be tuned carefully with the assignment of simulation and analysis processes to computing resources or large queues of unprocessed data may accumulate. Considering all of the above the optimization technique of stochastic hill climbing aided by search is able to produce sustained achievement of ever higher maxima with the ability to break through previous thresholds in a reasonable amount of time with limited computing resources.*

To conclude this look at Experiment #6 it is worth looking at the molecular sizes produced (Figure 91) which, while they did not match the performance of Experiment #2, did sustain a small growth in the population of larger molecules with size three.

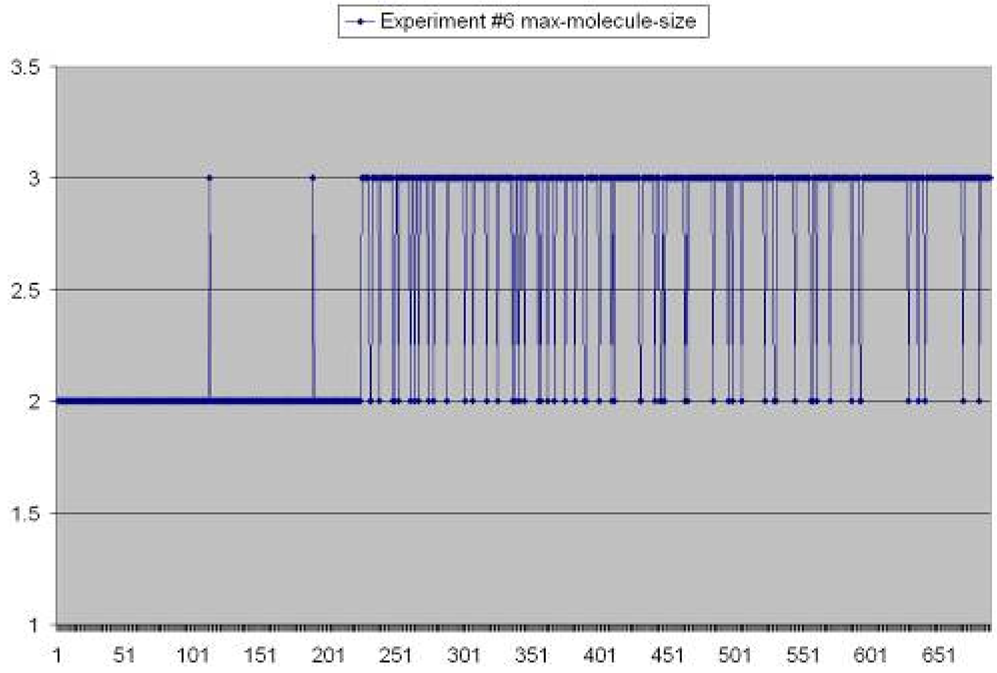
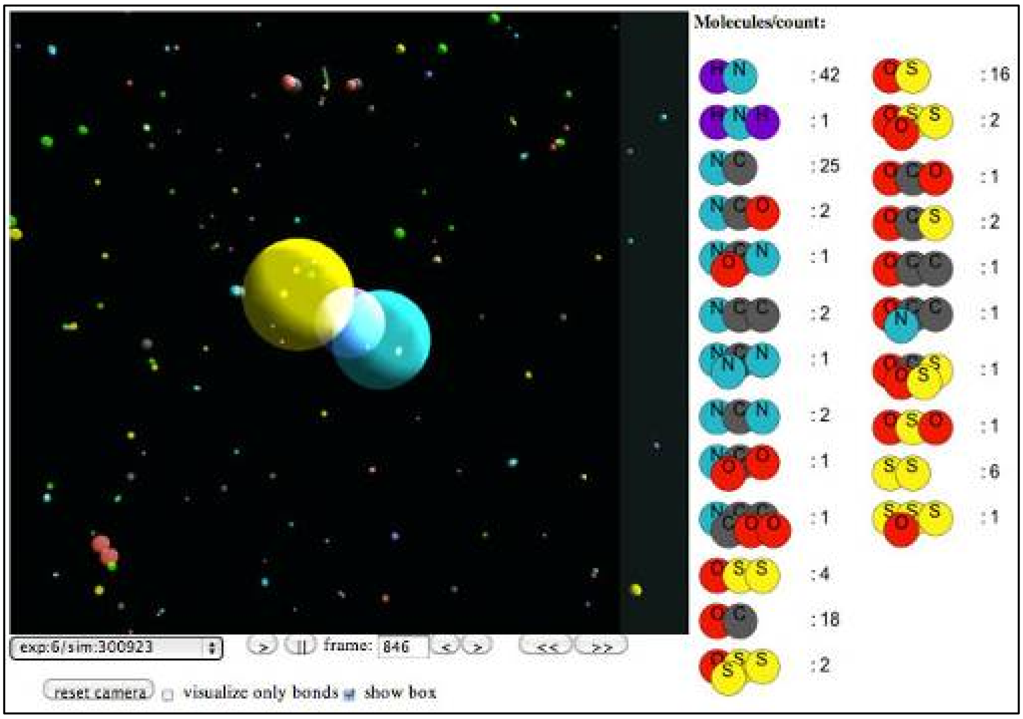


Figure 91 Experiment #6 molecular sizes observed

Figure 92 gives a look into the 3D representation of one later frame of Experiment #6 showing many larger molecules of sizes within a field of 102 molecules.



Molecules/count:

H N	:42
H N H	:1
N C	:25
N C O	:2
N C N O	:1
N C C	:2
N C N N	:1
N C N	:2
N C O O	:1
N C C C O O	:1
O S S	:4
O C	:18
O S S S	:2
O S	:16
O S S O	:2
O C O	:1
O C S	:2
O C C	:1
O C C N	:1
O C S O S	:1
O S O	:1
S S	:6
S S S O	:1

exp:6/sim:300923

frame: 846

□ visualize only bonds ☑ show box

Figure 92 Experiment #6 showing a later frame with a population of 134 molecules

In conclusion it seems that Experiment #6 had the properties of richness of both the number and size of molecular products and the capability, at least as of the time of this writing, of increasing that richness. The reader may notice what seem to be molecules of size four, five and six but it should be noted that in this experiment the molecular size is computed based on the number of bonds present in the longest graph through the molecular structure, which in the case for the larger molecules was still three (the specific geometry of these larger molecules needs to be better represented).

3.5.11 Experiment #7: Size of Molecules

In contrast to Experiment #6, the final experiment produced little in the way of the intended output. Figure 93 shows that while molecules of size two were present for each processed frame, only two occurrences of molecules of size three were observed.

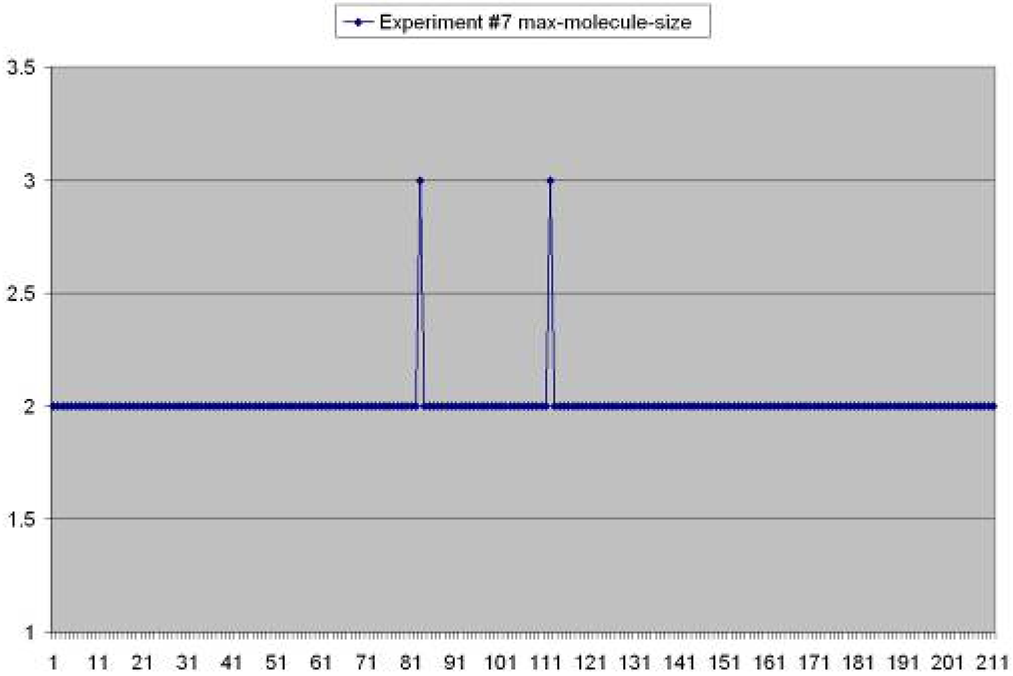


Figure 93 Experiment #7 chart showing the maximum molecular size observed in all processed simulations

The cause of the failure of the system to build on these maxima is not apparent. One hypothesis is that the degradation factor too rapidly caused the abandonment of the branches containing the high scoring frames. Another hypothesis is that in both cases of the maxima of three, a subsequent high scoring simulation crashed and as a result generated no future branches.

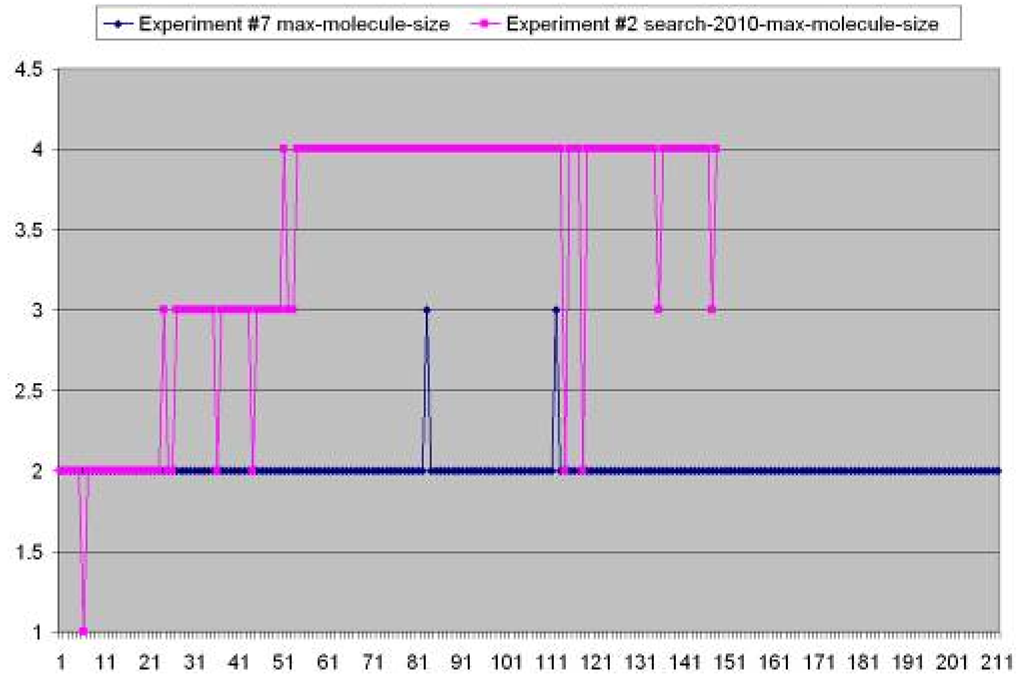


Figure 94 Comparison of Experiment #7 with Experiment #2

Figure 94 compares the results of Experiment #2 which attained maxima of molecule size four with the results of Experiment #7, which in the time available struck a plateau at molecule size two.

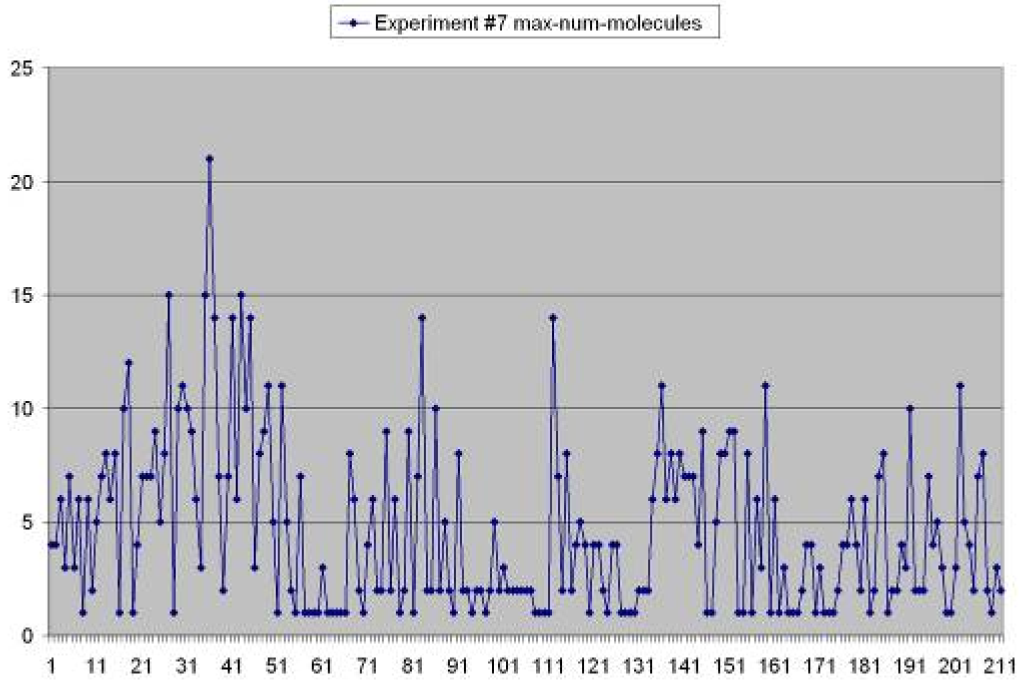
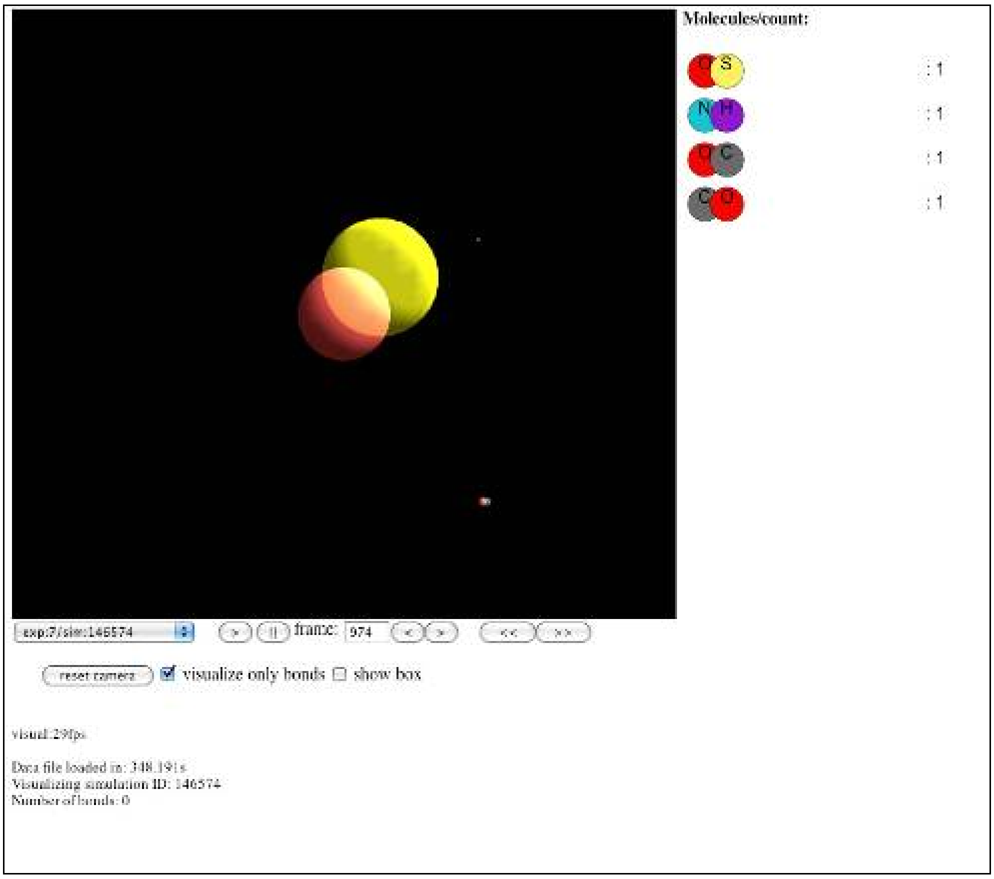


Figure 95 Experiment #7 number of molecules plotted against the simulation time

Figure 95 plots the number of molecules observed in Experiment #7 with no clear trend present and a poverty of molecules being produced at many points. Similar to the chart in Figure 82 for Experiment #5, it could be that if the system finds itself trapped in local minima of the production of numbers of molecules, it would also thereby have great difficulty producing molecules of larger size.

Figure 96 shows a 3D view of an average frame in Experiment #7 with four molecules of size two each.



Molecules/count:

O S	: 1
N H	: 1
O C	: 1
C O	: 1

exp:7/sim:146574

frame: 974

☑ visualize only bonds □ show box

visual:29fps

Data file loaded in: 348.191s
Visualizing simulation ID: 146574
Number of bonds: 0

Figure 96 An average frame in Experiment #7

Figure 97 shows the generation hierarchies for Experiments #6 and #7 rotated on their sides and shown slightly cropped with the first processed simulation on the left side.

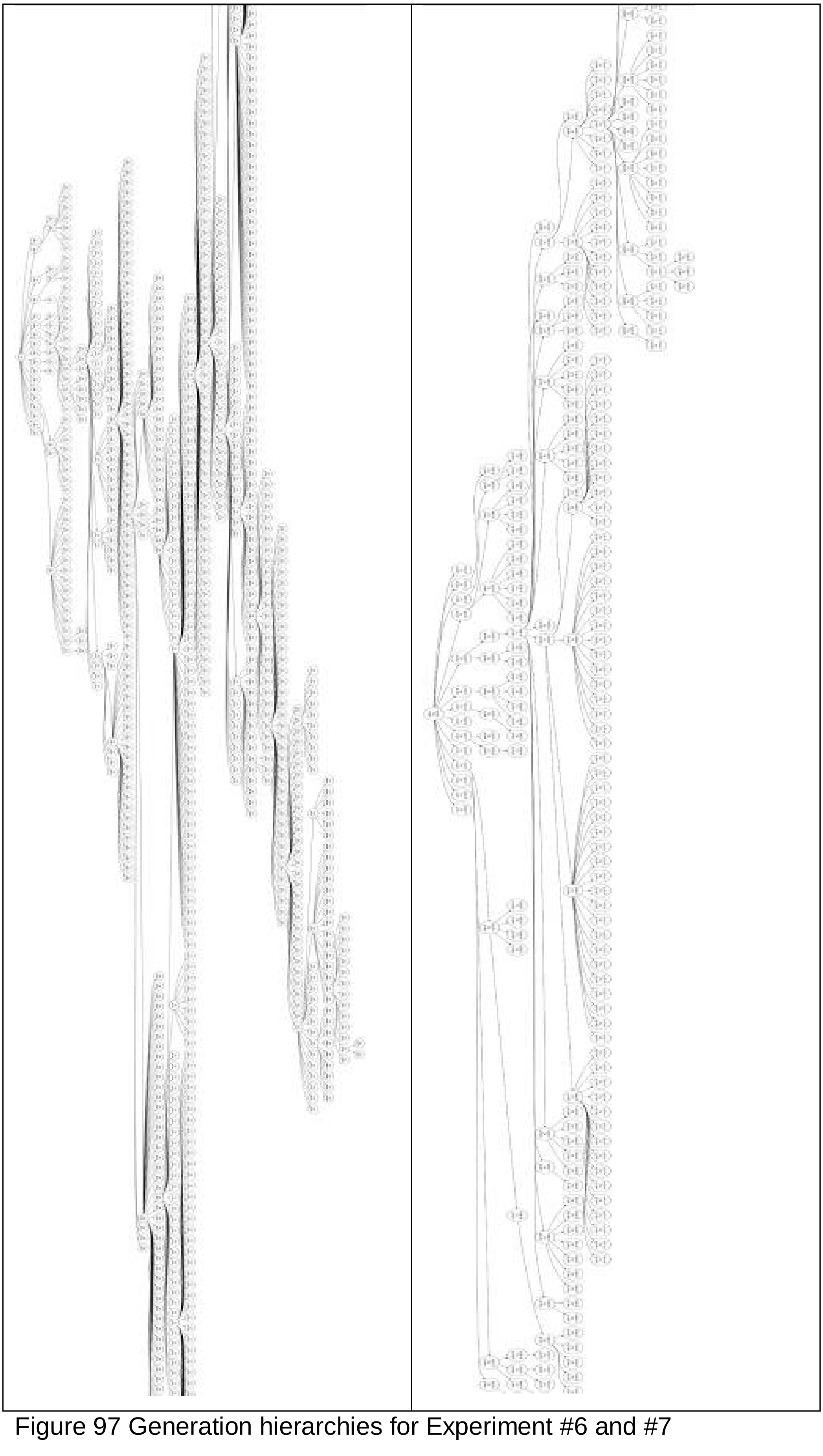


Figure 97 Generation hierarchies for Experiment #6 and #7

When comparing these with the hierarchy trees for Experiments #4 and #5 (Figure 84) what stands out is how much more depth and complexity is present in these two trees. While Experiments #4 and #5 were bogged down in processing a large number of simulations at one or two levels, the active selection daemon permitted pruning of these broad branches and exploration into further realms. While both of these generation trees look similar, appearances can be deceiving as the tree for Experiment #7 does not represent a success while the tree for #6 does, at least by our measure.

3.5.12 Summary of Experiments

Experiment	**#Processed (#Failed) Simulations**	**#Incomplete Simulations**	**Complete: Incomplete Ratio**	**Highest Score**
1- num mol	187 (26)	11015	1:51.713615023474	60
2- mol size	148 (39)	40654	1:217.40106951872	4
3 -control	155 (21)	16178	1:91.920454545455	N/A
4-num mol	158 (39)	5884	1:29.868020304569	33
5-mol size	71 (42)	38820	1:343.53982300885	2
6-num mol	966 (134)	61415	1:50.422824302135	141
7-mol size	212 (45)	4391	1:16.759541984733	3

Table 11 Summary of all experiments as of May 4, 2011

Table 11 summarizes all seven experiments undertaken as of May 4, 2011. It is useful to compare and contrast each. Experiments #1 and #2 used a straightforward form of stochastic hill climbing with the *priority_simple* function and produced a reliable upward trend in observed complexity (number of molecules and size of molecules). Experiment #3 ran with the same starting frame contents and conditions as #1 and #2 but with no search function enabling hill climbing. The selection daemon still selected frames at random for processing but with no regard to their score or priority. Experiment #3 produced a slight upward trend in the number of molecules (with no sustained level) and no trend in the size of molecule. Experiment #3 acted as a control on all six other experiments employing the search function. Experiments #4 and #5 implemented a new function: *priority_generation*, which used a degradation factor which it was hoped would back the system out of local maxima enabling it to find higher maxima. In addition, the selection daemon was turned off to force these experiments to process every branched simulation. Experiments #4 and #5 did not result in any sustained, irreversible growth in complexity, although there was a suggestive trend for higher maxima for the number of molecules in Experiment #4. Finally, Experiments #6 and #7 saw the return of the selection daemon in combination with the degradation function. Experiment #6 expressed the best performance of our hypothesized optimization of any experiment, with consistent, irreversible growth in numbers of molecules as well as a parallel modest growth in molecular size. Experiment #7 produced poor performance in that molecular size remained flat at size two while the population of molecules fluctuated and then declined.

3.6 Summary Analysis and Conclusions

The main result reported in Chapter 3 is the successful implementation, deployment and testing through multiple experiments of the EvoGrid prototype. The prototype was operated in two generations over a period of several months on a number of hardware and software configurations. The prototype implemented the following system to test the hypothesis stated in Chapter 1: *a distributed processing and global optimization computing grid employing search coupled with stochastic hill climbing to support the emergence of phenomena (the formation of molecular bonds) within small volume, short time frame molecular dynamics simulations*. The analysis of seven experiments carried out in the Prototype2010 implementation provided the following positive result for the hypothesis:

> *Distributed processing and global optimization employing search coupled with stochastic hill climbing can produce*

significant performance improvements in the generation of emergent phenomena within small volume, short time frame molecular dynamics simulations over non-optimized solutions.

Given our experience of these seven experiments we must now add to this general hypothesis the caveat that such methods can produce significant performance improvements *given the correct tuning of the scoring, search and processing algorithms.* This conclusion emerged from the failure of several experiments to make progress greater than that shown by the control experiment. This conclusion deserves further clarification through additional analysis of the data.

3.6.1 Comparison of Key Experiments

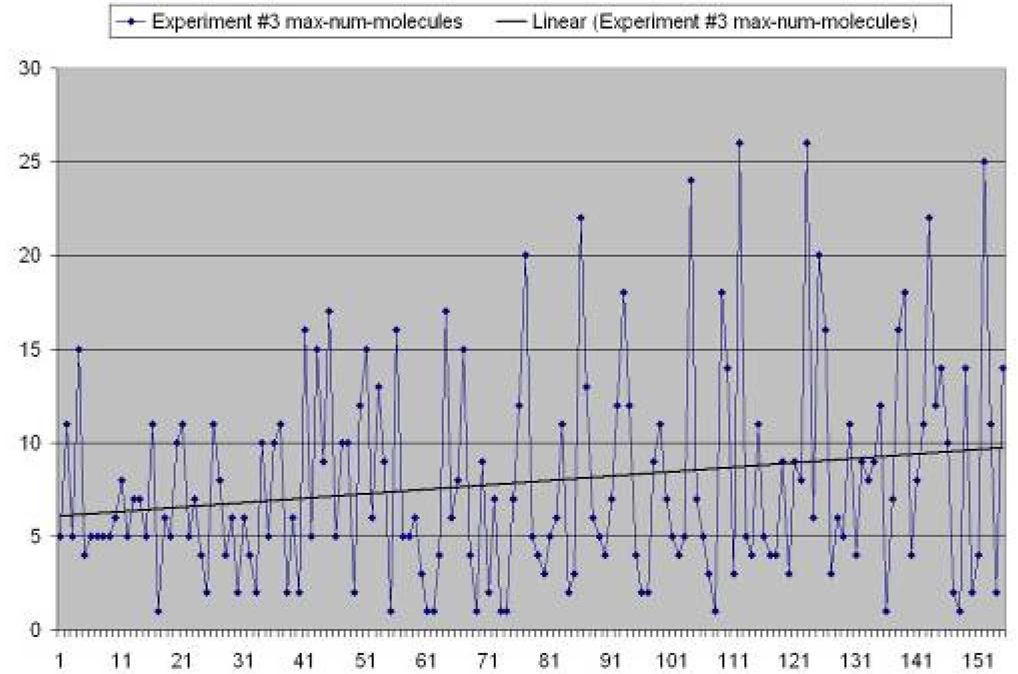


Figure 98 Simple linear trendline through the scores of Experiment #3

Let us now compare the performance of the control, Experiment #3 (no directed search) with a “successful” experiment and an “unsuccessful” experiment both of which employed directed search. Figure 98 depicts a simple linear trendline (averaging the Y-axis values) through the number of molecules score data for Experiment #3. We opted to use the score for number of molecules as it has large enough values to plot trendlines. Molecular size data is too coarse-grained in its whole numbers of 1, 2, 3, or 4. As stated previously, it was expected that this score (complexity) would grow through time even though no directed search optimizations were performed. We can see from Figure 98 that an approximate average of 6 molecules observed in the first frame grew to an approximate average of 10 molecules by the 155th frame processed.

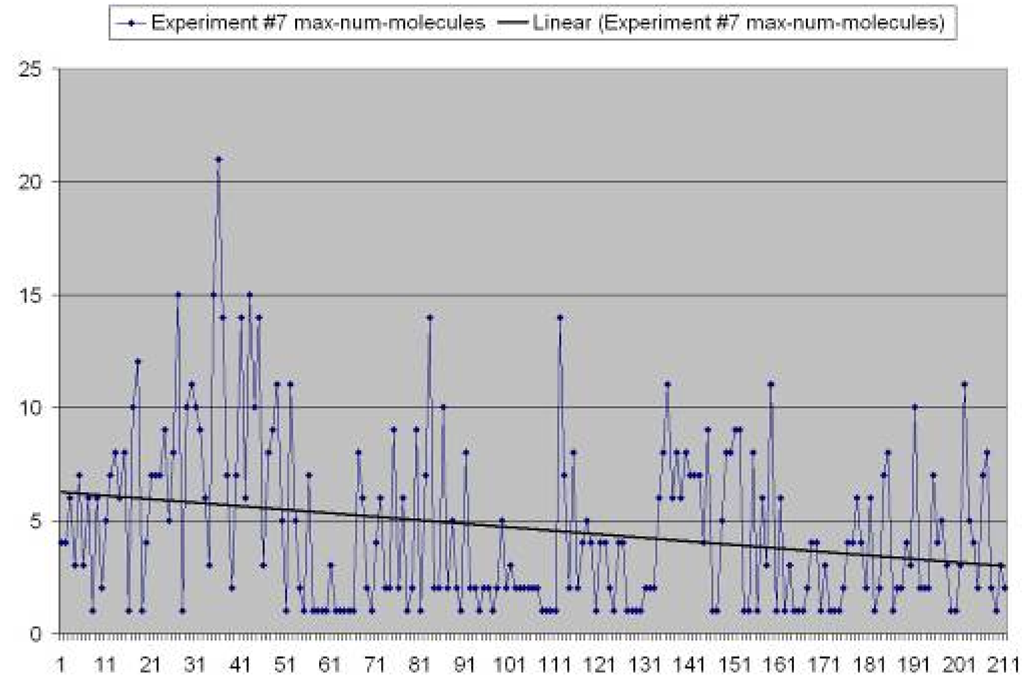


Figure 99 Linear trendline through the number of molecules scores in Experiment #7

We now present Experiments #6 and #7 as a comparison. It should be recalled that Experiment #6 employed a search function looking for an increasing number of molecules while #7's search function was searching for increased molecular size. However, as we have already seen with Experiment #2 above, searching for molecular size may actually have the side effect of producing more molecules. Therefore it is worth comparing the number of molecules produced by all three of these experiments. Both Experiments #6 and #7 were run on the same hardware configuration. For our "unsuccessful" Experiment #7, Figure 99 shows a decline in the average number of molecules from six to approximately three. The cause of this failure to grow the complexity of the simulation is likely to be the sensitivity of the search function (for molecular size) to the degradation factor. The rapid degradation of the coarser grained numerical scoring of molecular sizes of 2, or 3 prevented the conservation of the gains of initial local maxima or the gaining of new maxima.

Figure 100 illustrates an opposite effect: the consistent growth of the linear trendline from an approximate number of molecules of six per frame to an observed maximum population of 141 molecules and the trendline average of 110 by the final frame, well more than *a full order of magnitude greater than the control.*

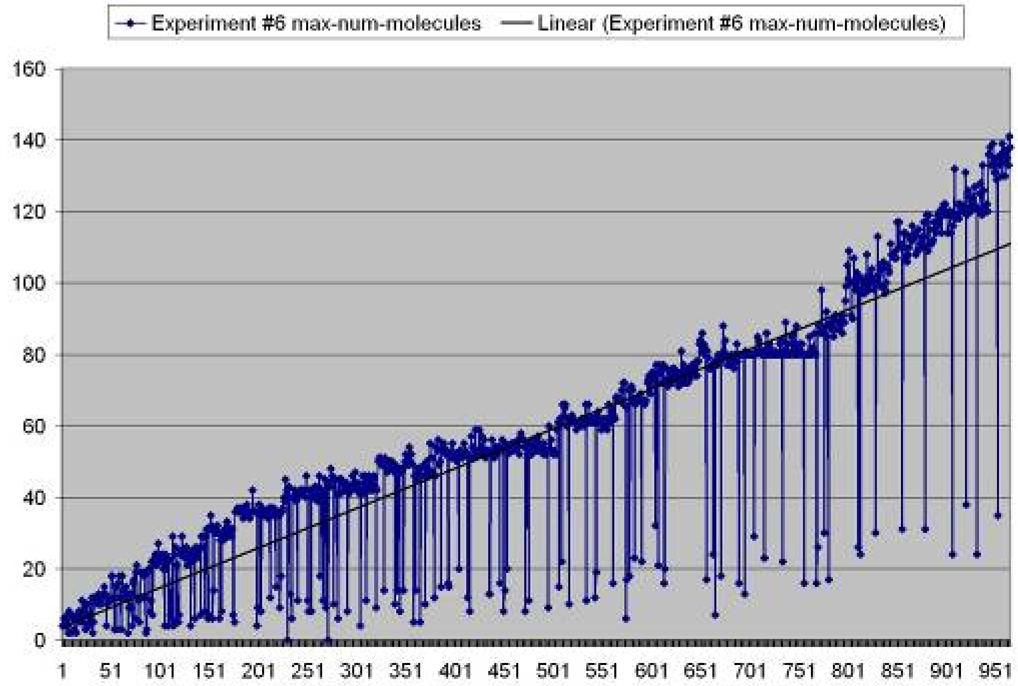


Figure 100 Linear trendline through the scores from Experiment #6

Computer simulation experiments, as with bench chemical experiments, are sensitive to their initial conditions. In the circumstance where a series of simulations are tested, initial conditions altered and then these simulations are run again, *this sensitivity may be vastly magnified*. It is *this magnification of sensitivity* which we are putting forth as a key benefit to science in the simulation of small volumes of molecular dynamics in which emergent phenomena are sought.

This concludes our treatment of the *intrinsic value* of the optimization methods employed in the EvoGrid prototype. We will devote the next and final chapter to the potential *extrinsic value* of these methods to science, and conclude by considering possible impacts of cyberbiogenesis endeavours on science, and society.

Chapter 4: Limitations, Roadmap, Open Questions and Broader Considerations for Endeavours Seeking to Compute Life's Origins

Introduction

As was discussed in the introduction to this thesis the long term goal of the new field of cyberbiogenesis is to produce *in vitro* experimental verification of an *in silico* simulated pathway from nonliving to living molecular assemblages. Chapter 1 presented the hypothesis that a computer software prototype, the EvoGrid, could be designed and built to verify whether or not *distributed computing with search and hill-climbing optimization of simple emergent phenomena within small molecular dynamics volumes is able to produce desirable outcomes (ideally open ended growth in complexity) versus non-optimized solutions*.

While delving into the prior art of the cognate fields depicted in Figure 32 we were able in Chapter 2 to propose the design and the implementation of a system in which an off-the-shelf molecular dynamics component would be brought into a distributed computing framework which would then be able to test our hypothesis. Chapter 3 chronicled our construction of two versions of the EvoGrid prototype and their subsequent testing through seven experiments which included one control. Although we implemented a naïve form of chemistry in very small volumes over short time scales, our results suggest that further development along these lines could be valuable to the science of the origins of life and other fields such as complexity systems science. We based this conclusion on the observation that several of our experiments produced significant computational cost savings while bringing desired emergent properties into view in shorter time frames. A key result was that though slight modification of one of our later experiments, we were able to break through a threshold of complexity that had become a plateau for an earlier one. This result suggests that our method may aid simulation environments to achieve the capability of *open-ended growth of complexity*. This open-ended facility is an essential property of any experiment (computational or chemical) involving research into the origins of life, or alternatively, into new theories of universal properties of complexification as proposed by Kauffman.

As was also reported in Chapter 3, the experiments carried out showed that, properly tuned, the methods employed *produced significant performance gains with the added benefit of creating a system capable of scalable computation and flexible search*. The same search facility used to score frames for overall complexity can be employed to identify specific molecular structures and chemical reactions. The distribution of frames onto a computing grid to be simulated and then analyzed by search functions can be used to build a scalable architecture to tackle larger problems in chemistry and origins of life.

We therefore have confidence that *the approach demonstrated by the prototype EvoGrid is capable of supporting the larger mission of computational origins of life endeavours*. In this chapter we will pursue a road map for future EvoGrid implementations which could be developed and deployed to tackle a number of important origins of life experiments.

The enumeration of the limitations of the current EvoGrid prototype is a good logical starting point to propose a *road map* for future development. We believe that this road map will specify the features necessary to build the current prototype into a more fully developed system capable of supporting key objectives in the science of the experimental simulation of life’s origins. A next objective of this chapter is to put forth a number of chemical experimental models that lie within the range of tractable computability for a system like the EvoGrid. We will also look ahead and suggest a series of larger experiments that could be of value to the field but that may lie outside the range of near term computability. Next we will briefly consider some of the broader scientific, philosophical, and societal considerations surrounding cyberbiogenesis and computational origins of life endeavours. Lastly we will conclude with a summary of the contributions to knowledge made by this work and some final thoughts.

4.1 Limitations of the EvoGrid Prototype

The most apparent limitation in the EvoGrid's first prototype implementations come through our employing of a naïve bond formation methodology. In the prototype, bonds are formed by simple proximity calculations using the positions, velocities and other data for atom objects exported from GROMACS. The GROMACS package is excellent for providing fast molecular dynamics simulations on individual computers but it was designed to handle existing molecular structures, finding their lowest energy states. This explains GROMACS' popularity in protein folding and other similar molecular geometry simulations. However, GROMACS was not optimized for bond formation while the simulation is being run. As a result of our bonding methods excess heat would be introduced into the simulation for which we had to compensate. This heat arose from the placing of atoms into bonding distances that were too close. This placement imparted potential energy into the atoms resulting in introduced motion which translated into heat. This naïve bond formation forced the heat during each simulation step to climb requiring us to employ an artificial thermostat to dampen heat back to the 300 Kelvin range and prevent a runaway effect which would crash the GROMACS engine. The goal of this thesis work was not to implement an accurate quantum mechanical approximation of bond formation but to test a theory of optimization within distributed computing as applied to molecular dynamics simulations. We feel, therefore, that our naïve bond formation with simulation adjustments produced a sufficiently viable step in the process to permit the testing of the architecture and its optimizations.

There are extensions to GROMACS including external packages that could be employed to implement a full complement of quantum calculations enabling bond formation. One such package is the Open MOPAC2009 library (Stewart, 1990, Stewart, 2009) which could be employed directly by GROMACS for covalent bond formation and the representation of other molecular affinities such as those produced by electrostatic and van der Waals forces. This implementation was deemed to be a major effort in terms of software development and well beyond the scope of this work. In addition, the added overhead of libraries like MOPAC2009 would strain the commodity computing clusters we had access to. GROMACS was selected due to its open source availability, speed on commodity hardware, stability with repeated restarts (other packages experienced crashes) and active community of support. Despite other limitations, GROMACS worked well for generating the experiments described in this thesis.

A further limitation arose from our branching operations which were meant to simulate the flow of material through the environment and natural forces causing bond breakages which were also naïve at best. While our universe is closer to a simulation of real chemistry (and nature) than many comparable AChem simulation environments (reference section 2.4), it still has a long way to go.

A final major limitation from the current prototype that would have to be addressed for future work emerged from the limitation of the performance of our available computing hardware. As was covered in Chapter 1, computing on sufficient spatial scale and time frames to create simulations of biologically interesting phenomena was beyond the scope of this project and only now just becoming possible. As computing hardware and software scale up in performance over the next several decades this will enable future EvoGrids to engage in biologically relevant simulations.

4.1.1 An Implementation Road Map for Future EvoGrids

Choice of MD simulation engine, bonding methodologies and hardware platforms aside, there are a number of low level improvements for the platform itself surrounding simulated physics and environments which could be implemented:

- Models permitting fluctuations and periodicity in the environment creating non-equilibrium conditions including thermal gradients and other spatiotemporal environmental heterogeneities.
- Modeling of dissipative systems (sinks) which will continually use up energy could be coded as thermal inversions or vortices creating heterogenic conditions.
- Implementation of fluxes of material through volumes might be encoded through the distribution of molecular inventories and support of reactions such as autocatalytic cycles.
- Simulation of plausible liquid, gas and solid phases of matter could be implemented to support the modeling of small contained spaces and phase transitions.
- As mentioned in section 2.2, the global optimization method could be extended to employ a hybrid implementation combining stochastic hill climbing with other methods including gradient search or simulated annealing.
- The computing resources of Calit2 at the University of California at San Diego and elsewhere could be scaled up to hundreds or thousands of computers to carry out more ambitious experiments.
- Simulation support for graphics processing units (GPUs) as well as dedicated supercomputing hardware could benefit future EvoGrid implementations. In particular, support for the Compute Unified Device Architecture (CUDA) defined by the Nvidia company could provide benefits to the platform (Nvidia, 2011). Alternatively OpenCL, which is independent of hardware manufacturer, could be implemented.

4.1.2 The Generalisability of the Optimizations

The gains in efficiencies produced by the optimization methods employed here are potentially very impressive; however, the test cases are so simple as to only be a suggestion of possible gains made by more sophisticated scenarios. The detection of the formation of macromolecular structures such as lipid vesicles, the observation of autocatalytic sets in operation or the witnessing of replicating informational molecules are laudable goals for a future EvoGrid but well beyond the scope of this preliminary work with the available technology. For complex biomolecular simulation spaces the number of interactions and time frames involved likely outstrip all current computing grid technologies. In addition, the definition of search functions to help to optimize systems toward bio-realistic behaviors is also an unsolved challenge. Indeed, it is likely that in order to reach the goal of the simulated formation of a single closed lipid vesicle, possibly dozens of intermediate stages would have to be set up and experiments run. It is possible, however, that the savings made by the optimizations demonstrated by our prototype may make significant contributions to these more challenging experiments.

In keeping with the philosophy that the EvoGrid is a system that can be used to produce simulations that can be validated in actual chemistry, each of these stages would have to be laboratory tested. At any point the laboratory analogue to the EvoGrid simulation may diverge from the simulation results and while the computer simulation may have produced a sterling result, it might not be of productive use to chemists. Refinement and iteration of the virtual experiments would hopefully bring them more in line with the chemistry. It is conceivable that, along the lines of the revolution of gene sequencing (Venter et al., 2001), the direct coupling of a future EvoGrid with automated chemical experiment and measurement apparatus could accelerate these iterations, bringing a qualified *high fidelity* emergent chemical simulation capability to reality.

Needless to say, such a capability would be of significant scientific and commercial value, possibly being generalized away from the origin of life roots of the effort to become a “ChemoGrid” used by medical science, materials science, microelectronics and other fields. Ironically it is the invention of microfluidics, derived from chip-making techniques, which is bringing to the chemistry laboratory the promise of automation and miniaturization. Flasks, test tubes and Bunsen burners may one day be replaced by supercomputing clusters coupled to super chemical computing microfluidics arrays.

By employing large numbers of small cameo simulations, the EvoGrid can identify initial conditions and simulation methods to enable much larger computer or chemical simulations combining several experiments (Figure 42). In such a stepping-stone manner, we believe that the simulation of entire protocells including their emergence through intermediate steps may be accomplished. The simulation of a structure as large as a protocell would require several different simulation engines operating in parallel at multiple levels of scale. One module might perform a coarse grained simulation of the membrane encapsulating the protocell while another would employ MD techniques to simulate the movement of molecules across that membrane. The EvoGrid has been designed to meet this challenge. The framework was constructed in an open manner to be able to incorporate many different modules all running using the same optimization techniques on a single distributed computation network.

4.1.3 The Power and Necessity of an Open System Employing Search in de Novo Emergent Environments

Search is a vitally important benefit provided by the EvoGrid architecture. Flexible and powerful search capabilities are critical in any environment where *de novo* emergence of structures and behaviors is a desired outcome of the simulation. If millions or billions of small simulations are to be run, search through these simulations must be flexible, powerful and automated. The EvoGrid prototype implements an open system to permit export of simulation frame states to a JSON-based file containing generically described objects and properties. The Simulation Manager then processes these files, applying search and restarting of simulations. In principle any environment could parse this file and apply its heuristics. As new AChem engines are supported in future EvoGrids, this frame state file would evolve but remain an open, published standard. Therefore, in such a loosely coupled system any search optimization technique can be applied independent of the AChem engine used. The key thing here is that the incorporation of search within the EvoGrid transforms it from a *simulation system into a discovery system*. The potential value of such a system to real origin of life experiments is covered in the next section.

4.2 Candidate Case Study Experiments for Future EvoGrids

Guidance sought from external advisors throughout this effort produced a continuum of recommended *in vitro* experiments which could benefit from the discovery of predictive properties or experimental pathways using *in silico* simulation. This will be described here with only a cursory consideration of all the aspects of what it would take to create high fidelity chemical models in software. These are listed as possible destination points along a road map.

When future EvoGrids can simulate both large enough volumes with sufficient molecular content over biologically significant time scales contributions to laboratory origins of life experiments could be made in several areas. As a part of this road map we researched and are able to present here a continuum of *in vitro* experiments that could be supported in some meaningful way by computational simulation. In each case a researcher suggested the experiments and expressed a belief that computational treatments could be helpful as a predictive tool to design experiments and test options. The experiments are presented in order of difficulty in the sense of computability. The final experiment is a highly speculative multi-stage experiment being developed by the author and colleagues which could illustrate one pathway to the origin of complex containment expressing lifelike properties. The experiments to be covered include:

4.2.1 Near term viability: formation of simple molecules and small self-organizing clusters of molecules

- Experiment #1: Interstellar Chemistry Model
- Experiment #2: FLiNT Nanocell Model

4.2.2 Medium term viability: formation of catalysts, informational molecules and autocatalytic sets

- Experiment #3: Riboyzyme Selection Experiment Model
- Experiment #4: Model of a Hypopopulated Reaction Graph for the Study of Autocatalytic Sets and the Adjacent Possible
- Experiment #5: Model for RNA-Making Reactors in Hydrothermal Vents

4.2.3 Long term viability: supramolecular structures

- Experiment #6: Model for Encapsulation of Polymers in Multilamellar Structures through Wet/Dry Cycles
- Experiment #7: Model of the FLiNT Protocell Life Cycle

4.2.4 Very long term viability: multi-stage supramolecular model for an end-to-end origin of life

- Experiment #8: CREATR Model

Each experiment is given a measure of complexity which is a rough estimate of the number of atoms which would need to be simulated and an equally rough estimate of the duration of real-time chemistry which would have to be simulated (from nanoseconds to seconds). The next category assigned each experiment is a measure of when each simulation might become possible (in years in the future):

- Near term: immediately possible with current technology, 1-2 years to implement simulation framework.
- Medium term: simulation is possible in approximately five years.
- Long term: simulation is possible in no less than ten years, likely probable in fifteen.
- Very long term: simulation not possible before twenty years and could be implemented within thirty years.

4.2.1 Near term viability: formation of simple molecules and small self-organizing clusters of molecules

Experiment #1: Interstellar Chemistry Model

Experiment	**Complexity**	**Tractability**
Interstellar Chemistry	1K-10K atoms 100 nanoseconds	Near term

As was concluded in Chapter 1 the design of the first EvoGrid most closely corresponds to a model of a diffuse mixture of elements in the gas phase similar to the interstellar vacuum. It was therefore suggested by Deamer that a next step for the EvoGrid would be toward modeling the interstellar medium more completely (Deamer and Damer, 2010b). Deamer has described this medium as containing "simple compounds like water and carbon dioxide; dust particles composed of silica… and metals like iron and nickel" (Deamer, 2011, p. 18). Upon examining the results of the EvoGrid prototype experiments, Deamer noted that a number of molecules formed are suggestive of compounds founds in the interstellar medium, sometimes called *cosmochemistry*. It was therefore suggested that a next step would be to formulate better bond formation and test the resulting products against known assays of cosmochemical surveys. The recent Stardust spacecraft collected samples from comet Wild 2 (Sandford et al., 2006) and gave tantalizing evidence of biologically relevant interstellar molecules.

From a discussion (Deamer and Damer, 2011a) any simulation of this environment would need to run at very low temperatures, close to absolute zero (a few degrees Kelvin). An activating energy source to drive bond formation might well be light from a nearby star. It is believed that bonds are formed in this environment not by collisions but directly by incoming energy or by gradual heating and melting of grains. Long polymers and lipid-like molecules can form in these environments as evidenced by the collection of meteorite samples and as suggested by experiments carried out within terrestrial analogs in vacuum chambers (Deamer and Pashley, 1989).

Simulation Approach

This model is the closest to the current form of the EvoGrid prototype. Implementation of realistic chemical bonds caused by direct activation energies as well as heating and cooling cycles might bring the simulation closer to corresponding to chemistry in the interstellar medium. Very small volume simulations could be carried out to see what products and yields occur and compare these molecular species with those observed by direct astronomical spectral observation, samples returned by spacecraft or the products of terrestrial vacuum chamber experiments.

Experiment #2: FLiNT Nanocell Model

Experiment	**Complexity**	**Tractability**
Fellermann: FLiNT Nanocell Model	100K atoms, Microseconds to minutes	Near term especially using coarse graining techniques in replacement of MD or MPI techniques

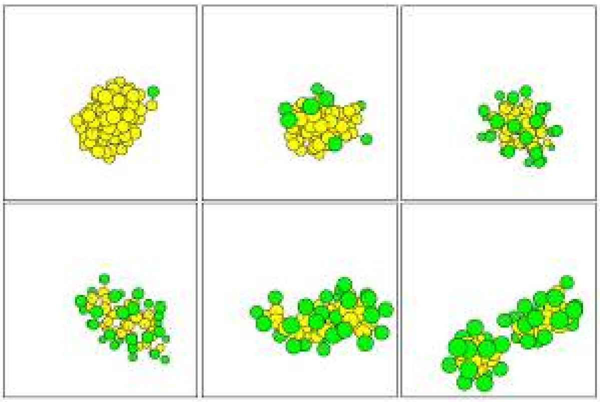


Figure 101 From Fellermann: Metabolism and fission of a nanocell with surfactant (green) surrounding lipid (oil in yellow) and the water is not shown.

The FLiNT nanocell model describes a minimal self-replicating system involving surfactant (soap) molecules associating with a droplet of lipid (oil) which they start to break down, converting into more surfactant (see Figure 101 from (Fellermann, 2009a, p. 67)). This surfactant/lipid structure self-organizes into an analogue of a tiny cell which has a simple yet dynamic surface and interior with which it exchanges. This is one of the simplest systems which exhibits these properties and the system has been modelled *in silico* using DPD methods (see description of DPD in section 1.2.2) and experimentally studied *in vitro* the laboratory (Fellermann et al., 2007). This system has a tractably small number of molecules involved and as such is a good candidate for simulation at the atomic level in MD. As illustrated in Figure 101 the initial surfactant metabolizes a precursor droplet and turns it into a functioning nanocell (top panels 1-3). This nanocell continues to consume its droplet and other lipid in the environment and driven by the changing of the lipid precursor to surfactant ratio (lower left panels 4-5) begins to elongate. These elongated structures are stable for a short time until all precursors are converted into surfactants at which point the nanocell divides, becoming two daughter cells (last panel).

Fellermann provided a description of the complexity of this system such that we can determine a rough estimate of its computational tractability. Recall that in DPD, atoms and molecules are grouped together in a coarse graining fashion into “beads” for greater ease of simulation. Fellermann begins by describing the basic atomic population of the nanocell itself (Fellermann, 2011):

> …one DPD solvent bead accounts for about 5-6 water molecules. Hence… [we have an experimental size] of 10^3 [molecules] and bead density of 3, include [five or six water molecules]*3*10^3 = 15,000 to 18,000 water molecules or 45,000 to 54,000 atoms… some of the systems have size 12 and 5^3, which corresponds to about 30,000 to 35,000 water molecules or about 100,000 atoms… should not be too much for GROMACS run on a cluster.
>
> However, if you really consider to model these systems in MD, be aware that the time scale of MD is an expected three orders of magnitude slower than DPD (Groot and Warren, 1997). While people have successfully modeled self-assembly of micelles the simulation times of metabolically induced micellar fission might be way out of reach for MD.

Fellermann then goes on to estimate the time scales that would be required to simulate this model:

> From an experimental point of view, the closest time estimates I could relate my simulations to are the fast and slow relaxation dynamics of micelles -- i.e. the typical times for micelles to exchange monomers with the bulk phase (fast relaxation time) or to assemble and disintegrate from solution (slow relaxation time scale). The fast time scale is on the order of microseconds, whereas the slow relaxation time is on the order of microseconds to minutes.

Simulation Approach

The conclusion is that this simulation is within range of MD engines in the class of GROMACS but that DPD and coarse graining techniques are likely to bring this class of simulation into nearer tractability. Alternatively using a multi processor interface (MPI) within the EvoGrid would allow four cores of a quad core processor to each run one quarter of a simulation space. Solving the boundary communication issues within the CPU array may still be somewhat costly but effective simulations of about 100,000 atoms on a distributed cluster running GROMACS or another MD engine is possible today as evidenced by the Folding@home successes (Pande et al., 2003). The primary value of engaging in this simulation might be to develop tools and techniques to permit larger simulations of the bulk aqueous phase, lipids and boundaries occupied by smaller molecules such as surfactants. All of these become factors in larger origins of life simulations.

4.2.2 Medium term viability: formation of catalysts, informational molecules and autocatalytic sets

Experiment #3: Riboyzyme Selection Experiment Model

Experiment	**Complexity**	**Tractability**
Bartel & Szostak: Riboyzyme Selection Experiment	10K atoms 100 nanoseconds to one microsecond	Mid term

Another computably tractable early test case would be to reproduce a simplified version of the groundbreaking experimental work by Bartel and Szostak (1993) in the isolation of new ribozymes from a large pool of random sequences. This experiment was entirely carried out within a bead-filled flowing column of reactants.

Simulation Approach

It is plausible that an array of MD simulation systems could be allocated to compute segments of a simplified version of this flow and be able to model the natural selection of ribozymes occurring at the molecular level. EvoGrid optimization and search would be used to create frames modelling the flow of random sequences through the reactor column used in this experiment. New ribozymes would be sought and the source input of the column adjusted using the method similar to the search trees of prototype2010. However, based on advice from Deamer (2011a) this experiment involved ten trillion molecules with 300 randomly placed nucleotides, making even a small subset of it computably intractable. Instead, our simulation could take the approach of applying search to mixtures of many fewer random molecules which show the ability to catalyze their own replication (ligation). A key step would be to fold a few hundred nucleotides within the randomly generated RNA and then select for known active sites which support ligation. In this manner, the EvoGrid could simulate the salient portions of this experiment without having to tackle the large size of the bulk flow of molecules.

Experiment #4: Model for RNA-Making Reactors in Hydrothermal Vents

Experiment	**Complexity**	**Tractability**
RNA-Making Reactors	100-250K atoms Microseconds	Mid term

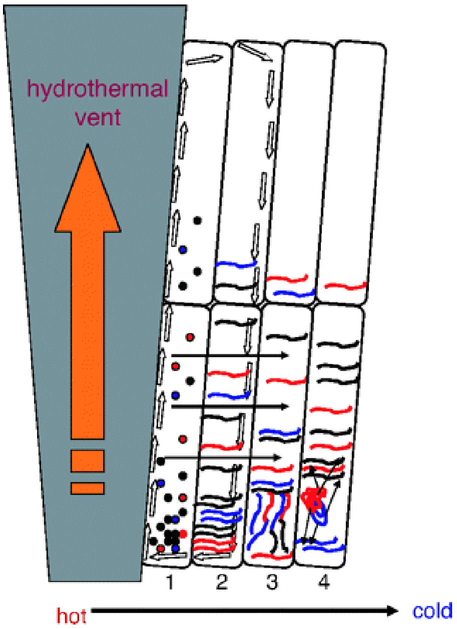


Figure 102 Evolution of an RNA population in a network of inorganic compartments driven by hydrothermal vent flows driven by thermoconvection, and thermophoresis.

(Koonin, 2007) and (Baaske et al., 2007) describe a model (Figure 102) based on seafloor observations of hydrothermal vents in which flows from these vents travel through pores and compartments by thermoconvection and thermophoresis. Early in these flows, compartments (1) serve to support the accumulation of mononucleotides. In subsequent compartments (2), accumulation of abiogenically synthesized RNA molecules occurs followed by exploration of the nucleotide sequence space through ligation and recombination (3) and finally (4) the emergence of an “RNA world” able to continue with processes of natural selection. This putative network of compartments could support the phases of evolution as a dedicated reactor. Such a system was modelled in bulk phase by computer by Baaske et al. (p. 9350). They simulated thermal gradients, microcompartment pore shapes and studied the resulting concentrations of RNA and other molecular products that might be produced in such a reactor.

Simulation Approach

It is not hard to see that there is a good fit for a more granular simulation of this hypothetical system, perhaps at the MD level. Frames of volume simulations could be stacked in a grid overlain on the topography of the compartments. Simulation of the flow through this system would be carried out with molecules handed off from one frame to the next. Search functions would look for the emergence of mononucleotides, RNA, RNA polymerases and the occurrence of competition and selection of RNA sequences. The authors talk about millimetre sized pores producing and concentrating nucleotides at a 10^9-fold accumulation. This is obviously a very large space and a large number of molecules, not including the bulk phase of water. It may, however, be possible to simulate a still-viable miniature version of this already Lilliputian system.

Based on the advice of Deamer (2011a) a simpler approach needs to be considered as the amount of material flowing and the length of the path is even larger than that in the Bartel and Szostak experiment. A frame branching structure driven by searches for the formation of lipid and/or RNA could be applied to small volumes containing a simple heat gradient and a small amount of flowing molecular material. This configuration might suffice to model enough of this experiment to guide future *in vitro* implementations.

Experiment #5: Model of a Hypopopulated Reaction Graph for the Study of Autocatalytic Sets and the Adjacent Possible

Experiment	**Complexity**	**Tractability**
Kauffman: Hypopopulated Reaction Graphs	10K atoms estimated Time not determined	Mid term

More challenging but likely still tractable in the mid term would be to simulate solutions wherein autocatalytic cycles (Kauffman, 1993) may be observed to emerge *de novo* in plausible prebiotic soups. The EvoGrid would deploy frames of a very large number of objects (millions of atoms) but break the simulation into areas handling the boundary issues to support molecules moving between these areas. As mentioned in the section in Chapter 1 of this thesis, discussions with Stuart Kauffman identified an additional future application of the EvoGrid as a "system to model reaction dynamics in the messy world of prebiotic chemistry from which messy life emerged" (Kauffman and Damer, 2011ab). Specifically Kauffman saw a future EvoGrid implementation being capable of taking on "how on a 'hypopopulated' vast reaction graph fluctuations did NOT die out, but advanced into different Adjacent Possibles" (Kauffman and Damer, 2011b). This second goal relates to theories about as yet uncharacterized properties of the universe that sustain a ratcheting of complexity and that these properties are key factors, along with Darwinian natural selection, in the emergence of life. From a private communication with Dr. Wim Hordijk (Hordijk, 2011) who is working with Kauffman and the CERN origin of life group project:

> [w]hat we are interested in is to see what happens when we start with a large graph of possible reactions, but only a very small starting set of molecules. Given the possible reactions and molecule concentrations, what will the flow of these molecules look like over the complete reaction graph? Will we see things like autocatalytic sets forming? We already know they exist in the underlying reaction graph structure (given a high enough level of catalysis), but the question is

> whether (and how quickly) they will actually be realized given some (non-equilibrium/non-ergodic) molecular dynamics starting from a simple "food set".

What is not clear is the number of molecules, size of computing space and time frames involved in creating this large reaction graph. It is also not obvious whether full scale chemical simulation is required or whether abstract, perhaps CA-based artificial chemistries would suffice. However, while discussions on this exciting potential application of the EvoGrid are just beginning, it seems that our work has captured the interest of leading researchers.

Simulation Approach

For now this is the most abstract of the considered experiments, with uncertain chemical models, time frames and populations. The author plans to follow this up with meetings in the near future.

4.2.3 Long term viability: supramolecular structures

Experiment #6: Model for Encapsulation of Polymers in Multilamellar Structures through Wet/Dry Cycles

Experiment	**Complexity**	**Tractability**
Deamer: Polymer Encapsulation in Multilamellar Structures	100K atoms (wet and dry compartments), coarse graining lipid boundaries Microseconds for multiple condensation reactions to occur in both phases	Mid term

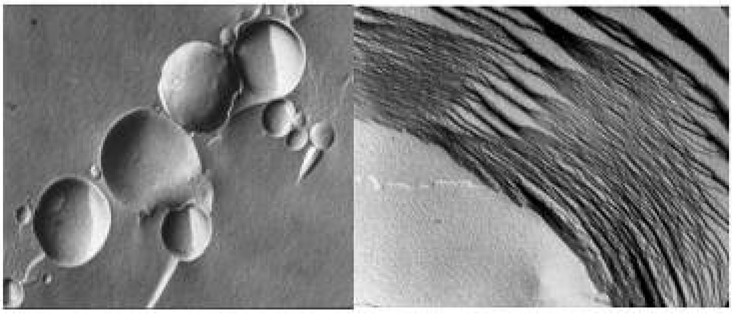


Figure 103 Electron micrograph of vesicles composed of the phospholipid lecithin (left), and multilamellar array of this lipid in a dry state (right) (images courtesy Dave Deamer)

Figure 103 shows the formation of complex molecular structures, particularly lipid containers (left) which flatten out and form two dimensional compartments (right) through a series of wet-dry cycles (Rajamani et al., 2010). The resulting multilamellar bilayers act as an organizing environment concentrating available nucleotides and encouraging the condensation reactions which form of ester bonds and the resulting polymerization of strands of RNA-like molecules. As water has largely diffused out of the environment it is not present to participate in the back reaction and break the ester bonds of the polymers. As a result, the drying environment is able to serve as a dynamic ordering environment for the formation of biologically important macromolecules and upon rehydration those molecules are present in vesicles (Deamer, 2009).

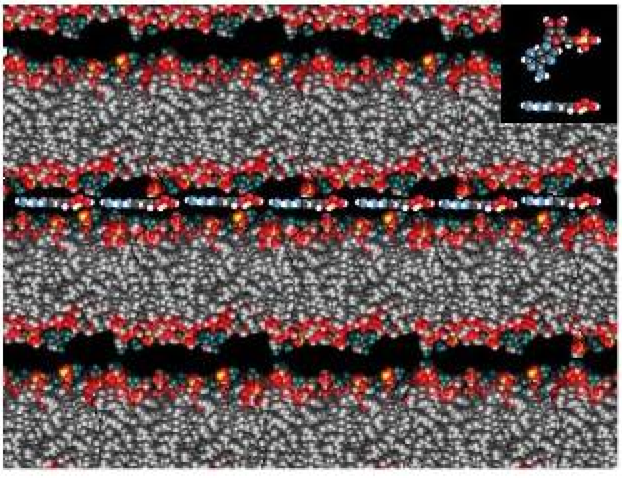


Figure 104 A molecular model of adenosine monophosphate (AMP) a monomer of RNA (inset) organized into a polymeric strand of RNA

trapped between lipid bilayers of a dry state multilamellar array (image courtesy Dave Deamer).

Figure 104 shows a computer model of the multilamellar structures with RNA-like polymeric strand of adenosine monophosphate trapped in between. This result solves a fundamental problem in research on the origin of life by showing a process by which polymers capable of catalysis and replication might have been produced on the early Earth. In this case RNA-like polymers are synthesized non-enzymatically from mononucleotides in lipid environments.

Simulation Approach

According to Deamer (Deamer and Damer, 2011a) a viable and valuable approach to simulating this experiment would be to ask the question: what is the probability of free diffusion forming an ester bond (the basis of polymers and life) in a 3D aqueous medium versus the probability of such bond formation in a 2D, dried medium? The departure of water molecules means that the back reaction (hydrolysis) is less likely so polymerization is more likely in condensation reactions. So the simulation approach for the EvoGrid would be to simulate one large frame of molecules representing the concentration and energies of the wet, encapsulated vesicle contents, and a second frame simulating the dry cycle multilamellar compartment.

No attempt would be made to simulate the drying or the fusing of membranes as according to Deamer this is very complicated. The encapsulating medium (lipid membranes) need not be modelled in detail in either the bulk 3D phase or the 2D multilamellar dried phase. Thus we would use coarse graining techniques to model the membranes, but fine graining MD techniques on the solution phases.

Experiment #7: Model of the FLiNT Protocell Life Cycle

Experiment	**Complexity**	**Tractability**
Rasmussen: FLiNT Protocell Life Cycle	Millions of atoms Microseconds to milliseconds	Long term

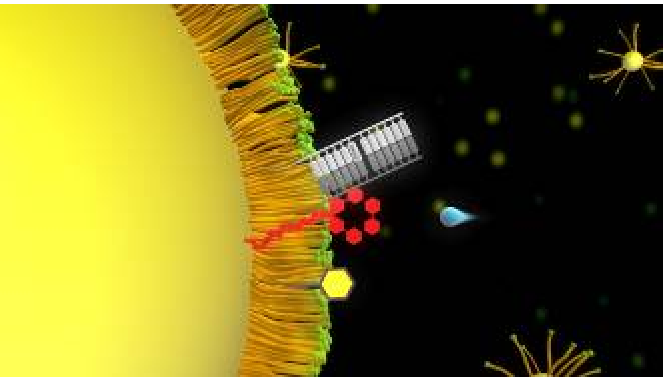


Figure 105 Detailed view of an animation of the FLiNT protocell life cycle: the photonic energized division and eventual ligation of short DNA segments via a ruthenium complex embedded in a surfactant layer around an oil droplet (source: DigitalSpace)

Figure 105 shows an animation for the protocell model being developed by Steen Rasmussen and his group in the Center for Fundamental Living Technology (FLiNT) at the University of Southern Denmark (Rasmussen et al., 2008, Rasmussen et al., 2003b, Fellermann et al., 2007). In this model, informational molecules (short strand DNA) ligate and interact with other molecular assemblages anchored into a surfactant layer surrounding an oil droplet. Thus a 2D membrane forms on the droplet and provides a matrix for the interaction of the system with energy (photons), and the production of new surfactant molecules.

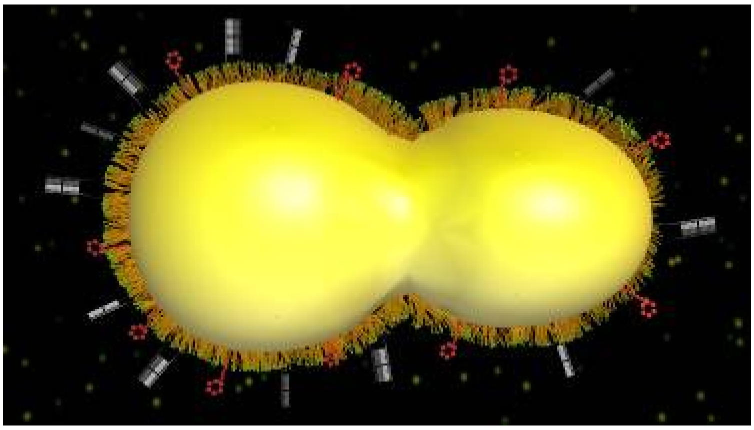


Figure 106 Animation sequence showing the division of the protocell once sufficient oil droplet feeding, ligation of new DNA and absorption of energy has occurred (source: DigitalSpace).

Figure 106 depicts another animation frame in which, after sufficient growth of the underlying oil droplet, the system is dividing, and carrying the surface mounted molecular assemblages along into daughter droplets.

Simulation Approach

Due to the sheer size of the molecular structures in this experiment, extensive coarse-graining would be required. The oil droplet itself would best be simulated as a single entity, similar to a soft body object in classical large body 3D simulations. The bulk phase of water and population of free lipid would also have to be largely simulated as a single, monolithic representation. The MD simulation would be free to focus on the surfactant layer and interactions between the bulk phase and underlying droplet. Individual patches of this layer could be simulated in frames in the EvoGrid with motion within the layer permitting molecules to travel between frames.

4.2.4 Very long term viability: multi-stage supramolecular model for an end-to-end origin of life

Experiment #8: CREATR Model

Experiment	**Complexity**	**Tractability**
Damer et al: CREATR end-to-end OOL model	Millions to billions of atoms Seconds to minutes of real-time simulation	Very long term

The following model is being developed by the author and collaborators (Damer et al., 2011b). While this model is in an early phase of conceptualization it illustrates what was covered in the conclusion to the EvoGrid prototype experiments in Chapter 3: *that any pathway to an origin of life is likely to be explored by a series of chemical and computer simulation experiments in an iterative cycle of increasing fidelity*. The model below is an amalgam of previous work, including elements from the above-described experiments. The model was inspired in part by Freeman Dyson's "double origins" hypothesis (Dyson, 1982, Dyson, 1999) which posits that an origin of life could have occurred through the combination of independently emergent mechanisms for reproduction of a structure and the replication of the coding of inheritable traits. The model proposes a natural system that repeatedly generates a large number of Dyson's "garbage bags" whose contents consists of non-random collections of "dirty water". The plausibility of this model remains to be seen but it can serve as a provisional exemplar of what might be termed an *end-to-end* origin of life model. In this model, which we are calling Complex Repeated Encapsulation of Attached Template Replicators (CREATR), begins with a population of simple prebiotic molecules, traverses several stages, and ends up as a system of macromolecular assemblages reproducing under the control of an informational copying mechanism and subject to adaptation through Darwinian natural selection.

The CREATR model’s first few steps are built on the prior work from the experiments described in this section. Experiment #3 could be a mechanism for producing Riboyzymes, Experiment #5 for providing RNA-making reactors within micropores, Experiment #6 for showing that encapsulation of polymers in multilamellar structures is possible. Later steps rely on results from (Hanczyc and Szostak, 2004) in the formation and replication of vesicles.

Step 1: Association of an informational molecule with a membrane

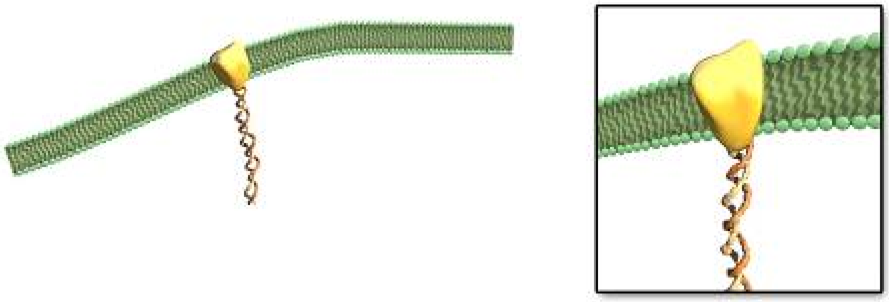


Figure 107 The innovation of anchoring of an informational molecule to a membrane.

(Vlassov et al., 2001) provided evidence that supramolecular RNA complexes can bind to phospholipid bilayers. Polymerization may have been encouraged by the wet-drying cycles described by (Deamer, 2009) thereby resulting in the situation that the RNA complex is closely associated with the membrane (Figure 107). The formation of an RNA complex with the adaptation of anchoring to the membrane is one step, while the parallel mechanism for producing ribozymes to catalyze the production of the RNA or the anchoring mechanism might result from the factors described by (Bartel and Szostak, 1993).

Step 2: Partial pore encapsulation by a membrane

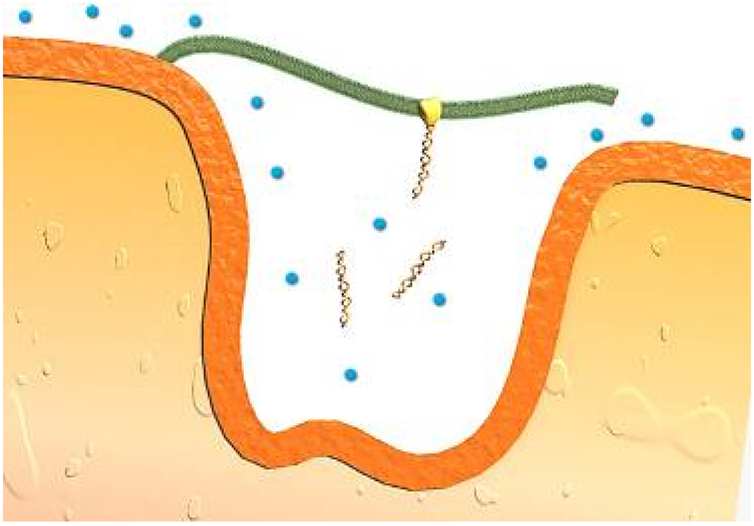


Figure 108 Partial containment of micropore by lipid membrane.

The formation of membranes over small pores either in shoreline environments where wet-drying cycles could occur or associated with hydrothermal vents would produce a small, temporary but free form of encapsulation with the benefit that the system begins acting in a cell-like fashion permitting transmission of molecules passing by the pore through the membrane (Figure 108).

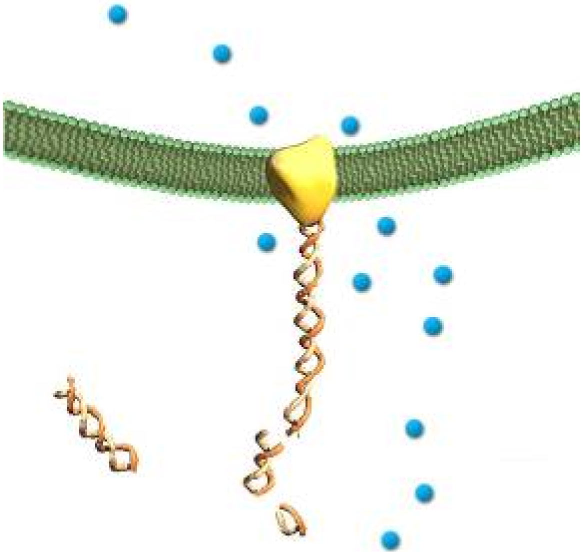


Figure 109 Anchored RNA complex preferentially influencing membrane permeability

The RNA complexes which are anchored to this membrane may promote and regulate the local permeability of the membrane, providing a mechanism to “select” what molecules cross into the micropore (Figure 109). (Vlassov et al., 2001) discovered that the binding of the supramolecular RNA complexes to phospholipid bilayers produces a disruption in the layer which forms a channel. Those molecules may therefore cross into the micropore in the vicinity of the anchored RNA complex thus encouraging any process associated with that complex, notably the templated replication of the complex and its associated anchoring mechanism. The anchoring mechanism may be produced by another process, an autocatalytic set for example. If the anchoring function is ultimately coded for by the nucleobases of the RNA then it will become a directly selected for mutation of the RNA complex.

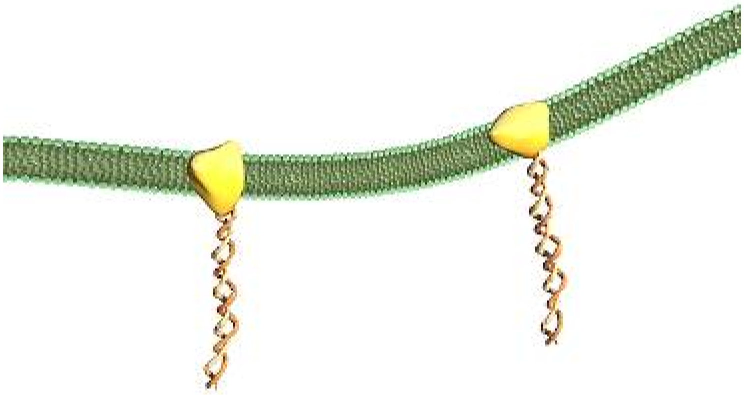


Figure 110 Anchoring of two RNA complexes to the membrane

If the contained system is capable of promoting the templated ligation of the RNA complexes then a second RNA complex may anchor onto the membrane some distance from the original complex (Figure 110).

Step 3: Full encapsulation of a complex

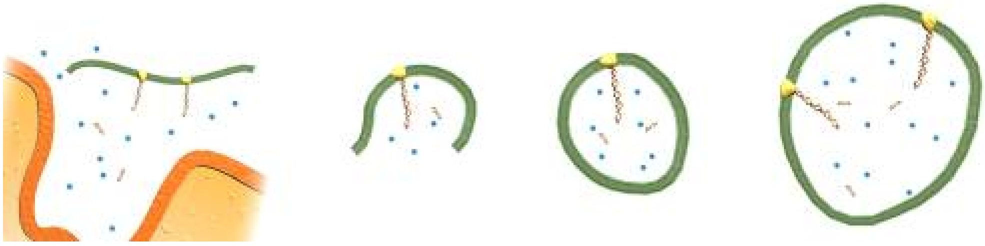


Figure 111 Dissociation of the membrane and associated complex from the micropore, encapsulation of the complex followed by growth

Due to a physical process such as wave action the membrane complex may become detached from the micropore, and carry along with it the affixed RNA complexes and other local contents. This free-floating complex may quickly close to form a full encapsulation. Encapsulations may contain few elements (center) or a number of molecular structures including two or more anchored RNA complexes (right). It is important to note that the original micropore would again be free to support new membrane formation and the activity of partial containment. This and other micropores would therefore act as a factory to produce numerous encapsulations, each with different contents, but possibly somewhat similar for each individual micropore. Therefore, a population of vesicles encapsulating a variety of complexes would be released into a local region.

Step 4: Division of vesicles

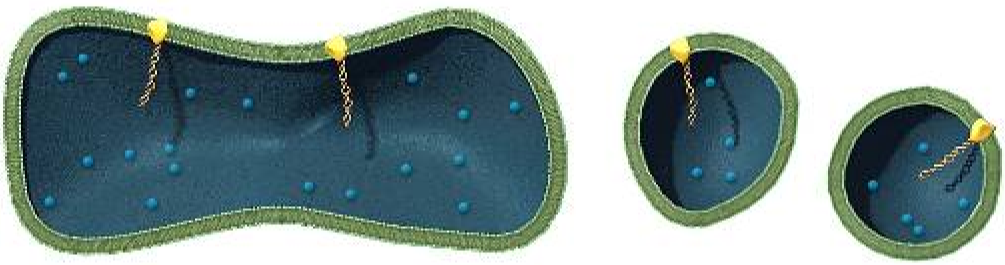


Figure 112 Division of protocell vesicles through growth of surrounding lipid membrane

These vesicles, or “protocells”, would “compete” for the building blocks of their membranes (lipids) and absorb other small molecular components of the environment which would enable internal activities. One key activity might be the replication of membrane-embedded RNA complexes which are themselves acting as primitive regulators of membrane permeability. Contained catalysts in concert with active functions on the RNA complexes might also mediate a sort of metabolism driven across the membrane at the disruption channels. Another key activity, perhaps performed by other contained RNA polymers, is the regulation of membrane growth, permitting the protocell to enjoy a more stable existence than non-regulated protocells. In a conversation with David Deamer about the CREATR model, he posed the question (Deamer and Damer, 2011b): *how could these encapsulated molecules regulate the tendency for larger vesicles to break down into smaller ones, i.e. could the presence of these molecules destabilize the larger membrane and activate the process of division at a specific time?* For Deamer it is this control mechanism between the encapsulated molecules and the growth and point of division of the vesicle that is a key point in the emergence of a plausible protocell. Finally as the protocell grows (Figure 112, left), the elongation that precedes a possibly controlled division into daughter protocells (right) (Hanczyc and Szostak, 2004) may force attached RNA complexes apart so that they are statistically more probable to end up in a separate daughter protocell. Of course other important metabolic molecular machinery would also have to somehow end up in two copies, one in each daughter cell. Returning to Dyson’s reasoning, catalysts might statistically, rather than exactly, reproduce this machinery.

Step 5: Development of lengthening competing protocell lines

At this point the entire viability of any given protocell is determined by the combination of its structure, contents, and activity both in the interior and across the membrane. Therefore protocells which do not support an increased probability of the successful and complete division of daughter cells will tend toward extinction. Due to error rates at each stage, the probability of successful division must tend toward a high likelihood (a local maximum) quickly or the entire "line" would be lost. Dyson points out that if the probability of at least one daughter protocell containing a complete self-reproducing population of molecules is greater than one half, then multiple functional daughters will result in a chain reaction producing a "line" of statistically reproduced protocells. Natural selection as described by Darwin would come into effect even in this inexact statistical reproduction of protocell lines. Competing lines would over time refine the mechanisms of their catalysts, the effectiveness of the channels, and the regulatory and replicating capability of the RNA complexes. If after a few divisions a given line of protocells eventually dies off there will always be the up-stream micropores able to produce a new combination to try again. In addition if the broken parts of destroyed protocell lines are retained in the local region, they could serve as building blocks for the next micropore encapsulation. In addition, membrane fragments with affixed complexes from destroyed protocells might combine with other protocell fragments to create a new protocell line. This would be an early form of sexual recombination with the "genes" of the affixed RNA complexes of both fragments forming a union of sorts and possibly producing new beneficial functions. Through this trial and error process the lines of protocells might therefore persist for more and more generations until they can avoid catastrophic failure and persist essentially forever. When a protocell line persists forever, then life's origin has occurred.

Summary

Our model represents a form of *combinatorial chemistry* (Deamer and Damer, 2011b) in which millions of encapsulated vesicles could be produced, each representing a different "natural experiment". Most of these experiments would end with the dissolution of the vesicle and loss of its contents but if only a single encapsulated complex held the right combination of contents and active pathways to support the properties discussed above, life would have a chance to get going. Of course, beyond the regulation of growth and the formation of controlled channels, such vesicles would have to also possess the machinery to capture energy and nutrients, and the ability to copy these mechanisms such that this capability is propagated to the daughter cells. Presumably the membrane attached RNA complex could also do double duty encoding the mechanisms to catalyze the construction of this machinery. At some point the multipurpose role of the RNA as membrane pore mechanism, growth regulator and other functions would bifurcate into separate molecular complexes and we would witness the emergence of a cell genome, ribosomes, energy mechanisms and other separate functions.

Simulation Approach

The CREATR Model is an example of a multi-stage end-to-end origin of life simulation challenge alluded to in section 4.1. Future EvoGrids would have to be capable of simulating and searching for behavior amongst millions or billions of atoms over time scales of seconds to minutes in order to process experiments surrounding the stages of this model. Parallel *in vitro* laboratory experimentation would have to be carried out in analogs to this environment to disqualify, verify and calibrate the *in silico* experiments.

4.3 Open Problems for Computational Origins of Life Endeavours

Inspired by the open questions for artificial life posed by Bedau et al. (2000) cyberbiogenesis computational origins of live endeavours also have a number of related open issues which are summarized below. The first set of open problems is related to the construction of systems like the EvoGrid and is listed next in rough order of importance.

4.3.1 Open Problems in Computational Origins of Life Related to Future EvoGrid-like Systems

- Problem #1: The EvoGrid cannot escape the teleologically-derived problem of all designed simulation environments: if we set up and simulate a system acting in the ways we accept as probable, then that system is much less likely to act in improbable and potentially informative ways, as results are always constrained by the abstractions and assumptions used. Another way of stating this very central conundrum is that as long as we do not know how chemical molecules might be able to exhibit emergence of important characteristics such as replication we will not be able to design the fitness functions to actually select for these molecules or their precursors. The fitness-function generation problem is as yet unsolved. However, the EvoGrid framework is being built to: 1) allow each potential experimenter to code in their own definition of fitness, accumulating knowledge applicable to the problem in an iterative fashion; and 2) support a more exotic solution in which the search functions themselves 'evolve' or 'emerge' alongside the simulation being searched. Actually building the second option would first require a much more extensive treatment from the field of information theory.

- Problem #2: Bedau et al. (2000) call for creating frameworks for synthesizing dynamical hierarchies at all scales. The heterogeneous nature of EvoGrid simulations would allow for coarse-graining procedures to focus simulation from lower levels to higher ones, saving computing resources by shutting off the less critical, more detailed simulations below. An example of this would be to switch to coarse grained simulation of an entire lipid vesicle, ceasing simulation of individual vesicle wall molecules. Conversely, fine grained simulations could be turned on for locally important details, such as templated replication of an informational molecule. However, it should be noted that interfacing different software engines and representations of simulation space is notoriously difficult. Running the same simulation space at multiple levels employing multiscale physics from quantum and molecular dynamical, to coarse grained dissipative particle dynamics, and beyond to smooth particle hydrodynamics is a very challenging problem that awaits future research.

- Problem #3: A general theory of cameo chemical simulations needs to be developed to understand the minimum number of interacting objects and physical simulation properties required in these simulations for the emergence of “interesting” phenomena pertinent to life’s building blocks. Our hypothesis that the *genes of emergence* in cameo simulations would apply to larger simulations also needs to be tested in the context of more ambitious COoL efforts capable of supporting artificial evolution.

- Problem #4: Related to Problem #3 is the question of at what level to simulate, i.e.: what volume of space, quantity of objects, interactions, and what time scales should a simulation accommodate? If, for example, the level chosen were molecular dynamics then due to limitations on computing power it would force simulations to exist in only within nanometer or micrometer volumes for far sub-second duration.

- Problem #5: How does one design a simulation in which organization goes from simpler to more complex forms through multiple levels of organization in an open ended fashion? Related to this is the question of what kind of tools must be developed to determine whether complexification is actually occurring within the simulation? The problem of simulations reaching points where all phenomena that can be observed have been observed is well known in the Alife field. In chemistry combinatorial explosions can result in terminated experiments consisting of black tars. The achieving of simulations and frameworks that support multiple levels of complex emergent phenomena is one of the greatest challenges in computer science.

- Problem #6: How can a simulation be developed such that the automated observer functions or human operators that are examining the simulation state do not unduly bias the simulation compared to an un-observed version? This is the back end of the front end problem of teleology stated in Problem #1. Initial conditions will bias the experiment at the front end, but observation and selection can bias or be detrimental for emergent phenomena at the back end. Isolation of the simulation engines from the observation mechanisms is one way to limit operator bias.

4.3.2 Open Problems in Computational Origins of Life as a Field

The emerging practices of cyberbiogenesis *in silico* and *in vitro* simulation have a number of higher level open questions which this work has brought into focus:

- Problem #7: How will it be possible to know whether an object in a simulation is to be judged “artificially alive” (assuming the “strong” Alife hypothesis that the entity is a true form of living system) when previously the simulation had only parts or behaviors judged “artificially non-alive”? Related to this is the problem of how to define life itself (Dawkins and Damer, 2000). A related question is whether there a kind of “Turing Test” that could be devised to test the strong Alife hypothesis? This challenge will be taken up in the section titled *An Origin of Artificial Life Turing Test for the EvoGrid* later in this chapter.

- Problem #8: How much influence in the simulation is permissible such that the emergence observed is not considered “intelligent design”? Of course a computer simulation framework by definition is “intelligently designed” (refer to Abel in section 1.4) but presumably at least some of the emergent phenomena observed are not consciously anticipated. This question is intimately involved with the history of evolutionary theory. Darwin himself began his book *Origin of Species* (Darwin, 1859) with a discussion of human-directed selection in the breeding of domesticated animals.

- Problem #9: Can cyberbiogenesis pathways be successfully explored through the use of digital, electronic computers or is this beyond the capability of even a large grid of machines or a dedicated supercomputer based on the von Neumann design? Are more exotic computing technologies or architectures required to simulate nature at the level of chemistry? Abel's skepticism versus Bentley's more optimistic Systemic Computing approach discussed in section 1.4 yields some insight here.

- Problem #10: Would abstract universes with more tractable simulations be a wiser direction to take versus attempts at simulating nature at the chemical level? If an abstract universe is selected, what are the minimum required physical approximations necessary for an artificial origin of life event to occur? This assumes that all required physical phenomena can be successfully simulated which in itself is an open question.

- Problem #11: If an artificial chemistry is selected for the approach, at what level must we target the simulation: quantum dynamics, molecular dynamics, coarse-graining mesoscale simulation (Cieplak and Thompson, 2008, Ortiz et al., 2005), or a hybrid approach of two or more levels (i.e., multiscale simulation)? Also related is if higher levels are selected to construct the model, does this departure from the "fundamental chemistry" of the simulation put at risk the viability of an artificial origin of life emerging in the simulation?

- Problem #12: How can the problem of emergence be addressed through the solving of the problem of our own perception (Gordon, 1998)? A related question is: how could the simulation itself develop its own perception, which seems to be a prerequisite to self organization and living processes?

- Problem #13: How can science best develop the techniques to enhance the likelihood of the emergence of highly improbable events through automated algorithmic shortcuts such as stochastic hill climbing or enrichment techniques (Kalos and Whitlock, 2008)?

- Problem #14: As there is no universal definition of what makes an object alive (Ablondi, 1998), it is reduced to being "merely" a configuration of molecules that function together in specific roles (containment, metabolism, or replication for example). Is it then possible to manipulate a set of simulated molecules towards imitating a living organism? This would discover at least one pathway from nonliving to living matter without the added overhead of a hands-off black box approach to emergent phenomena. Could this kind of "intelligent coaxing" produce a meaningful result that could assist in cyberbiogenesis? The reverse case is also interesting: taking apart a living cell into its non-living components. Would a re-assembly of non-living components that produces a more minimal yet still viable cell provide insight into pathways to the origin of that minimal cell?

- Problem #15: Langton coined the term "artificial life" (Langton, 1989) and envisaged an investigation of "life as it could be". Should cyberbiogenesis systems be constrained to models of the emergence of life on Earth? More abstract simulations may shine a light on life as it might be out in the universe (Gordon and Hoover, 2007), as a tool for use in the search for extraterrestrial intelligence (SETI) (Damer and Brown, 2010, Gordon and Hoover, 2007), or as a separate technogenesis within computing or robotic worlds.

- Problem #16: A critic of theories of chemical evolution, cosmologist Sir Fred Hoyle used the statement about a ready-to-fly 747 aircraft being assembled by a tornado passing through a junk yard of parts (Hoyle, 1984) to ridicule the idea of spontaneous generation of life at its origin. This idea today fuels creationist claims for irreducible complexity as one of their strongest arguments for the existence of a Creator. Should therefore practitioners of cyberbiogenesis efforts take on these challenges? (Gordon, 2008) and (Barbalet and Daigle, 2008, Barbalet et al., 2008) recently engaged this theme through dialogues about the origins of life and the debate between creationists and scientists.

- Problem #17: Gordon predicted (Gordon, 2008, p. 359) that "Alife enthusiasts have an opportunity to solve the 'Origin of Artificial Life' problem well before the chemists will solve the 'Origin of Life' problem". The question begged by this statement is whether or not the field of origins of life should place more resources on the *cyber* part of *cyber*biogenesis and what investments would be made early on to establish that Gordon's claim is in fact plausible? A related question is whether or not the field should wait for computing power to increase further before making such investments? The author of this thesis started on this work in 1985 and opted to wait through almost a quarter century of progress in computing before returning to it.

- Problem #18: What capabilities will digital simulation ever be able to offer chemistry or biology especially in origins of life questions? Any given computational system might be able to show fascinating emergent phenomena but there is a real possibility that such discoveries might well stay trapped *in silico* and never transition over to be replicable *in vitro*. In this case would a new duality of *non-overlapping magisteria* emerge similar to Stephen J. Goud's (Gould, 1997) non-overlapping magisteria of science and religion?

4.4 Societal Considerations Posed by the EvoGrid Endeavour

Given the likely distant prospects for any successful cyberbiogenesis effort, perhaps a valuable preparatory activity in the interim is to consider the societal impact of such a project on a range of human endeavors. Any enterprise that sets as its goal the emergence of an *artificial origin of life*, testable in chemistry and therefore ultimately realizable as a new form of chemical life is likely to draw in controversy from many quarters. Considering this likely controversy will serve us today to uncover a number of conundrums that lie at the basis of science, technology, religion and philosophy.

4.4.1 Scientific Conundrums

The goals of cyberbiogenesis beg basic questions in science including:

1. How does science define a living entity or a whole living system? Will scientists simply “know it when they see it” when what might be considered a *bona fide* living system is observed in a digital simulation? These questions were discussed during the author’s visit with Professor Richard Dawkins at his home in Oxford (Dawkins and Damer, 2000) and summarized in Appendix B.1.

2. What is the experiment to be undertaken and at what point does it start? Do experiments begin with pre-built components of some complexity but not considered to be living, and proceed from there as suggested by Gordon (2008)? Or should simulation experiments be initiated much further down the ladder with simpler artificial precursor molecules, or even farther from basic atoms assembling precursor molecules within an *ab initio* primal soup?

3. How much influence is required to induce a sufficient measure of emergence? In other words, how much “intelligent design” is required in the setting up and operating of cyberbiogenesis experiments? What degree of ongoing human guidance should be permitted in both the virtual and chemical experiments which follow?

4. Would an entirely artificially evolved entity pose a current or future threat to any part of natural environment in the Earth’s biosphere or to technological or biological elements within human civilization? How could such a threat be mitigated? If such a threat were possible, is it grounds for not pursuing this line of research?

4.4.2 Technological Conundrums

A decade ago the Artificial Life community took stock of their field and proposed a set of Open problems in Artificial Life (Bedau et al 2000) which provide a clear look at the brickwork of the technological foundations of any serious cyberbiogenesis effort. The authors set a challenge in the second open problem to study abiogenesis in an artificial chemistry and identifying that “[b]etter algorithms and understanding may well accelerate progress… [and] combinations of… simulations… would be more powerful than any single simulation approach" (p. 367-68). The authors also point out that while the digital medium is very different from molecular biology, it "has considerable scope to vary the type of 'physics' underlying the evolutionary process" and that this would permit us to "unlock the full potential of evolution in digital media" (p. 369). Ten years later as projects such as the EvoGrid take a run at a simulated abiogenesis, the technological conundrums have come to the fore:

1. What level(s) do you simulate at, and at what scale? Is molecular dynamics a sufficient level to simulate at, or are quantum dynamical effects required? Or is a more abstract artificial chemistry which can exhibit desired properties a better starting point than aiming at high fidelity to chemistry?

2. Nature operates in parallel at multiple scales with multiple physical properties emerging from these scales. So how can von Neumann computers (essentially serial processors) be adapted to meet this challenge or does this challenge belong to the domain of special purpose hardware or an amalgam of digital and chemical computing?

3. What computational corners can be cut but still retain plausibility in nature and viability in experimental chemistry? Related to this is the claim by Abel (2009b) that any computational simulation is formulaic, subject to predicative knowledge and not based on physicodynamic factors so may never be representative of solutions *in vitro*. In addressing this question Gordon presents the following possibility for future EvoGrid implementations:

    > Consider having the EvoGrid simulate a less plausible approximation to chemistry. Allow a more abstract chemistry to be tested which also might be subject to a proof by construction in mathematics. The components will be decent approximations of real chemistry. Allow yourself to introduce all the bias that you want but as long

> as the program constrains you to do things that are physically realistic then you might argue that you have something artificially alive. You just don't have the pathway to the end point but you know there is a way back. Decomposed parts could be markers on many paths to life. The important point is proving that one such path exists (Gordon et al., 2010).

4. How much do the search functions and human designed initial conditions and sought after end points to experiments limit their ultimate creativity? This is the problem of systems employing a teleological approach: bias toward the sought after goals limits the usefulness of the system as an open ended discovery mechanism. Evolution does not strive toward goals. Even though nature cannot be praised for the best "designed" solutions to problems it also cannot be faulted for teleological bias. Gordon makes the following points along this line:

> Examine the case of the EvoGrid where you act as the intelligent designer and use the tools to determine the minimal artificial organism. If you could then put one together then you could look for the properties and potential pathways to that minimal artificial organism. You could also consider an experiment where you start with bigger building blocks that are considered to be non alive and see if they assemble into something you would consider to be alive (Gordon et al., 2010).

4.4.3 Religious, Ethical, and Philosophical Conundrums

The goals of cyberbiogenesis endeavours will attract questions and controversy in society including:

1. Does a future successful cyberbiogenesis disprove the need for a supernatural creator as an agent in the origin of life and for the guiding of life's evolution?

2. What is the consequence for the world's religions of the creation of an artificially alive (in computer simulations) or a chemically alive entity?

3. Would an artificially sourced living entity be protected as an endangered species? Would only the chemical entity be protected or the virtual one as well?

4. Does the enterprise of cyberbiogenesis represent a willful achievement of human innovation or is it an inevitable expression of the entire biosphere and life itself, with humans as mere agents forwarding life into a new mechanism of evolution? Is this the means by which life is expanding itself out into other parts of our solar system or the universe? Are we willing or unwilling agents of this expansion?

In a discussion of ethical concerns in (Rasmussen et al., 2003b, p. 67) the authors echo some of the above:

> Generating life de novo will create public reactions. The reactions will probably be along two lines: (i) Environmental concerns that the life-producing technology could "get out of control", and (ii) Religious and moral concerns, based on the beliefs that humankind must restrain from certain endeavors on grounds that they are fundamentally amoral.

Ethical questions arising around the possible creation of cells from existing biology or completely new molecular constructions have a storied history. Non-technical press reaction from announcements in biotechnology and genomics such as the research on minimal cell research (Fraser et al., 1995) and the announcement of the sequencing of the human genome (Venter et al., 2001) often turns to talk of "Frankencells". Concerns about more artificial *de novo* nanostructures able to survive and reproduce in natural environments have also been discussed in the nanotechnology community (Merkle, 1992). It is clear that future work on the EvoGrid or any cyberbiogenesis-oriented system will have to plan to address the above issues. Initially the response to concerns might be to state that lifelike objects in a simulation are merely objects in a simulation and of no threat to people or the biosphere. The argument might be made that these objects might be a threat to computer networks, however, akin to computer viruses. However, the size of virtual environment needed to sustain an artificially alive system would be so complex and large that the system would effectively be an isolated island, similar to large database systems. If, however, there is an active effort to reproduce these lifelike objects in physical chemistry, the alarm will be raised with many in the public and scientific community.

4.4.4 An Origin of Artificial Life Turing Test for the EvoGrid

Related to concerns about Frankencells is the question of: how do you know when something is lifelike enough, especially in a computer simulation, to declare it “alive”? In his 1950 paper *Computing Machinery and Intelligence* (Turing, 1950) Alan Turing wrote “I propose to consider the question 'Can machines think?” (p. 433) and defined a test of machine intelligence, one variation of which consisted of a human judge conversing (through a text interface) with an unseen human and a similarly hidden computer (p. 442). If the human judge could not tell the difference then the machine would be judged to have reached some sort of cognitive equivalence to the human participant.

At some point in the near or far future, a group of engineers, biologists, philosophers and others may be assembled in a room. In the next room, project specialists would be observing a rich array of lifelike objects moving about in a virtual world running on some future EvoGrid. The esteemed guests may be asked to undertake a sort of updated *Origin of Artificial Life Turing Test*, wherein they hear or read abstract descriptions of the objects and environments witnessed but not revealed by the staff. They would also be exposed to descriptions of real, living entities and their environments. Over many hours or days they would be able to ask questions about both environments. If in the end if a majority of the visitors cannot consistently tell which environment is in fact the real, biological one and which is the one witnessed in the simulation then the EvoGrid has produced a progeny which has passed this form of Turing Test. Of course there will likely be strong arguments for and against the "aliveness" of the EvoGrid entities, especially if they exist in an abstract universe far from the norms of chemistry. If, at some later date, a chemistry faithful EvoGrid entity is fabricated with molecules and survives to breed within a physical soup, then the concerns of the doubters may be quelled. Ironically the likely optimal stage at which an EvoGrid entity is fabricated and tested in chemistry is when the entity is in its embryo stage.

4.4.5 A Visionary View: A Lens on Universes of Life

Let us now roll forward to some far future date when working cyberbiogenesis systems abound and when one can witness the closure of the loop whereby the observation of *in vitro* adaptations feeds back to changes in the *in silico* ecosystem. Going even further, numerous types of entities and environments could be simulated, extending our ability to model origins of life in alien habitats and to cast light onto *life as it might be* in the universe. Of course there may well be a substantial range of viable artificial living systems for which there would exist no physical medium in which they could be instantiated. In this case the only universe in which that these creatures could be chemically rendered out of the simulation into physical reality is a parallel one possessed of truly exotic physics. We arrive to the conclusion that some nearly infinitely endowed future cyberbiogenesis system could serve as a telescope (or microscope), a lens into where in this universe or others life might arise and projecting how far it might evolve. Indeed, in the unlikely event that an intelligent higher form of life should arise in a simulation, an idea today very securely in the realm of science fiction, would we choose to instantiate it physically or seek out where its naturally evolved cousins might be resident? Presumably at that point that form of life would have to have its own say in the matter.

4.5 Summary and Contributions to Knowledge

In this work we sought to create both intrinsic and extrinsic contributions to knowledge. The major intrinsic contribution surrounds innovations around optimizations for the distributed computing of artificial chemistries, notably molecular dynamics. The extrinsic contributions deal with the wider relevance of such innovations as tools that can support endeavours in the field of the study of possible pathways to the origin of life on Earth.

The EvoGrid prototype which was designed, built and operated for this work provided a proof by implementation that a distributed small volume molecular dynamics simulation could be coupled with a global search optimization function enabling emergent phenomena to be more effectively generated. Let us conclude by summarizing the contributions to knowledge made by this work. The core intrinsic contribution to knowledge is the establishment of the architectural principles of such systems, a system design, specific engineering solutions and, finally, implementation and experiments to test the stated hypothesis:

> *Distributed processing and global optimization employing search coupled with stochastic hill climbing can produce significant performance improvements in the generation of emergent phenomena within small volume, short time frame molecular dynamics simulations over non-optimized solutions*.

From the analysis of the data from the operation of the prototype it was concluded that the prototype verified that the optimization methods *did produce significant performance improvements in terms of time saved and computational products produced over a non-optimized solution, in one case generating a full order of magnitude more molecular bonds*. A discovery was made that the simulation is highly sensitive to initial conditions, especially within the logic of the search function and any degradation factors. This is a confirmation of common knowledge in the design of global optimization systems. This would require the designers of future experiments to carefully manage this process. Other intrinsic benefits to science accruing from the optimized design include the demonstration of reconfigurable search functions, open and extensible interfaces, and scalability across hardware networks. With further development, we feel that this design is therefore suitable to support a variety of origins of life endeavours.

Several key extrinsic contributions to knowledge were also provided in the course of this work and reported in this thesis:

- A history of artificial life and chemical simulation endeavours as well as a literature review of informative fields contributing to the design of viable computational origin of life frameworks.
- A vision and definition of a new term *cyberbiogenesis* which captures the marriage of *in silico* computer simulation and *in vitro* chemical experiment for origin of life endeavours.
- A map of the major cognate fields that illustrate how these fields inform cyberbiogenesis enterprises.
- A listing of current limitations and a technical roadmap for the improvement of the current EvoGrid prototype and a roster of experiments in origins of life research to which future EvoGrid platforms may be applied.
- A series of open questions both for the EvoGrid and for an emerging field of computational origins of life simulation.
- An illustration and discussion of scientific, philosophical, religious, and general societal conundrums posed by this line of research.

Final Thoughts

This thesis aims to introduce a major innovation into the field of research on the origins of life. The research described in this thesis makes an original and substantial contribution to this and other fields of knowledge by successfully designing, building and characterizing a unique discovery system, the EvoGrid. This system combines chemical simulation and search with hill climbing techniques to more optimally permit complex behavior to arise within small volume, distributed molecular dynamics simulations. I hope that this study provides both inspiration and methodologies of substantial use to future scholars and practitioners who, within the next few decades, will help to develop the new field of *cyberbiogenesis*. Through the work of these future researchers, systems derived from the experience of the first EvoGrid may cast some light on the dark unknowns of the origins of life on Earth and in the universe. If a descendant of this work were to “achieve the transition to life in an artificial chemistry *in silico*” (Bedau et al., 2000) then the “Evo” in EvoGrid would have been realized and Darwinian natural selection or another form of open ended evolution would be witnessed for the first time in a digital simulation.

Towards the end of the research work for this thesis and based on the recommendations of colleagues and friends, I obtained a copy of Greg Egan’s 1994 novel *Permutation City* (Egan, 1994). Set in the mid 21st Century, Egan gives a clear account of a hypothetical computing system, the *Autoverse*, that was successful in creating an *in silico* abiogenesis. What was surprising was Egan’s clear prescience about the actual details of the technical implementation and challenges that have been uncovered in building an actual system such as the EvoGrid. Indeed, the EvoGrid itself could be seen to be a significant step along the road to realizing Egan’s vision. For Egan starts his journey in the world of low level cellular automata from which, one day in the first decade of the Twenty First Century, the first EvoGrid’s atoms and molecules then arose.

It is therefore fitting to conclude this work with the following quotation from Egan’s *Permutation City*:

> Real-world biochemistry was far too complex to simulate in every last detail for a creature the size of a gnat, let alone a human being. Computers could model all the processes of life -- but not on every scale, from atom to organism, all at the same time (p. 26)… The individual cubic cells which made up the Autoverse were visible now, changing state about once a second. Each cell's "state" -- a whole number between zero and two hundred and fifty-five -- was recomputed every clock cycle, according to a simple set of rules applied to its own previous state, and the states of its closest neighbors in the three-dimensional grid. The cellular automaton which was the Autoverse did nothing whatsoever but apply these rules uniformly to every cell; these were its fundamental "laws of physics." Here, there were no daunting quantum-mechanical equations to struggle with – just a handful of trivial arithmetic operations, performed on integers. And yet the impossibly crude laws of the Autoverse still managed to give rise to "atoms" and "molecules" with a “chemistry" rich enough to sustain "life." (pp. 29-30)

Bibliography

ABEL, D. L. 2009a. The biosemiosis of prescriptive information. *Semiotica,* 174**,** 1-19.

ABEL, D. L. 2009b. The capabilities of chaos and complexity. *Int J Mol Sci,* 10**,** 247-91.

ABEL, D. L. 2010. Constraints vs. Controls. *Open Cybernetics and Systemics Journal,* 4.

ABEL, D. L. 2 May 2011 2011. *RE: Personal communication with regard to the EvoGrid and concepts of the Cybernetic Cut and Abel's approach to formal versus physical systems.*

ABLONDI, F. 1998. Automata, living and non-living: Descartes' mechanical biology and his criteria for life. *Biology & Philosophy,* 13**,** 179-186.

ANDERSON, D. P. 2004. BOINC: A system for public-resource computing and storage. *Proceedings of the 5th IEEE/ACM international Workshop on Grid Computing, International Conference on Grid Computing.* Washington, DC: IEEE Computer Society.

ANDREWS, S. S. & BRAY, D. 2004. Stochastic simulation of chemical reactions with spatial resolution and single molecule detail. *Phys Biol,* 1**,** 137-51.

BAASKE, P., WEINERT, F. M., DUHR, S., LEMKE, K. H., RUSSELL, M. J. & BRAUN, D. 2007. Extreme accumulation of nucleotides in simulated hydrothermal pore systems. *Proc Natl Acad Sci U S A,* 104**,** 9346-51.

BARBALET, T. 2008. *Biota Podcasts* [Online]. Available: http://www.biota.org/podcast [Accessed January 6 2010].

BARBALET, T. & DAIGLE, J. P. 2008. *Biota Podcast 19* [Online]. Available: http://www.archive.org/download/biotapodcasts/biota_052308.mp3 [Accessed April 8 2010].

BARBALET, T., DAMER, B. & GORDON, R. 2008. *Biota Podcast 44* [Online]. Available: http://www.archive.org/download/biotapodcasts/biota_032909.mp3 [Accessed April 8 2010].

BARRICELLI, N. 1962. Numerical Testing of Evolution Theories: Part II. *Acta Biotheoretica,* 16.

BARRICELLI, N. A. 1953. *Experiments in Bionumeric Evolution Executed by the Electronic Computer at Princeton, N. J.*, Archives of the Institute for Advanced Study, Princeton, NJ.

BARTEL, D. P. & SZOSTAK, J. W. 1993. Isolation of new ribozymes from a large pool of random sequences [see comment]. *Science,* 261**,** 1411-8.

BAUM, R. 2003. *NANOTECHNOLOGY Drexler and Smalley make the case for and against 'molecular assemblers'* [Online]. Chemical & Engineering News. Available: http://pubs.acs.org/cen/coverstory/8148/8148counterpoint.html [Accessed 21 March 2011].

BEDAU, M. A., MCCASKILL, J. S., PACKARD, N. H., RASMUSSEN, S., ADAMI, C., GREEN, D. G., IKEGAMI, T., KANEKO, K. & RAY, T. S. 2000. Open problems in artificial life. *Artif Life,* 6**,** 363-76.

BENTLEY, P. J. 2007. Systemic Computation: A Model of Interacting Systems with Natural Characteristics. *In:* ADAMATZKY, A., TUEUSCHER, C. AND ASAI, T. (ed.) *Special issue on Emergent Computation in Int. J. Parallel, Emergent and Distributed Systems (IJPEDS).* Oxford: Taylor & Francis pub.

BENTLEY, P. J. 2009. Methods for improving simulations of biological systems: systemic computation and fractal proteins. *J R Soc Interface,* 6 Suppl 4**,** S451-66.

BOWERS, K., CHOW, E., XU, H., DROR, R., EASTWOOD, M. P., GREGERSEN, B. A., KLEPEIS, J. L., KOLOSSVARY, I., MORAES, M. A., SACERDOTI, F. D., SALMON, J. K., SHAN, Y. & SHAW, D. E. 2006. Scalable Algorithms for Molecular Dynamics Simulations on Commodity Clusters. *Supercomputing 2006.* Tampa, Florida: IEEE.

BROWNLEE, J. 2011. Clever Algorithms: Nature-Inspired Programming Recipes. Lulu.

CAMPBELL, B. 2003. *Review of Digital Biota 3 and Oworld Summit* [Online]. Available: http://bdcampbell.net/oworld/ [Accessed 21 March 2011].

CHAMPIN, P. A., BRIGGS, P., COYLE, M. & SMYTH, B. 2010. Coping with noisy search experiences. *Knowledge-Based Systems,* 23**,** 287-294.

CIEPLAK, M. & THOMPSON, D. 2008. Coarse-grained molecular dynamics simulations of nanopatterning with multivalent inks. *J Chem Phys,* 128**,** 234906.

CORMAN, S. R. 2011. *Complex Systems Problems in the War of Ideas* [Online]. Journal of the Terrorism Research Initiative. Available: http://www.terrorismanalysts.com/pt/index.php/pot/article/view/30/html [Accessed 21 March 2011].

DAMER, B. 1995. *Amoeba: A Simulator for Molecular Nanotechnology, presented at the Fourth Foresight Conference on Molecular Nanotechnology* [Online]. Palo Alto, California. Available: http://www.digitalspace.com/papers/nanopap1.html [Accessed 21 March 2011].

DAMER, B. 1996. Inhabited Virtual Worlds: A New Frontier for Interaction Design. *ACM interactions.* ACM Press.

DAMER, B. 1997. *Avatars: Exploring and Building Virtual Worlds on the Internet,* Berkeley, Peachpit Press.

DAMER, B. 1997-2001. *Biota Conference Series 1997-2001* [Online]. Available: http://www.biota.org/conferences/ [Accessed 21 March 2011].

DAMER, B. 2001. *DigiBarn: A Brief Telling of the Story of the Elixir Products 1987-94* [Online]. DigiBarn Computer Museum. Available: http://www.digibarn.com/collections/software/elixir/gallery/index.html [Accessed 21 March 2011].

DAMER, B. 2003a. *Cray Supercomputer Collection at the DigiBarn Computer Museum* [Online]. Available: http://www.digibarn.com/collections/systems/crays/index.html [Accessed].

DAMER, B. 2003b. A Virtual Walk on the Moon. *In:* LAUREL, B. (ed.) *Design Research.* Cambridge, Massachussetts: MIT Press.

DAMER, B. 2008. The God detector: a thought experiment. *In:* SECKBACH, J. & GORDON, R. (eds.) *Divine Action and Natural Selection: Science, Faith and Evolution.* Singapore: World Scientific.

DAMER, B. 2010. *DigitalSpace Corporation Projects 1995-2010* [Online]. Available: http://www.digitalspace.com [Accessed 31 December 2010].

DAMER, B. 2011. Search for books with term "origin of life" in their title Amazon.com 3/21/2011 revealed 879 volumes. 2011 ed.: Amazon.com.

DAMER, B. & BROWN, A. 2010. *The EvoGrid: Building an Origin of Life Simulator & Its Implications for Life, the Universe and Everything*

[Online]. Menlo Park, CA: SETI Institute. Available: http://www.seti.org/csc/lecture/archive/2010 [Accessed June 10 2010].

DAMER, B. & FURMANSKI, T. 2005. Nerve Garden. *In:* ADAMATZKY, A. & KOMOSINSKI, M. (eds.) *Artificial Life Models in Software.* Berlin: Springer.

DAMER, B., GOLD, S. & DE BRUIN, J. 2000. Conferences and Trade Shows in Inhabited Virtual Worlds: A Case Study of Avatars98 & 99. *In:* HEUDAN, J. C. (ed.) *Virtual Worlds, Second International Conference, VW 2000.* Paris, France: Springer.

DAMER, B., NEWMAN, P., GORDON, R., BARBALET, T., DEAMER, D. & NORKUS, R. 2010. The EvoGrid: A Framework for Distributed Artificial Chemistry Cameo Simulations Supporting Computational Origins of Life Endeavors. *Proceedings of the 12th Conference on Artificial Life*.

DAMER, B., NEWMAN, P. & NORKUS, R. 2011a. *EvoGrid project web site* [Online]. Available: http://www.evogrid.org [Accessed 10 April 2011].

DAMER, B., NEWMAN, P., NORKUS, R., GRAHAM, J., GORDON, R. & BARBALET, T. 2011b. Cyberbiogenesis and the EvoGrid: A 21st Century Grand Challenge. *In:* SECKBACH, J. G., R. (ed.) *Genesis: Origin of Life on Earth and Planets [in preparation].* Dordrecht, Springer.

DAMER, B., NORKUS, R. & DHIREN, D. 2008. *EvoGrid "The Movies" and Script and Storyboards Concept* [Online]. Available: http://www.evogrid.org/evogrid-movie/index.html [Accessed].

DAMER, B., RASMUSSEN, D., NEWMAN, P., NORKUS, R. & BLAIR, B. 2006. Design Simulation in Support of NASA's Robotic and Human Lunar Exploration Program. *AIAA Space 2006.* NASA Ames Research Center, Moffett Field, CA.

DARWIN, C. 1859. *On the origin of species by means of natural selection,* London,, J. Murray.

DARWIN, C. 1871. Letter to Hooker. *Darwin Online Library*.

DAWKINS, R. 1986. *The Blind Watchmaker: Why the Evidence of Evolution Reveals a Universe Without Design*, W.W. Norton & Company.

DAWKINS, R. & DAMER, B. 10 Jul 2000 2000. *RE: Conversation between Richard Dawkins and Bruce Damer regarding the concept of a competition to create an artificial origin of life.*

DE BONET, J. S., ISBELL, L. & VIOLA, P. 1997. *MIMIC: Finding Optimal by Estimating Probability Densities*, MIT Press.

DE GROOT, A. 1969. *Methodology: Foundations of Inference and Research in the Behavioral Sciences*, Mouton De Gruyter.

DEAMER, D. 2009. Compartments And Cycles: Testing An Origin Of Life Hypothesis. *Science 2.0* [Online]. Available from: http://www.science20.com/stars_planets_life/compartments_and_cycles_testing_origin_life_hypothesis [Accessed 2 April 2011].

DEAMER, D. & DAMER, B. 29 January 2010a. *RE: Coversation about the EvoGrid and interstellar chemistry.*

DEAMER, D. & DAMER, B. 14 October 2010 2010b. *RE: Personal communication on the EvoGrid and modeling interstellar chemistry.*

DEAMER, D. & DAMER, B. 2 May 2011 2011a. *RE: Conversation on chemical experiments which might become simulation targets for the EvoGrid.*

DEAMER, D. & DAMER, B. 2 May 2011 2011b. *RE: Conversation with David Deamer on the Complex, Free Encapsulation model.*

DEAMER, D. W. 2011. *First life : Discovering the Connections between stars, cells, and how life began,* Berkeley, University of California Press.

DEAMER, D. W. & BARCHFELD, G. L. 1982. Encapsulation of macromolecules by lipid vesicles under simulated prebiotic conditions. *J Mol Evol,* 18**,** 203-6.

DEAMER, D. W. & PASHLEY, R. M. 1989. Amphiphilic components of the Murchison carbonaceous chondrite: surface properties and membrane formation. *Orig Life Evol Biosph,* 19**,** 21-38.

DICTIONARY, M. W. O. 2011. *Teleology definition* [Online]. Available: http://www.merriam-webster.com/dictionary/teleology [Accessed].

DITTRICH, P., ZIEGLER, J. & BANZHAF, W. 2001. Artificial chemistries--a review. *Artif Life,* 7**,** 225-75.

DYSON, F. J. 1982. A model for the origin of life. *J Mol Evol,* 18**,** 344-50.

DYSON, F. J. 1999. *Origins of Life,* Cambridge, Cambridge University Press.

DYSON, G. 1997. *Darwin among the machines : the evolution of global intelligence,* Reading, Mass., Addison-Wesley Pub. Co.

EDBOAS. 2011. *Physics on a molecular scale (diagram)* [Online]. Available: http://en.wikipedia.org/wiki/File:MM_PEF.png [Accessed 30 March 2011].

EGAN, G. 1994. *Permutation city,* New York, HarperPrism.

FARKIN, B. & DAMER, B. 2005. BrahmsVE: From human-machine systems modeling to 3D virtual environments. *Proceedings of the 8th International Workshop on Simulation for European Space Programmes SESP 2004.* Noordwijk, the Netherlands.

FELLERMANN, H. 2009a. *Physically Embedded Minimal Self-Replicating Systems – Studies by Simulation.* PhD, Universit¨at Osnabr¨uck.

FELLERMANN, H. 2009b. Spatially resolved artificial chemistry. *In:* ADAMATZKY, A., KOMOSINSKI, M. (ed.) *Artificial Life Models in Software 2nd Edition.* Springer.

FELLERMANN, H. 30 Mar 2011 2011. *RE: Personal communication with B. Damer regarding the computable tractability of the FLiNT nanocell model.* Type to B., D.

FELLERMANN, H., RASMUSSEN, S., ZIOCK, H. J. & SOLE, R. V. 2007. Life cycle of a minimal protocell--a dissipative particle dynamics study. *Artif Life,* 13**,** 319-45.

FINE, R. D., DIMMLER, G. & LEVINTHAL, C. 1991. FASTRUN: A special purpose, hardwired computer for molecular simulation. *Proteins: Structure, Function, and Genetics,* 11**,** 242–253.

FRASER, C. M., GOCAYNE, J. D., WHITE, O., ADAMS, M. D., CLAYTON, R. A., FLEISCHMANN, R. D., BULT, C. J., KERLAVAGE, A. R., SUTTON, G., KELLEY, J. M., FRITCHMAN, R. D., WEIDMAN, J. F., SMALL, K. V., SANDUSKY, M., FUHRMANN, J., NGUYEN, D., UTTERBACK, T. R., SAUDEK, D. M., PHILLIPS, C. A., MERRICK, J. M., TOMB, J. F., DOUGHERTY, B. A., BOTT, K. F., HU, P. C., LUCIER, T. S., PETERSON, S. N., SMITH, H. O., HUTCHISON, C. A., 3RD & VENTER, J. C. 1995. The minimal gene complement of Mycoplasma genitalium. *Science,* 270**,** 397-403.

FREDKIN, E. 2004. Special issue: Theoretical aspects of cellular automata. *Theoretical Computer Science,* 325.

GARDNER, M. 1970. Mathematical Recreations: The Game of Life. *Scientific American,* October 1970.

GOODSELL, D. S. 2009. *The machinery of life,* New York, Copernicus Books.

GORDON, R. 1966. *Lattice Membrane with Nearest Neighbor Interactions [16mm movie],* Eugene, University of Oregon.

GORDON, R. 1967. *Steady State Properties of Ising Lattice Membranes [Chemical Physics Ph.D. Thesis, Supervisor: Terrell L. Hill],* Eugene, Oregon, Department of Chemistry, University of Oregon.

GORDON, R. 1968a. Adsorption isotherms of lattice gases by computer simulation. *J. Chem. Physics,* 48**,** 1408-1409.

GORDON, R. 1968b. Steady-state properties of Ising lattice membranes. *J. Chem. Physics,* 49**,** 570-580.

GORDON, R. 1980. Monte Carlo methods for cooperative Ising models. *In:* KARREMAN, G. (ed.) *Cooperative Phenomena in Biology.* New York: Pergamon Press.

GORDON, R. 1998. The emergence of emergence through the evolution of perception.

GORDON, R. 2008. Hoyle's tornado origin of artificial life, a computer programming challenge. *In:* GORDON, R. & SECKBACH, J. (eds.) *Divine Action and Natural Selection: Science, Faith and Evolution.* Singapore: World Scientific.

GORDON, R., DAMER, B. & NEWMAN, P. 2010. *RE: Conversation between Richard Gordon, Bruce Damer and Peter Newman prior to the implementation of the full EvoGrid design.*

GORDON, R. & HOOVER, R. B. 2007. Could there have been a single origin of life in a big bang universe? *Proc. SPIE,* 6694**,** doi:10.1117/12.737041.

GOULD, S. J. 1997. Non-Overlapping Magisteria. *Natural History.*

GROOT, R. D. & WARREN, P. B. 1997. Dissipative particle dynamics: Bridging the gap between atomistic and mesoscale simulation. *J. Chem. Phys.,* 107**,** 4423–4435.

HALDANE, J. B. S. 1927. *Possible worlds and other essays,* London,, Chatto & Windus.

HANCZYC, M. M. & SZOSTAK, J. W. 2004. Replicating vesicles as models of primitive cell growth and division. *Curr Opin Chem Biol,* 8**,** 660-4.

HASSLACHER, B. 1987. *Discrete Fluids*.

HICKINBOTHAM, H., FAULCONBRIDGE, A. & NELLIS, A. 2010. The BlindWatchmaker's Workshop: three Artificial Chemistries around Eigen's Paradox. *In:* FELLERMANN, H., DÖRR, M., HANCZYC, M.,LADEGAARD LAURSEN, L., MAURER, S., MERKLE, D., MONNARD, PA., STOY, K., RASMUSSEN, S. (ed.) *Proceedings of the Twelfth International Conference on the Synthesis and Simulation of Living Systems.* Odense, Denmark: MIT Press.

HOOGERBRUGGE, P. & KOELMAN, J. 1992. Simulating microscopic hydrodynamic phenomena with dissipative particle dynamics. *Europhys. Lett.,* 19**,** 155–160.

HORDIJK, W. 20 March 2011 2011. *RE: Personal communication with Dr. Wim Hordijk of the CERN group on origins of life.* Type to B., D.

HOYLE, F. 1984. *The Intelligent Universe,* New York, Holt Rinehart and Winston.

HUTTON, T. J. 2007. Evolvable self-reproducing cells in a two-dimensional artificial chemistry. *Artif Life,* 13**,** 11-30.

JONES, S. 2003a. *My Other Computer Is a Supercomputer* [Online]. Linux Journal. Available: http://www.linuxjournal.com/node/7090/print [Accessed 11 March 2011].

JONES, S. 2003b. *My Other Computer Is a Supercomputer* [Online]. Linux Journal. Available: http://www.linuxjournal.com/node/7090/print [Accessed 4 April 2011].

KALÉ, L., SKEEL, R., BHANDARKAR, M., BRUNNER, R., GURSOY, A., KRAWETZ, N., PHILLIPS, J., SHINOZAKI, A., VARADARAJAN, K. &

SCHULTEN, K. 1999. NAMD2: Greater Scalability for Parallel Molecular Dynamics. *Journal of Computational Physics,* 151**,** 283-312.

KALOS, M. H. & WHITLOCK, P. A. 2008. *Monte Carlo Methods,* Weinheim, WILEY-VCH Verlag GmbH & Co. KGaA.

KAUFFMAN, S. 2010. *Can A Changing Adjacent Possible Acausally Change History? The Open Universe IV* [Online]. NPR. Available: http://www.npr.org/blogs/13.7/2010/02/can_a_changing_adjacent_possib.html [Accessed 20 March 2011].

KAUFFMAN, S. & DAMER, B. 18 March 2011 2011a. *RE: Communication from Stuart Kauffman on EvoGrid effort to CERN Origins of Life group.*

KAUFFMAN, S. & DAMER, B. 15 March 2011 2011b. *RE: Conversation with Stuart Kauffman on the creation of a simulation of a large hypopopulated reaction graph.*

KAUFFMAN, S. A. 1993. *Origins of Order: Self-Organization and Selection in Evolution,* New York, Oxford University Press.

KAUFFMAN, S. A. 1995. *At home in the universe : the search for laws of self-organization and complexity,* New York, Oxford University Press.

KAUFFMAN, S. A. 2000. *Investigations,* Oxford, Oxford University Press.

KIRKPATRICK, S., GELATT, C. D., JR. & VECCHI, M. P. 1983. Optimization by simulated annealing. *Science,* 220**,** 671-80.

KOONIN, E. V. 2007. An RNA-making reactor for the origin of life. *Proc Natl Acad Sci U S A,* 104**,** 9105-6.

KOSZTIN, D., BISHOP, T. C. & SCHULTEN, K. 1997. Binding of the estrogen receptor to DNA. The role of waters. *Biophys J,* 73**,** 557-70.

LACK, D. 1940. "Evolution of the Galapagos Finches". *Nature,* 146**,** 324–327.

LANGTON, C. G. 1984. Self-reproduction in cellular automata. *Physica D,* 10**,** 135–144.

LANGTON, C. G. 1986. Studying artificial life with cellular automata. *Physica D,* 22**,** 120-149.

LANGTON, C. G. 1989. *Artificial life : the proceedings of an interdisciplinary workshop on the synthesis and simulation of living systems held September, 1987 in Los Alamos, New Mexico,* Redwood City, Calif., Addison-Wesley.

LANGTON, C. G., TAYLOR, C., FARMER, J. D. & RASMUSSEN, S. 1992. *Artificial Life II,* Reading, Addison-Wesley.

LEVY, S. 1993. *Artificial Life: The Quest for a New Creation,* New York, Vintage Books.

LIMBACH, H., ARNOLD, A., MANN, B. A. & HOLM, C. 2006. ESPResSo – an extensible simulation package for research on soft matter systems. *Comput. Phys. Commun.,* 174**,** 704–727.

LLOYD, S. 2006. *Programming the universe : a quantum computer scientist takes on the cosmos,* New York, Knopf.

LUISI, P. L. & DAMER, B. F. May, 2011 2009. *RE: A private conversation with Luigi Luisi regarding the necessity of simulating solvents.*

MARGULIS, L. 1997. Beliefs and biology: Theories of life and living. *Isis,* 88**,** 522-523.

MARGULIS, L. & SAGAN, D. 2000. *What is life?,* Berkeley, University of California Press.

Author. 2009. Wanted: Home Computers to Join in Research on Artificial Life. *New York Times*, September 29, 2009.

MERESCHCOWSKY, K. 1909. *The Theory of Two Plasms as the Basis of Symbiogenesis, a New Study or the Origins of Organisms*.

MERIAM-WEBSTER. 2011. *Teleology definition* [Online]. Available: http://www.merriam-webster.com/dictionary/teleology [Accessed].

MERKLE, R. 1992. The Risks of Nanotechnology. *In:* LEWIS, B. C. J. (ed.) *Nanotechnology -Research and Perspectives.* Cambridge, MA: MIT Press.

MILLER, S. L. 1953. A production of amino acids under possible primitive earth conditions. *Science,* 117**,** 528-9.

MITCHELL, M. 2009. *Complexity : a guided tour,* Oxford England ; New York, Oxford University Press.

NAKANO, A., KALIA, R. K., NOMURA, K., SHARMA, A., VASHISTA, P., SHIMOJO, F., VAN DUIN, A. C. T., GODDARD, W. A., BISWAS, R. & SRIVASTAVA, D. 2007. A divide-and-conquer/cellular-decomposition framework for million-to-billion atom simulations of chemical reactions. *Computational Materials Science,* 38**,** 642-652.

NELLIS, A. & STEPNEY, S. 2010. Automatically moving between levels in Artificial Chemistries. *In:* FELLERMANN, H., DÖRR, M., HANCZYC, M.,LADEGAARD LAURSEN, L., MAURER, S., MERKLE, D., MONNARD, PA., STOY, K., RASMUSSEN, S. (ed.) *Proceedings of the Twelfth International Conference on the Synthesis and Simulation of Living Systems.* Odense, Denmark: MIT Press.

NVIDIA. 2011. *CUDA Zone Web Site* [Online]. Available: http://www.nvidia.com/object/cuda_home_new.html [Accessed 2 April 2011].

O'CONNOR, K. 1994. *The alchemical creation of life (takwin) and other concepts of Genesis in medieval Islam (PhD dissertation, UPenn 1994).* PhD, Univeristy of Pennsylvania.

OFRIA, C. & WILKE, C. O. 2004. Avida: a software platform for research in computational evolutionary biology. *Artif Life,* 10**,** 191-229.

OPARIN, A. I. & MORGULIS, S. 1938. *The origin of life,* New York,, The Macmillan Company.

ORTIZ, V., NIELSEN, S. O., DISCHER, D. E., KLEIN, M. L., LIPOWSKY, R. & SHILLCOCK, J. 2005. Dissipative particle dynamics simulations of polymersomes. *J Phys Chem B Condens Matter Mater Surf Interfaces Biophys,* 109**,** 17708-14.

PANDE, V. 2011. *Recent Research Papers from Folding@home* [Online]. Available: http://folding.stanford.edu/English/Papers [Accessed].

PANDE, V. S., BAKER, I., CHAPMAN, J., ELMER, S. P., KHALIQ, S., LARSON, S. M., RHEE, Y. M., SHIRTS, M. R., SNOW, C. D., SORIN, E. J. & ZAGROVIC, B. 2003. Atomistic protein folding simulations on the submillisecond time scale using worldwide distributed computing. *Biopolymers,* 68**,** 91-109.

PASKO, A., ADZHIEV, V. & COMNINOS, P. (eds.) 2008. *Heterogeneous Objects Modelling and Applications*: Springer.

PHILIPS, R. & MILO, R. 2009. A feeling for the numbers in biology. *Proceedings of the National Academy of Sciences,* 106**,** 21465–21471.

PHILLIPS, J. C., BRAUN, R., WANG, W., GUMBART, J., TAJKHORSHID, E., VILLA, E., CHIPOT, C., SKEEL, R. D., KALE, L. & SCHULTEN, K. 2005. Scalable molecular dynamics with NAMD. *J Comput Chem,* 26**,** 1781-802.

PLIMPTON, S. J. 1995. Fast parallel algorithms for short-range molecular dynamics. *J Comp Phys,* 117**,** 1-19.

POPPER, K. R. 1959. *The logic of scientific discovery,* New York,, Basic Books.

RAJAMANI, S., VLASSOV, A., BENNER, S., COOMBS, A., OLASAGASTI, F. & DEAMER, D. 2010. Lipid-assisted Synthesis of RNA-like Polymers from Mononucleotides. *In:* SCHWARTZ, A. W. (ed.) *Origins of Life.* Springer.

RANGARAJAN, A. & DAMER, B. 26 June 2010 2010. *RE: Personal communication on the subject of the current state of complexity theory vis. a vis. the EvoGrid.*

RASMUSSEN, S., BEDAU, M. A., CHEN, L., DEAMER, D., KRAKAUER, D. C., PACKARD, N. H. & STADLER, P. F. (eds.) 2008. *Protocells: Bridging Nonliving and Living Matter,* Cambridge: MIT Press.

RASMUSSEN, S., BEDAU, M. A., RAVEN, M. & KEATING, G. 2003a. Collective intelligence of the artificial life community on its own successes, failures, and future. *Artif Life,* 9**,** 207-235.

RASMUSSEN, S., CHEN, L., NILSSON, M. & ABE, S. 2003b. Bridging nonliving and living matter. *Artif Life,* 9**,** 269-316.

RAY, T. S. 1991. An approach to the synthesis of life. *In:* LANGTON, C., C. TAYLOR, J. D. FARMER, & S. RASMUSSEN (ed.) *Artificial Life II, Santa Fe Institute Studies in the Sciences of Complexity.* Addison-Wesley.

RAY, T. S., HART, J. 1998. Evolution of Differentiated Multi-threaded Digital Organisms. *In:* C. ADAMI, R. K. B., H. KITANO, AND C. E. TAYLOR [EDS.] (ed.) *Artificial Life VI proceedings.* The MIT Press.

RUSSELL, S. J. & NORVIG, P. 2003. *Artificial intelligence : a modern approach,* Upper Saddle River, N.J., Prentice Hall/Pearson Education.

SANDFORD, S. A., ALEON, J., ALEXANDER, C. M., ARAKI, T., BAJT, S. & ET_AL. 2006. Organics captured from comet 81P/Wild 2 by the Stardust spacecraft. *Science,* 314**,** 1720-4.

SEGRE, D. & LANCET, D. 1999. A statistical chemistry approach to the origin of life. *Chemtracts – Biochemistry and Molecular Biology 12***,** 382–397.

SHAW, D. E. & DROR, R. 2008. Anton, a special-purpose machine for molecular dynamics simulation. *Communications of the ACM,* 51**,** 91–97.

SHAW, D. E., DROR, R. 2009. Millisecond-scale molecular dynamics simulations on Anton. *Proceedings of the ACM/IEEE Conference on Supercomputing (SC09).* Portland, Oregon: ACM/IEEE.

SHAW, D. E., MARAGAKIS, P., LINDORFF-LARSEN, K., PIANA, S., DROR, R. O., EASTWOOD, M. P., BANK, J. A., JUMPER, J. M., SALMON, J. K., SHAN, Y. & WRIGGERS, W. 2010. Atomic-level characterization of the structural dynamics of proteins. *Science,* 330**,** 341-6.

SHENHAV, B. & LANCET, D. 2004. Prospects of a computational origin of life endeavor. *Origins of Life and Evolution of Biospheres,* 34**,** 181-194.

SIMS, K. 1991. Artificial evolution for computer graphics. *ACM Computer Graphics,* 25**,** 319-328.

SOLÉ, R., SARDANYES, J., FELLERMANN, H., VALVERDE, S., MACIA, J. & A., M. 2009. Models of protocell replication. *In:* RASMUSSEN, S. E. A. (ed.) *Protocells: Bridging Nonliving and Living Matter.* MIT Press.

STAHL, F. 1961. *On Artificial Universes* [Online]. [Accessed].

STEWART, J. J. 1990. MOPAC: a semiempirical molecular orbital program. *J Comput Aided Mol Des,* 4**,** 1-105.

STEWART, J. P. 2009. *Open MOPAC Home Page* [Online]. Available: http://openmopac.net [Accessed 2 April 2011].

SZOSTAK, J. W., BARTEL, D. P. & LUISI, P. L. 2001. Synthesizing life. *Nature,* 409**,** 387-90.

TAIJI, M., NARUMI, T., OHNO, Y., FUTATSUGI, N., SUENAGA, A., TAKADA, N. & KONAGAYA, A. 2003. Protein explorer: A petaflops special-purpose computer system for molecular dynamics simulations.

Proceedings of the ACM/IEEE Conference on Supercomputing (SC03).

TAYLOR, L., SCHMITT, H., CARRIER III, D. & NAKAGAWA, M. 2005. The Lunar Dust Problem: From Liability to Asset. *Proceedings of the 1 st Space Exploration Conference: Continuing the Voyage of Discovery.* Orlando, FL.

TOYODA, S., MIYAGAWA, H., KITAMURA, K., AMISAKI, T., HASHIMOTO, E., IKEDA, H., KUSUMI, A. & MIYAKAWA, N. 1999. Development of MD engine: High-speed accelerator with parallel processor design for molecular dynamics simulations. *Journal Computational Chemistry,* 20**,** 185–199.

TREVORROW, A. & ROKICKI, T. 2011. *Golly Game of Life Simulator Home Page* [Online]. Sourceforge. Available: http://golly.sourceforge.net/ [Accessed 26 March 2011].

TURING, A. M. 1937. On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Math. Soc.,* s2-42**,** 230-265.

TURING, A. M. 1950. Computing machinery and intelligence. *Mind,* 59**,** 433-460.

VAN DER SPOEL, D. 2011. *GROMACS Web Site* [Online]. Available: http://www.gromacs.org/About_Gromacs [Accessed 29 March 2011].

VAN DER SPOEL, D., LINDAHL, E. & HESS, B. 2005. GROMACS: Fast, flexible, and free. *Journal of Computational Chemistry,* 26**,** 1701-1718.

VENTER, J. C., ADAMS, M. D., MYERS, E. W., LI, P. W. & ET_AL. 2001. The sequence of the human genome. *SCIENCE,* 291**,** 1304-51.

VICSEK, T. 2002. The bigger picture. *Nature,* 418**,** 131.

VILBRANDT, T., MALONE, E., LIPSON, H. & PASKO, A. 2008. Universal Desktop Fabrication. *In:* PASKO, A., ADZHIEV, V. & COMNINOS, P. (eds.) *Heterogeneous Objects Modelling and Applications.* Springer.

VILLA, E., BALAEFF, A. & SCHULTEN, K. 2005. Structural dynamics of the lac repressor-DNA complex revealed by a multiscale simulation. *Proc Natl Acad Sci U S A,* 102**,** 6783-8.

VLASSOV, A., KHVOROVA, A. & YARUS, M. 2001. Binding and disruption of phospholipid bilayers by supramolecular RNA complexes. *Proc Natl Acad Sci U S A,* 98**,** 7706-11.

VON NEUMANN, J. & BURKS, A. W. 1966. *Theory of self-reproducing automata,* Urbana,, University of Illinois Press.

WATSON, J. D. & COOK-DEEGAN, R. M. 1991. Origins of the Human Genome Project. *FASEB J,* 5**,** 8-11.

WIGNER, E. P. 1960. The unreasonable effectiveness of mathematics in the natural sciences. *Communications in Pure and Applied Mathematics,* 13**,** 1-14.

WIKIPEDIA. 2011. *Definition of term abiogenesis* [Online]. Available: http://en.wikipedia.org/wiki/Abiogenesis#Current_models [Accessed 20 April 2011].

WOLFRAM, S. 2002. *A new kind of science,* Champaign, IL, Wolfram Media.

Appendices

Appendix A. Relevant and Previously Published Work

A.1 Personal Recollection on the Origins of the EvoGrid

In 1985 I was in graduate school at the University of Southern California in Los Angeles. There I found myself well endowed with computing resources by virtue of our department's VAX 11/750 minicomputer, with a solid connection to the mid 1980s incarnation of the Internet. Together with an advanced Tektronix graphics terminal I set off programming a system to represent thousands of entities which would one day become a sort of digital adaptive soup into which I could pour a series of problems in image processing, and the entities would dutifully evolve the capability to accomplish pattern recognition in a much more expeditious way than simple linear algorithms or computationally costly methods such as the fast Fourier transform. My first problems were related to collegial resources: I had none. Nobody in my group in optical materials and devices, my advisor included, really had an understanding or an interest in this approach. I was able to find a compelling book published in the 1950s by John von Neumann on self-reproducing automata but that was about the extent of the literature that I could find at the time. A field called "Artificial Life" was to emerge a couple of years later, but too late for me, as I took my leave of university to pursue a series of careers in the outer world.

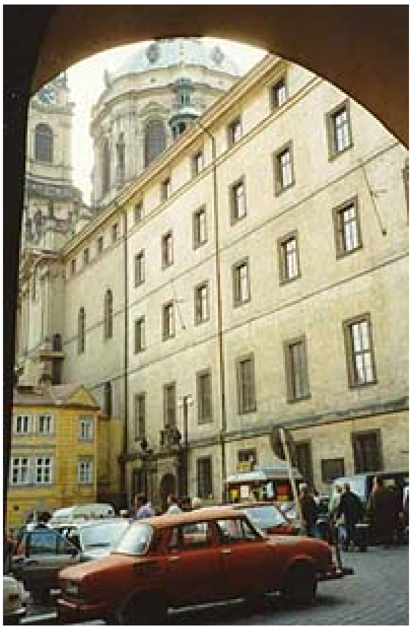


Figure 1 Charles University Math/Physics Faculty, Prague, 1993 (photo by the author)

It is the spring of 1993 and I am standing at the front of a heavily vaulted 16th century classroom at the Mathematics and Physics Faculty of Charles University in Mala Strana (Figure 1), Prague. The iron curtain has recently dissolved away and I am in the country setting up one of the first commercial software groups for Elixir Technologies Corporation founded by my mentor Basit Hamid. As a member of the staff of a computer science department with practically no computers, I am trying to impart the art of computer programming to bright faced young Czechs. I put chalk to the black board and start to sketch out visions for team projects these students will take on in a real personal computer lab we are building on the floor below.

The projects offered were in innovative user interfaces, and object-oriented programming, but the one that drew forth my passion, was to create a model of simple life forms that swim about in a digital universe, compete and evolve new traits. I realized even then that this project is way beyond the scope of a semester or two of part time work for these students. But I wanted to put it out to them anyway. This had been something I have attempted to do myself for over a decade but life's basic needs had always intervened. I finished sketching out the projects and then glanced through the window where, towering above the university was Prague Castle. I realized with no small sense of irony that it was there four centuries ago that some of the first experimentalist, the alchemists, were chartered by Holy Roman Emperor Rudolf II to find a way to "bring dead matter to life" amongst other miracles. Another view from the Faculty building took one's gaze over to the old Jewish quarter and the cemetery where arose the legend of the Golem, a programmatic clay humanoid robot gone amok through one change in the lettering of his code. I then recalled that the very term "robot" was coined by Czech writer Karel Capek from the word robota meaning "labor" or "drudgery". The automated figurines that emerge day after day from the clock in the nearby Prague old town square must have further inculcated the Bohemian mind to accept the automated person as an equal.

It struck me that humankind's quest to create life from parts that are not alive is very, very old. While it is a quest that predates Czech or even European civilization, I realized in that moment that it was here in central Bohemia that the belief in the inevitability of this outcome is perhaps the most deeply culturally ingrained. My selection of an "artificial life" project for these students to toil on in the very shadow of Prague castle paid intentional homage to this long tradition. While the students didn't get far in that eight month project, I went on to carry this quest for the next decade, into virtual world Cyberspace, up to the Burgess Shale fossil beds, and on through a decade of work in modeling space exploration with NASA.

In the spring of 1997 I found myself walking up to the entrance of Harvard University's Museum of Comparative Zoology. I had an appointment to see renowned palaeontologist Professor Stephen J. Gould. Despite my nervous state, I was going forward with this visit in the confidence that I could build bridges between two very different communities: palaeontology and computer science. This mission had started with a lifelong fascinating with how living systems work at their lowest level. After being bit by the computer software bug in 1981 at our local community college I began to pursue a vision of *virtual worlds*, shared graphical spaces inhabited by users, but also be entities which would be governed by their own rules, some of them inspired by biology. In the mid 1990s I engaged in early seminal work in this area, founding the Contact Consortium, the first organization dedicated to multi-user virtual worlds on the Internet. I also authored the first book on the medium and some of the first publications.

During the formative period in the summer of 1994 I visited Chris Langton at the Santa Fe Institute. Dr. Langton had coined the term "Artificial Life" in the 1980s and had helped to establish the Institute and semi-annual Artificial Life conference. I painted a picture for him of virtual worlds of a near future Internet which would be populated by millions of users and billions of objects. I posed the question: “wouldn’t these worlds be ideal places for experiments in artificial evolution to take place”? This vision drove me on to California and the establishment of the Consortium and one of its first special interest groups, Biota.org in 1996. Biota.org created a forum for ideas and projects at the intersection of virtual worlds with biology. The first project of Biota.org was to build a virtual world that had generative biologically inspired elements, in this case digital plants “grown” by the L-System formalism defined by Lindenmeyer. This virtual world was presented as an installation at SIGGRAPH 1997 in Los Angeles and included in computer graphics publications. The second project of Biota.org was to initiate a conference series by holding an innovative first event that would attract a unique combination of people from different worlds. This is why I found myself in the office of Professor Gould at Harvard.

As I had grown up in the mountains of south central British Columbia, Canada, I had heard about the famous Burgess Shale fossil deposits discovered by Charles Doolittle Walcott in the early 1900s. The fossils found in the Burgess Shale were the best preserved of representatives of fauna from the Middle Cambrian period over a half billion years ago. Professor Gould had written a book about the Burgess Shale called *Wonderful Life* which I read in the mid 1990s and moved me to bring the disparate communities together there for a conference. Organized with the support of Sara Diamond at the Banff New Media Institute, Banff Centre for the Arts, the conference was to feature a rather physically taxing hike to the actual Walcott quarry of the Burgess Shale followed by a meeting of minds for two days of comparative approaches to emergence in biology and in computer systems.

Once in the front offices of Professor Gould’s laboratory I was directed by a graduate student back into the bowels of the fossil collection to find him. Dr. Gould greeted me warmly. I explained my vision for the conference to be held at and about the Burgess Shale. I knew he had travelled there once in the mid 1980s. He explained how he had made it determinedly up the mountain but had collapsed on the way down and had to be airlifted out. He also revealed to me that much as he would have liked to come to our event as a speaker, he was determined to finish his next book, which was he explained going to be possibly his greatest work. That book, The Structure of Evolutionary Theory was published in 2002 only two months after his death from cancer. I sensed an urgency in him that may have been born from his earlier bout with another form of cancer and, he thought, only temporarily beating the odds.

Dr. Gould then said “well, I am not very digital” and pointed to the manual typewriter on a small desk sandwiched between stacks of fossil boxes against a window. He expressed interest in the results of our meeting and asked me to “keep him closely informed” and sent me packing with signed copies of several of his books. The event, which I had named *Digital Burgess* went off without a hitch and was given glowing reviews by many attendees and pointed out to me that the worlds of computer science and Palaeontology were about as far apart as any two fields could be. Computer scientists and programmers are *world makers*, able to draw from a seemingly unlimited store of virtual parts to create what they feel are worlds without limit. On the other hand, Palaeontologists are *detectives*, overwhelmed by data that often tells an incomplete or confusing story of a world that they do not have direct access to: the deep antiquity of early evolution on the planet. This became apparent during a conversation between rainforest ecologist (and creator of the artificial evolution simulation Tierra) Dr. Tom Ray and chief Palaeontologist of the Burgess Shale, Dr. Desmond Collins of the Royal Ontario Museum. Framed by the actual rock face of the Burgess Shale (Figure 2), Dr. Collins explained how the genetic record of organisms of the Burgess Shale suggested that they traced their lineage back to a time much earlier than the 535 million year old fossils behind them.

Figure 2 Two worlds meet, Dr. Tom Ray of the Tierra artificial evolution project (left) and Dr. Desmond Collins, chief Paleontologist for the Burgess Shale, Royal Ontario Museum, at the Burgess Shale, Canada, September 2, 1997 (photo by the author)

Wrapping up this long forward, this thesis is the result of quarter century of my personal quest to employ software, this new “playground of the mind”, to allow the human imagination to gain insight as to how life might have come into existence from nonlife. When the Evolution Grid (or EvoGrid) project was conceived and launched in 2007 it was first broadly and deeply described as an experiment in technogenesis, the use of technology to explore how the Genesis of life on Earth might have come about. It is in this spirit and in homage to those who have wondered and experimented on this theme for centuries past, that this endeavour is undertaken.

A.2 Book Chapter: Nerve Garden

The following book chapter describing work completed in 1997 was published in *Artificial Life Models in Software*, Springer-Verlag, London, 2004. Reprinted by permission of the editors.

Nerve Garden: germinating biological metaphors in net-based virtual worlds

Bruce Damer, Karen Marcelo, Frank Revi, Chris Laurel
Biota.org, Contact Consortium Special Interest Group

Todd Furmanski
University of Southern California

Contact Consortium, P.O. Box 66866
Scotts Valley, CA 95067-6866 USA
1 831 338 9400
www.ccon.org
bdamer@ccon.org

Introduction

Nerve Garden is a biologically inspired multi-user collaborative 3D virtual world available to a wide Internet audience. The project combines a number of methods and technologies, including L-systems, Java, cellular automata, and VRML. Nerve Garden is a work in progress designed to provide a compelling experience of a virtual terrarium which exhibits properties of growth, decay and energy transfer reminiscent of a simple ecosystem. The goals of the Nerve Garden project are to create an on-line 'collaborative A-Life laboratory' which can be extended by a large number of users for purposes of education and research.

1. History and Background of the Project

1.1 Artificial Life meets the World Wide Web

During the summer of 1994, one of us (Damer) paid a visit to the Santa Fe Institute for discussions with Chris Langton and his student team working on the Swarm project. Two fortuitous things were happening during that visit, SFI was installing the first Mosaic Web browsers, and digital movies of Karl Sims' evolving "evolving virtual creatures" (Sims, 1994) were being viewed through the Web by amazed students (see figure 1 and view on the Internet in the reference section at Sims, 1997). It was postulated then that the combination of the emerging backbone of the Internet, a distributed simulation environment like Swarm and the compelling 3D visuals and underlying techniques of Sims' creatures could be combined to produce something very compelling: on-line virtual worlds in which thousands of users could collaboratively experiment with biological paradigms.

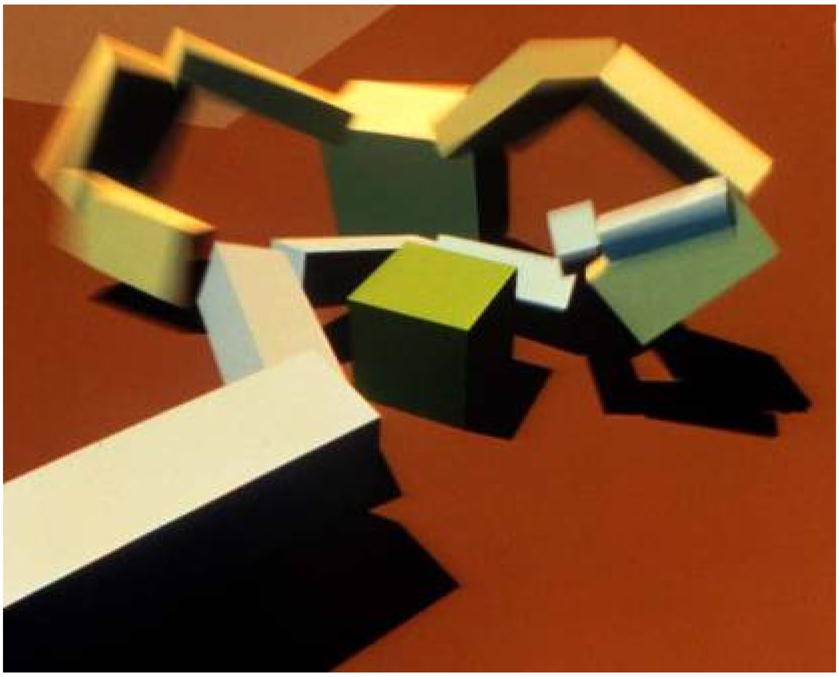


Figure 1: View of Karl Sims' original evolving block creatures in competition

One of the Contact Consortium's special interest groups, called *Biota.org – The Digital Biology Project*, was chartered in mid 1996 to develop virtual worlds using techniques from the Artificial Life (ALife) field. Its first effort was Nerve Garden experienced as an art installation at the SIGGRAPH 97 conference and made available on-line starting in August of 1997. Several hundred visitors to the SIGGRAPH "Electric Garden" Nerve Garden installation used L-systems and Java to germinate plant models into shared VRML (Virtual Reality Modeling Language) island worlds hosted on the Internet. Biota.org is now seeking support to develop a subsequent version of Nerve Garden, which will embody more biological paradigms, and, we hope, create an environment capable of supporting education, research, and cross-pollination between traditional Artificial Life (ALife) subject areas and other fields.

1.2 Background: L-Systems

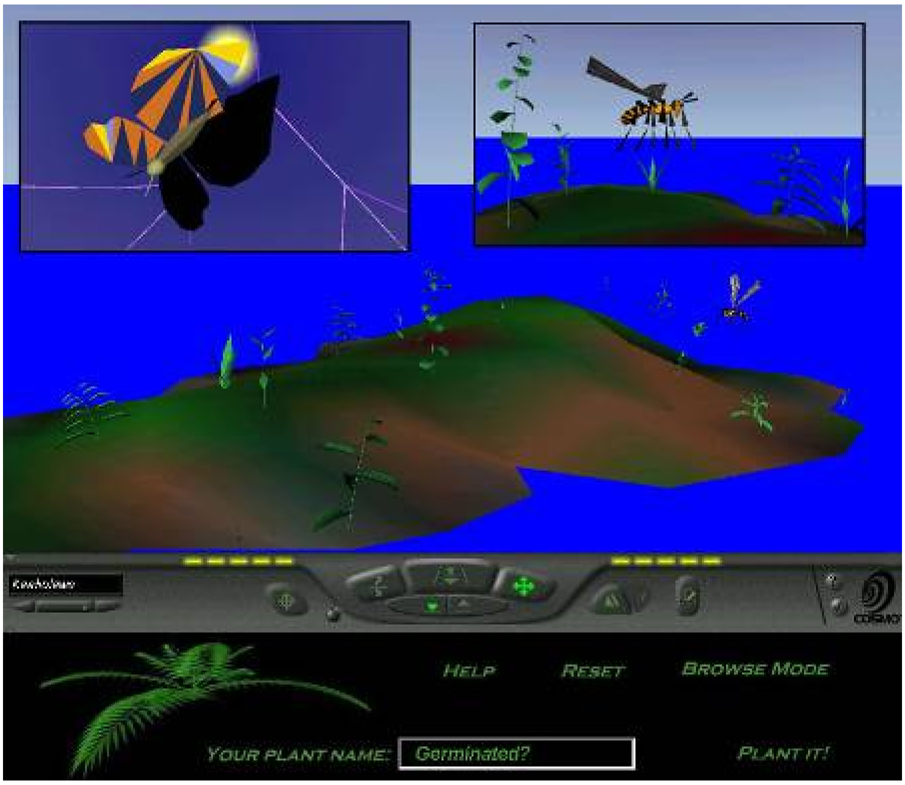
L-systems (Prusinkiewicz & Lindenmayer, 1990) have become a commonly used tool for many computer applications. Commercial 3D packages like Worldbuilder utilize L-systems to model and simulate vegetation (Worldbuilder, 2004). Instead of hand modeling potentially thousands of trees, procedural generation offers a large amount of data compression, and an incredible amount of variance. No two trees in a forest may look alike, but each could be identified as a pine or oak.

While L-systems have classically been used to describe plants, there have been several cases in which the grammars and implementations have been used for other ends. Karl Sims' own virtual creatures used L-system like branching structures. Limbs and sub limbs, much like arms and fingers on a human, determined the basic structure of the evolved animals. One program, LMUSe, converts L-system strings into MIDI format, transforming the systems into musical compositions (Sharp, 2003). Instead of moving in world space and drawing to the screen, the program interprets the grammar as cues to change pitch or transpose. Famous patterns like Koch's Snowflake can not only be seen but also heard.

L-systems have proven useful in modeling virtual cities (Parish, 2004). Tasks from generating street layouts to house and building appearances have been accomplished using L-systems in one way or another. The advantages of compression and levels of detail apply just as well in a "built" environment as a "grown" one. Buildings can show similarities, but nevertheless possess enough variance to avoid unrealistic repetition. Architectural "styles" offer an analog to biological "species" in this sense. The cities themselves can be seeded like forests, and expand over time, implying a complex history of growth and development. Local and global factors can be incorporated into such growth, further adding to the complexity and believability of the city.

The ability to generate complex geometries from simple rules means that, like Conway’s ”Game Of Life” (Gardner, 1970), L-Systems can be manipulated with a few simple parameters and permit children and adults alike to explore forms that with ordinary artistic abilities, they would not be able to express. The motivation for Nerve Garden was to permit ordinary users of the Internet to engage in this exploration using the familiar metaphors of landscapes featuring a range of L-system derived plant forms.

2. Nerve Garden I: Inspiration, Architecture and Experience



YOUR PLANT NAME: Germinated?

Figure 2: Nerve Garden interface in web browser

2.1 Inspiration

Nerve Garden I (interface shown in figure 2 above) is a biologically-inspired shared state 3D virtual world available to a wide audience through standard Internet protocols running on all major hardware platforms. Nerve Garden was inspired by the original work on ALife by Chris Langton (Langton 1992), the digital ecosystem called Tierra by Tom Ray (Ray 1994a), the evolving 3D virtual creatures of Karl Sims (Sims 1994), and the Telegarden developed at the University of Southern California (Goldberg, 1995). Nerve Garden sources its models from the work on L-systems by Aristide Lindenmayer, Przemyslaw Prusinkiewicz and Radomir Mech (Prusinkiewicz & Lindenmayer 1992) (Mech & Prusinkiewicz, 1996).

2.2 Architectural Elements

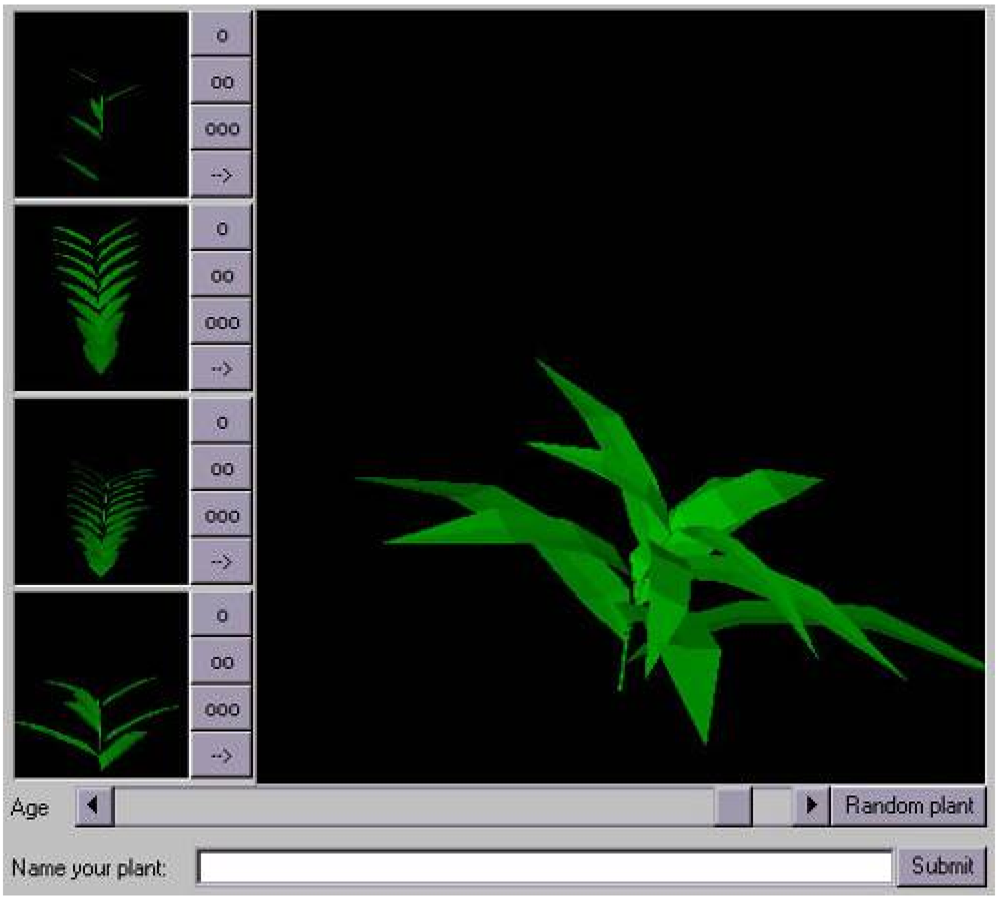


Figure 3: Lace Germinator Java client interface

Nerve Garden I allowed users to operate a Java-based thin client, the Germinator (see figure 3 above), to visually extrude 3D plant models generated from L-systems. The 3D interface in the Java client provided an immediate 3D experience of various L-system plant and even some arthropod forms (see figure 4 below). Users employed a slider bar to extrude the models in real time and a mutator to randomize select production rules in the L-systems and generate variants on the plant models. After germinating several plants, the user would select one, name it and submit it into to a common VRML97 scenegraph called the Seeder Garden.

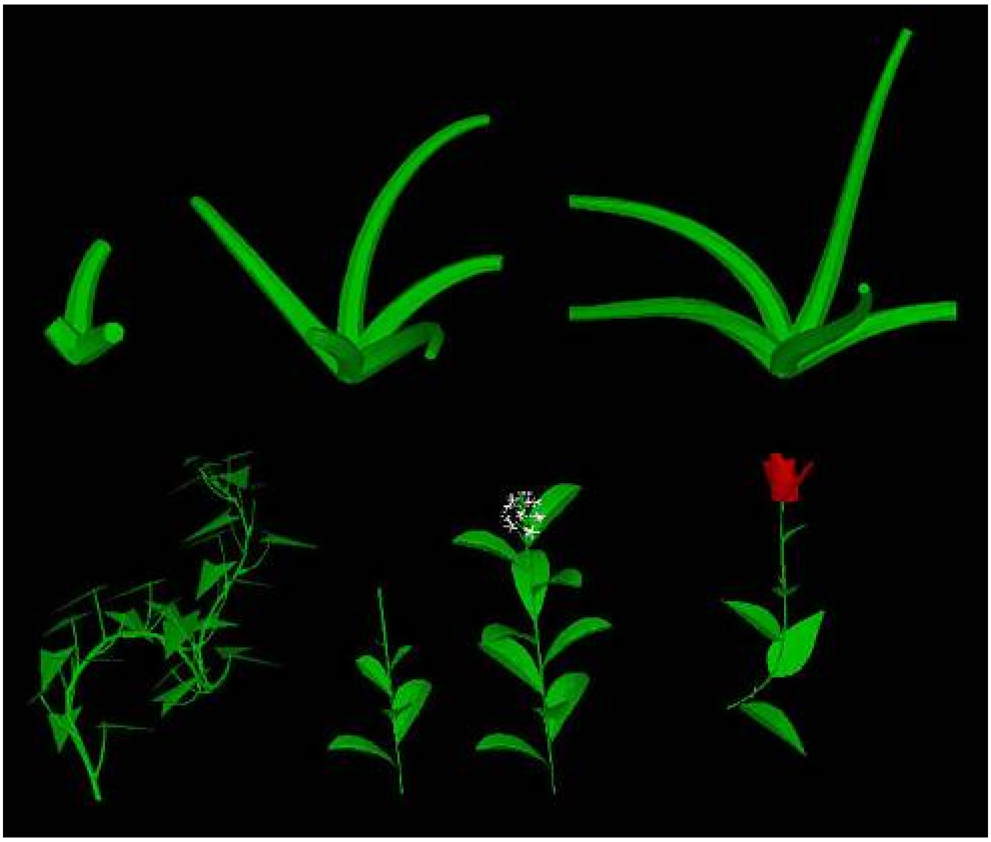


Figure 4: Lace Germinator Java client interface

The object passed to the Seeder Garden contained the VRML export from the Germinator, the plant name and other data. Another Java application, called NerveServer, received this object and determined a free 'plot' on an island model in a VRML scenegraph (shown in figure 2 above). Each island had a set number of plots and showed the user where his or her plant was assigned by a red sphere operated through the VRML external authoring interface (EAI). Cybergardeners would open the Seeder Garden window where they would then move the indicator sphere with their plant attached and place it into the scene. Various scenegraph viewpoints were available to users, including a moving viewpoint on the back of an animated model of a flying insect endlessly touring the island (the bee and butterfly shown in figure 2). Users would often spot their plant as the bee or butterfly made a close approach over the island. Over 10MB of sound, some of it also generated algorithmically, emanated from different objects on the island added to the immersion of the experience. For added effect, L-system based fractal VRML lightening (with generated thunder) occasionally streaked across the sky above the Seeder Garden islands.

NerveServer permitted multiple users to update and view the same island. In addition, users could navigate the same space using standard VRML plug-ins to Web browsers on SGI workstations, PCs or Macintosh computers from various parts of the Internet. One problem was that the distributed L-system clients could easily generate scenes with several hundred thousand polygons, rendering them impossible to visit. We used 3D hardware acceleration, including an SGI Onyx II Infinite Reality system and a PC running a 3D Labs Permedia video acceleration card to permit a more complex environment to be experienced by users. In the year 2004 and beyond, a whole new generation of 3D chip sets on 32 and 64 bit platforms will enable highly complex 3D interactive environments. There is an interesting parallel here to Ray's work on Tierra, where the energy of the system was proportional to the power of the CPU serving the virtual machine inhabited by Tierran organisms. In many Artificial Life systems, it is not important to have a compelling 3D interface. The benefits to providing one for Nerve Garden are that it encouraged participation and experimentation from a wide group of users.

2.3 Experience: what was learned

As a complex set of parts including a Java client, simple object distribution system, a multi-user server, a rudimentary database and a shared, persistent VRML scenegraph, Nerve Garden functioned well under the pressures of a diverse range of users on multiple hardware platforms. Users were able to use the Germinator applet without our assistance to generate fairly complex, unique, and aesthetically pleasing models. Users were all familiar with the metaphor of gardens and many were eager to 'visit their plant' again from their home computers. Placing their plants in the VRML Seeder Gardens was more challenging due to the difficulty of navigating in 3D using VRML browsers. Younger users tended to be much more adept at using the 3D environment. A photo of a user of the Nerve Garden installation at the Electric Garden emerging technologies pavilion at SIGGRAPH 1997 in Los Angeles is featured in figure 4 below.

Figure 4: User at SIGGRAPH Nerve Garden Installation, August 1997

While it was a successful user experience of a generative environment, Nerve Garden I lacked the sophistication of a 'true ALife system' like Tierra (Ray 1994a) in that plant model objects did not reproduce or communicate between virtual machines containing other gardens. In addition, unlike an adaptive L-system space such as the one described in (Mech & Prusinkiewicz, 1996), the plant models did not interact with their neighbors or the environment. Lastly, there was no concept of autonomous, self replicating objects within the environment. Nerve Garden II will address some of these shortcomings, and, we hope, contribute a powerful tool for education and research in the ALife community.

Did Nerve Garden attain some of the goals we set for presenting an ALife-inspired virtual world? The environment did provide a compelling space to draw attention while also proving that an abstraction of a world, that of a virtual forest of L-systems, could be transmitted in algorithmic form and then generated on the client computer, achieving great compression and efficiency. When combined with streaming and ecosystem controls, Nerve Garden II could evolve into a powerful virtual world architecture testbed.

Visiting Nerve Garden I

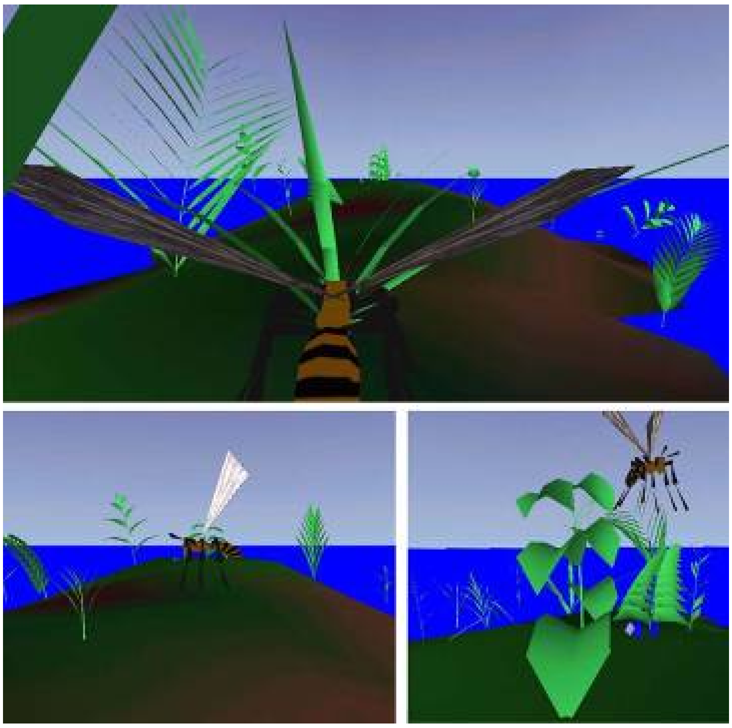


Figure 5: Bee flight through a Nerve Garden island populated by user-generated L-System plants

Nerve Garden I can be visited using a suitable VRML97 compatible browser running Java 1.1. Scenes like the ones in figure 5 above can be experienced in real-time rendered virtual islands that may be toured through the traveling "bee" viewpoint. All of the islands and L-Systems made at SIGGRAPH 97 can be viewed on the web at the references below. The Biota special interest group and its annual conferences are covered at http://www.biota.org.

3. A Next Evolutionary Step: Nerve Garden II

The Biota special interest group is seeking support for a subsequent version of Nerve Garden. Our goals for Nerve Garden II are:

- to develop a simple functioning ecosystem within the VRML scenegraph to control polygon growth and evolve elements of the world through time as partially described in (Mech & Prusinkiewicz, 1996);
- to integrate with a stronger database to permit garden cloning and inter-garden communication permitting cross pollination between islands;
- to embody a cellular automata engine which will support autonomous growth and replication of plant models and introduce a class of virtual herbivores ('polyvores') which prey on the plants' polygonal energy stores;

- to stream world geometry through the transmission of generative algorithms (such as the L-systems) rather than geometry, achieving great compression, efficient use of bandwidth and control of polygon explosion and scene evolution on the client side;

Much of the above depends on the availability of a comprehensive scenegraph and behavior control mechanism. In development over the past several years, Nerves™ is a simple but high performance general purpose cellular automata engine written as both a C++ and Java kernel. Nerves is modeled on the biological processes seen in animal nervous systems, and plant and animal circulatory systems, vastly simplified into a token passing and storage mechanism. Nerves and its associated language, NerveScript, allows users to define a large number of pathways and collection pools supporting flows of arbitrary tokens, token storage, token correlation, and filtering. Nerves borrows many concepts from neural networks and directed graphs used in concert with genetic and generative algorithms as reported by Ray, Sims (Ray 1994b, Sims 1994) and others.

Nerves components will underlie the Seeder Gardens providing functions analogous to a drip irrigation system, defining a finite and therefore regulatory resource from which the plant models must draw for continued growth. In addition, Nerves control paths will be generated as L-system models extrude, providing wiring paths connected to the geometry and proximity sensors in the model. This will permit interaction with the plant models. When pruning of plant geometry occurs or growth stimulus becomes scarce, the transformation of the plant models can be triggered. One step beyond this will be the introduction of autonomous entities into the gardens, which we term 'polyvores', that will seek to convert the 'energy' represented by the polygons in the plant models, into reproductive capacity. Polyvores will provide another source of regulation in this simple ecosystem. Gardens will maintain their interactive capacity, allowing users to enter, germinate plants, introduce polyvores, and prune plants or cull polyvores. Gardens will also run as automatous systems, maintaining polygon complexity within boundaries that allow users to enter the environment.

```
spinalTap.nrv

      DEF spinalCordSeg Bundle {
      -spinalTapA-Swim-bodyMotion[4]-Complex;
      -spinalTapB-Swim-bodyMotion[4]-Complex;
}
```

Figure 6: Sample NerveScript coding language

We expect to use Nerves to tie much of the above processes together. Like VRML, Nerves is described by a set of public domain APIs and a published language, NerveScript (Damer, 1996). Figure 6 lists some typical NerveScript statements which describe a two chain neural pathway that might be used as a spinal chord of a simple swimming fish. DEF defines a reusable object spinalCordSeg consisting of input paths spinalTapA and spinalTapB which will only pass the token Swim into a four stage filter called bodyMotion. All generated tokens end up in Complex, another Nerve bundle, defined elsewhere.

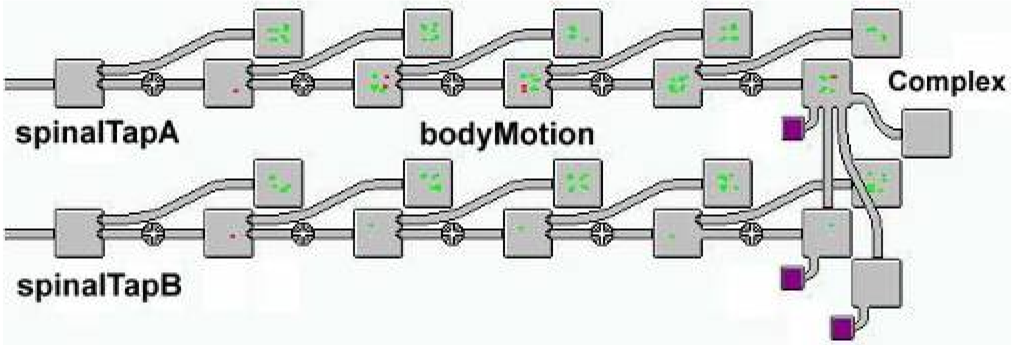


Figure 7: Nerves visualizer running within the NerveScript development environment

Figure 7 shows the visualization of the running NerveScript code in the NerveScript development environment. In the VRML setting, pathways spinalTapA and B are fed by eventOut messages drawn out of the scenegraph while the Nerve bundles generate eventIns back to VRML using the EAI. Nerves is fully described at the web address referenced at the end of this paper.

4. The Role of ALife in Virtual Worlds on the Internet

4.1 Multi-user on-line worlds: a rich space for biological metaphors

Multi-user "avatar" enabled Internet-based virtual worlds have evolved from relatively simple environments in the mid 1990s to multi-million dollar massively multiplayer online role playing games and simulations today (Damer, 1997). There is a large commercial and research driven motivation to create richer environments to attract and keep users of these on-line spaces. Techniques from the artificial life field, such as L-Systems, have become increasingly employed in online virtual worlds in the following roles:

- To provide biologically inspired behaviors, including animated behaviors, growth and decay of the environment, and generation and mutation of non-player characters to draw users into these spaces, for purposes of entertainment or learning about the living world.
- To power underlying architectures with biological metaphors.

4.2 Using ALife to draw attention span

The commercial success of non networked CD-ROM games such as 'Creatures' from Cyberlife of Cambridge, UK, Petz from P.F. Magic of San Francisco and the ubiquitous Tomogatchi of Japan have been successful in capturing the human imagination, attention span and the pocket book. For networked gaming in environments such as EverQuest™ The Sims™, AmericasArmy, Neverwinter's Night ™, Second Life™ and Star Wars Galaxies™, the drive for more lifelike animation, better non-player characters and more rich and changeable worlds inspires innovative efforts within many projects. The third Biota conference held at San Jose State University in 1999 (see Biota references below) focused on the application of ALife to this new world.

4.3 Artificial life techniques powering better virtual world architectures

Players soon tire of key-framed repeatable behavior sequences and yearn for objects that seem to learn their moves through stimuli from the human players. Believable physics, non-canned motion, stimulus and response learning drive developers to borrow from biology. Pets and gardens, perhaps our most intimate biological companions in the physical world, would serve to improve the quality of life in the virtual fold.

The key to delivery of better experiences to a variety of user platforms on low bandwidth connections is to understand that the visual representation of a world and its underlying coding need to be separated. This separation is a fundamental principle of living forms: the abstract coding, the DNA is vastly different than the resulting body. This phenotype/genotype separation also has another powerful property: compression. The VRML 3D scenegraph language simply defined a file format, a phenotype, which would be delivered to a variety of different client computers (akin to ecosystems) without any consideration of scaling, or adapting to the capabilities of those computers. A biologically inspired virtual world would more effectively package itself in some abstract representation, travel highly compressed along the relatively thin pipes of the Internet, and then generate itself to a complexity appropriate to the compute space in which it finds itself.

As the virtual environment unfolds from its abstraction, it can generate useful controls, or lines of communication, which allow it to talk to processes back on servers or to peers on the network. These lines of control can also create new interfaces to the user, providing unique behaviors. One might imagine users plucking fruit from virtual vines only to have those vines grow new runners with fruit in different places. With non-generative, or totally phenotypic models, such interaction would be difficult if not impossible. As we saw from the example of Nerve Garden earlier in this chapter, important scenegraph management techniques such as polygon reduction or level of detail and level of behavior scaling could also be accomplished by the introduction of ecosystem-styled metaphors. If we define the energy state of a virtual world inversely to the computing resources it is consuming, as in a natural habitat, it would be inevitable for any scenegraph or objects in it to evolve more efficient representations.

5. Other Examples of L-System-Based Virtual World Construction and Considerations for the Future Use of L-Systems

Chojo, depicted in figure 8 below, is a current mobile project developed by the Integrated Media Systems Center and the Cinema Television's Interactive Media Department at USC. Chojo makes use of emergent L-system rules, but uses the movements of human participants in the physical world as a primary generative force (Chojo, 2004). Tracking users through GPS, Chojo maps movements, path intersections, and user defined “traits” and uses these data to generate evolving shapes in a virtual space. A point in this virtual space can be viewed from a corresponding physical space…a viewer in front of undergraduate library might see a series of vine and crystal like structures covering the building through their PDA.

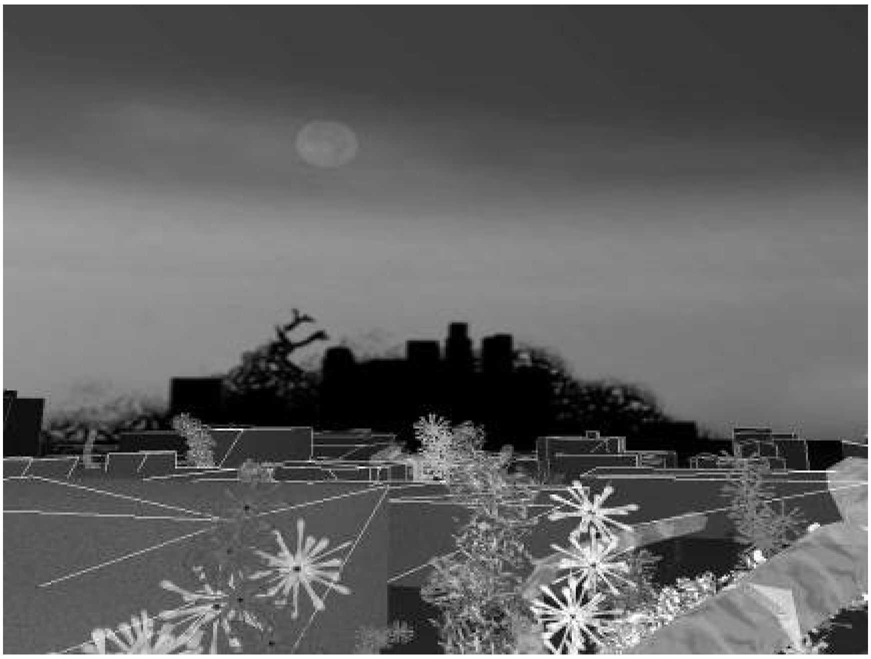


Figure 8: Visual output from USC's Chojo

Exterior forces can continue to enhance a-life systems. Tropism, for instance, can alter a branching pattern globally (Hart, 2003). Forces like wind and gravity change the way a tree grows, for instance. Flowers and leaves move to seek sunlight. L-systems can accommodate such external forces, adding a further lifelike quality. Tropism could also be used in a more abstract sense, depending on the context of the L-system. For instance, variables like population density could be integrated into an algorithm describing a city, or various goals and wants of a virtual creature could ripple through its physical structure.

The recursive and parametric nature of L-systems and other emergent algorithms means that a computer and handle and display varying degrees of resolution and detail. Networked applications like Nerve Garden must take into account computers of varying speeds and abilities. The ability to easily generate variable complexity from a fairly simple set of equations or library of shapes means that a world generated through these emergent methods can be as simple or complex as the machine allows.

We hope that the scope of projects like Nerve Garden will continue to expand not just in size but in relationships. In the physical world terrain affects how plants grow in a given area, but the terrain itself can change because of the presence of plants: A hillside without trees will be susceptible to landslides and erode from the wind. Animals migrate when their food supply dwindles, either due to season or overpopulation.

Much of the emergent and complex nature of artificial and real life arises from the interaction of fairly simple rules. The algorithmic principles underlying this complexity are often hard to divine in nature, yet casting biologically-suggestive rule-bases (such as L-Systems) in software and observing the results can prove challenging, entertaining, and informative.

Acknowledgements

In addition to the extensive contributions made by the authors of this paper, we would like to thank the following sponsors: Intervista, Intel, Silicon Graphics/Cosmo Software, ACM SIGGRAPH, and 3D Labs. Special thanks go to Przemyslaw Prusinkiewicz and numerous other individuals who have worked on aspects of the project since 1996.

References

Chojo, University of Southern California. 7 October 2004 available on the web at: http://interactive.usc.edu/chojo

Damer, B., 1997, Avatars, Exploring and Building Virtual Worlds on the Internet, Berkeley: Peachpit Press.

Gardner, Martin, The fantastic combinations of John Conway's new solitaire game "life",
Scientific American 223 (October 1970): 120-123.

Goldberg, K., Mascha, M., Gentner, S., Rothenberg, N., Sutter, C., and Wiegley, J. Desktop tele-operation via the world wide web. In Proceedings of the IEEE International Conference on Robotics and Automation, 1995.

Hart, John C. "Procedural Synthesis of Geometry" Texturing and modeling: A Procedural Approach Ebert, David S. et al. Amsterdam: Morgan Kaufman Publishers, 2003.

Kelly, Kevin, 1994, Out of Control: The New Biology of Machines, Social Systems and the Economic World, Perseus Press.

Langton, C. 1992. Life at the Edge of Chaos. Artificial Life II 41-91. Redwood City CA: Addison-Wesley.

Mech, R., and Prusinkiewicz, P. 1994. Visual Models of Plants Interacting with Their Environment. In Proceedings of SIGGRAPH 96 . In Computer Graphics Proceedings, 397-410. ACM Publications.

Parish, Y. I. H. and Müller P., "Procedural Modeling of Cities" SIGGRAPH 2001 Conference Proceedings, SIGGRAPH 2001 (Los Angeles, California, USA, August 12-17, 2001), pp. 301-308, 2001. Available on the web at: http://graphics.ethz.ch/Downloads/Publications/Papers/2001/p_Par01.pdf

Prusinkiewicz, P., and Lindenmayer, A., eds. 1990. The Algorithmic Beauty of Plants. New York: Springer Verlag.

Ray, T. S. 1994a. Netlife - Creating a jungle on the Internet: Nonlocated online digital territories, incorporations and the matrix. Knowbotic Research 3/94.

Ray, T. S. 1994b. Neural Networks, Genetic Algorithms and Artificial Life: Adaptive Computation. In Proceedings of the 1994 ALife, Genetic Algorithm and Neural Networks Seminar, 1-14. Institute of Systems, Control and Information Engineers.

Sharp, David. "LMUSe: L Systems to Music" 6 October 2003 available on the web at:
http://www.geocities.com/Athens/Academy/8764/lmuse/lmuse.html

Sims, K., "Evolving Virtual Creatures," Computer Graphics (Siggraph '94) Annual Conference Proceedings, July 1994, pp.43-50.

Sims, Karl, 1997, Karl Sims Retrospective at Biota.org on the Web at:
http://www.biota.org/conf97/ksims.html

WorldBuilder 4. Digital Element. 6 October 2004 available on the web at:
http://www.digi-element.com/site/index.htm

Online Resources

Nerve Garden with SIGGRAPH 1997 samples is available on the web at:
http://www.biota.org/nervegarden/

Damer, B. 1996, Nerves language definition and examples on the web at:
http://www.digitalspace.com/nerves/

Biota Special Interest Group of the Contact Consortium is described on the web at:
http://www.biota.org

The Contact Consortium is described on the web at:
http://www.ccon.org

The USC Telegarden is documented on the web at:
http://www.usc.edu/dept/garden/

Recent publications on virtual worlds are available on the web at:
http://www.digitalspace.com/papers/

A.3 Book Chapter: The God Detector

This book chapter was published in 2008 in *Divine Action and Natural Selection: Science, Faith and Evolution*, Eds.: R. Gordon & J. Seckbach. Singapore, World Scientific: 67-82. Reprinted by permission of the editors.

The God Detector
A Thought Experiment

Disclaimer

I am a technologist, and in this piece I shall approach the key questions of this book as a programmer and historian of technology. In my treatment I will not consider influences of the Divine in our lives in terms of matters of the heart, faith, hope, or the rest of the human cultural milieu. I will simply take on the claim made by some that God plays an active ongoing role in the mechanics of the universe and in the evolution of life. To me this seems like a question best approached from an engineer's frame of reference. A good starting point is to consider the lessons learned and the questions raised by those of us engaged in the new field of "artificial life".

The Artificial Life Programmer, the New Alchemist?

Like medieval alchemists before them, programmers developing artificial life software (often shortened to "A-life") are drawn to the elusive yet seductive proposition that they have the power to animate inanimate matter (Farmer & d'a Belin 1991). In this modern reincarnation of alchemy the inanimate medium is a microscopic substrate of billions of transistors. Popular media from science fiction to Hollywood often depicts A-life as computer viruses and self-reproducing robotics running amok. This means that A-life practitioners (in academia or the hobbyist community) attract quite a bit of press, much of it sensational. As a result, in these reports we are rarely treated to the subtle issues and challenges faced by coders of biologically-inspired virtual worlds.

Another key point is that there is often confusion between the fields of artificial life and artificial intelligence (AI). A-life developers agree that theirs is a "bottom up" approach wherein they simulate a large number of interacting components employing relatively simple rules from which complex behaviors of a whole system emerge (Langton 1991). AI on the other hand tackles the ever receding goal of creating a "conscious" entity with which we would one day be able to communicate. The apocryphal moment of the coming of walking, talking machine intelligence is sometimes referred to by pop-culture practitioners as "the singularity" (Kurzweil 2005). To complicate matters further, developers of A-life software cannot even agree on what defines an "authentic" A-life implementation.

Still, out of all of this confusion emerge some insights we could apply to the Intelligent Design/Creationism vs. Evolution/Science discussion. But before we can draw a hasty conclusion as to whether an artificial life programmer is acting as an "artificial god" (Adams 1998) and "intelligent designer" of his or her own authentic little virtual universe we have to understand the two diametric poles of the A-life continuum.

Two Kinds of God in the A-life Universe

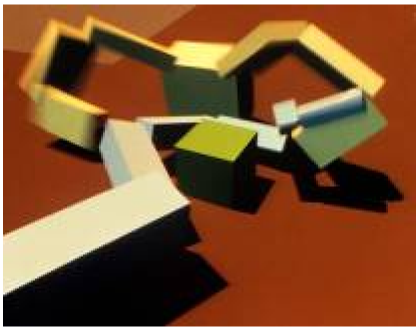


Figure 1: Karl Sims' Evolving Virtual Creatures (1994).

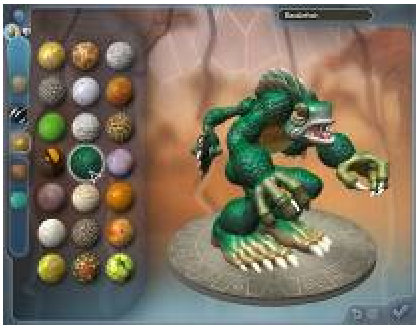


Figure 2: Will Wright's game Spore (2007).

Perhaps the best way to classify A-Life software is to look at two ends of a continuum represented on the one hand by Karl Sims' Evolving Virtual Creatures (Figure 1) and on the other by Will Wright's game Spore (Figure 2). Karl Sims' creatures started life as a simple pair of hinged blocks in a virtual universe that simulated basic physical properties such as fluid, collisions, gravity, and surface friction (Sims 1994). From that point on the simulation was allowed to continue on its own without human intervention. The creatures would perform simple tasks such as swimming or walking, or competing with other creatures for control of a block of "food". The best performers were allowed (by the system software) to reproduce. Random mutations were introduced automatically into the "genome" of the creatures between generations, affecting the external body shapes or internal control functions. In this completely "hands off" A-life system the virtual creatures "evolved" many of the same mobility strategies found in nature (swimming with four paddles like a turtle, slithering like a snake, or perambulating like a gorilla). All of these behaviors emerged without human programmer intervention.

In contrast, the computer game Spore, which is being developed by Will Wright of the Maxis-Electronic Arts Company, bears only a passing resemblance to an A-life environment. The release of Spore in 2008, will, however, be heralded as an "evolution" or "biological" game and yet most activities are largely directed by the human player and built-in procedures. Players use editor tools to design creatures, landscapes, dwellings and vehicles, guiding virtual creatures who inhabit toy planets to live out virtual lives from primordial soup to the space age. The populations "evolve" through procedural algorithms until the player (or game code itself) again intervenes to keep the action moving forward.

Given this continuum, we posit that there two kinds of God in the A-life universe: the Karl Sims' *God the Mechanic* building the machine that is the whole simulation, setting its initial conditions and then returning only occasionally to view the current state of the simulation; and Will Wright's *God the Tinkerer*, constantly poking and prodding to tweak the mechanisms of virtual creation. Clearly these definitions might also apply to different extremes of *god traditions* found in human cultures.

There are two key kernels of truth that we can winnow from these early decades of A-life alchemy:

> Kernel 1: That all attempts to render life down into its basic elements and then represent it abstractly come down to: a) creating an algorithm for making copies of the blueprints to make yet more algorithms and b) that imperfect copies of these blueprints are sometimes passed on, creating variations and, possibly, advantageous adaptations.
>
> Kernel 2: That after these algorithms run for a while, passing on a great number of blueprints and interacting within some kind of a simulated virtual environment, the whole system reaches a tipping point where, to our perception, it becomes opaque to complete understanding. Thereafter even the A-life developers themselves must assume the role of a biologist, dissecting the genomes of their virtual creatures or examining their "fossil record" looking for clues to what the process of artificial evolution hath wrought.

Lost in the Noise of the Data Explosion

Thus, the observer of the biologically inspired software simulation soon becomes "lost in the noise" (Negroponte 1995), much as a biologist might spend a lifetime to grasp one small aspect of the stupefyingly complex machinery of a single cell.

I propose that this property of *onset opacity* also holds for the world's religious traditions. For each there was an original prophet, and an original set of core stories and concepts (some new, some drawn from prior traditions). Once the copying of these stories got underway, a mutation and adaptation process began. The resulting data explosion of writings, stories, laws, debates, schools, conflicts, extinct lines, and new branches soon obscured many of the original statements attributed to the founding prophets. Religious seekers (and even many serious researchers) are unable or unwilling to apply reductionist methods to prune out later inserted, contradictory or inconsistent yet closely held beliefs or writings. In addition, modern monotheistic religions stand upon foundations of earlier belief systems, most of which carry no written record. Therefore, fundamental questions about God and the universe that might emerge from any religious tradition are likely to remain lost in the largely opaque "tree of noise" of religious histories and discourse. In other words, if at any time God ever made Himself unequivocally visible to a human being and uttered or physically manifested anything about life or the universe, that original direct experience of God's existence has become irretrievably lost. In modern times, no verifiable experience of God's presence in the physical universe that is not explainable by other means has been observed. Therefore, if we cannot validate the original claims, or detect any direct physical influence today, we have to look for evidence of God's Hand at another level.

The God Detector

For some of the other authors of this book, prior writings about God, or personal (but unverifiable) experiences of God is evidence enough of His existence. However, when making a strong claim about God the Intelligent Designer, such empirical evidence is not good enough to make the case. If God is a programmer tweaking the code of the physical observable universe (not just affecting our own independent thoughts) his influence has to be detectable and independently verifiable. To sense the hitherto unseen Hand of God, we hypothesize that is might be possible to employ a *God Detector* which could either be *found* or *built*. We will first take on the challenge of identifying an existing natural God Detector and later on in this chapter, consider building a God Detector using human technology. If you will indulge me, dear reader, I invite you to join me in the following thought experiment surrounding the *quest for the God Detector*.

Finding the God Detector

How to look for signs of God's influence comes down to *where* to look for them, and that comes down to *what* you look at and what you *exclude* looking at within the universe.

For a time, I set down my pen and declared to myself that this was an unsolvable problem. A few days later I was reading a history of the Institute for Advanced Study in Princeton in the USA where I am a currently a visitor. A brilliant account of John von Neumann's digital computer designed and built at IAS in the late 1940s contained an account of an impassioned researcher named N. Barricelli who was developing "numerical symbioorganisms" for this pioneering digital computer (Dyson 1997). I was stunned to realize that on a machine of such tiny capabilities, Barricelli was able to run basic artificial life code thirty five years before the term was coined.

This led me to the following insight: what if the universe could be reduced down at the lowest levels to a programmable machine running algorithms? Several theories of how the universe works at the quantum level propose that this is in fact how things work (Lloyd 2006). I realized that if you can render the universe's operation down to simple algorithms, basic questions could then be asked, and a natural God Detector could be found at a key code location found within one of the universe's algorithms.

God and the Copying Rule

A living organism differs from bare rock, gases or a pool of liquid in one very specific way: the living organism contains instructions that are copied, for the most part unaltered, from one version to the next. In fact the organism *must* copy these instructions or face extinction. Thus, there would be no copying mechanism if previous copying mechanisms ceased to work, so copying mechanisms can and must continue to copy. This is the Copying Rule, and, as we have seen previously, it can also be found at work in human culture, where language permits the telling and retelling of a story, and also within the new medium of digital computers and networks, where programs are copied between computers.

The universe contains a large number of seemingly copied objects, from rocks to stars to galaxies, but the process by which these objects were made did not involve construction from a blueprint, instead their existence is owed to the laws of physics applied to starting conditions. Therefore, as far as we know, all matter and energy in the universe inhabits one of two organizational regimes:

Regime 1 which is governed by…	**Regime 2 which is governed by…**
• Formulaic Laws of Nature • An element of uncertainty, or randomness we might call "R"	• Formulaic Laws of Nature • An element of uncertainty, or randomness we might call "R" • The Copying Rule

As we infer from the above table, along with the Laws of Nature and the Copying Rule, another of the distinct organizing operators of the universe is the element of uncertainty. This could be thought of in terms of unpredictable (i.e. random) effects either from some source in the very small (quantum fluctuations for example) or the very large (the mass overlapping effect of gravitational forces from atoms, stars and galaxies for example). We will take up this operator "R" later as it is the pivot on which this simple thought experiment turns.

The Copying Rule is well understood in molecular biology. For each genotype (information blueprint encoded in a cell's nucleus) a phenotype (a living body or other resulting output) is produced. The Copying Rule as seen operating in human culture is less well understood but clearly occurs. Copy-able cultural objects such as ideas, stories, music or instructions are sometimes referred to as "memes" within a new field called "memetics" (Dawkins, 1976). Clearly, technological objects (where copies of objects are made with reference to instructions) also execute the Copying Rule. As we addressed previously, a sub-specialty of computer software called artificial life attempts to emulate the biological implementation of the Copying Rule by creating software analogues to genotypes and phenotypes. More radical thinkers consider all software, such as ordinary applications like word processors, to also execute the Copying Rule with humans acting as the phenotype (the host) that is the mechanism to enable the copying of these programs (Dyson 1997).

A Simple Model of the Copying Rule

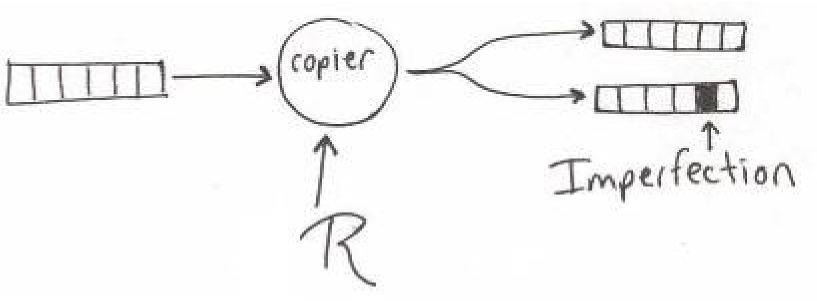


Figure 3: The Copying Rule.

A simple model of the Copying Rule is depicted in Figure 3. An input sequence of information, which could be encoded in molecular material, language or computer code, enters a copying mechanism upon which some random input R may or may not act, and two or more resultant output sequences are produced, some of which may contain random changes. There are variations of this mechanism, one that would take two input sequences and combine them into an output sequence. A Copying Rule could be said to have been "successfully executed" if the output information sequence is not so altered that it could not be used to produce a future copy. A "failed" application of the rule produces a sequence that can never again be copied.

Scope and Time Scales of the Copying Rule

The Copying Rule is the central driving mechanism within biological evolution, cultural evolution and technological evolution and operates across a range of time scales and scopes: from billions of years to kilo-years for molecular evolution to years or days for cultural evolution, and days to milliseconds for evolution in information systems (see table below).

Molecular copying 4 billion to1 kilo-years	**Cultural copying 1 kilo-year to 1 day**	**Digital copying 1 day to 1 millisecond**
• Development of multi-cellular life • Divergence of Galapagos finch populations	• Rise and fall of a great empire (or religion) • Spread of hoax on the Internet	• Spread of virus on the Internet • 1 millisecond of computation in SETI@Home grid

How God the Intelligent Designer Engages the Copying Rule

A "designer" is someone who makes a plan for the future and instructs other people or mechanisms to bring that plan into reality. If God is acting in the universe as an "intelligent designer" and desires to operate in places where there are living things, then He has no choice but to engage the Copying Rule.

God has two obvious ways to interact with Copying Rule:

1) God would engage the Natural Laws that make the copying happen or
2) God would influence the operation of the Copying Rule by engaging the nondeterministic forces we call R, which create the imperfections or mutations in the copying process.

Common sense dictates that God cannot use both of these mechanisms at the same time as they work in opposition. For example, while the laws of gravity cause a feather to fall predictably, the random motions of the air through which the feather travels produce an unpredictable landing place.

By calling God a "designer" it is implied that He an actor upon the forces that shape the universe and is not those forces themselves. A God who is operating solely through deterministic laws is a God with no free-will. These laws affect the universe in pre-ordained ways with predictable outcomes. After the creation of the universe (and these laws) this *God the Mechanic* would simply leave the universe to run on autopilot and thereafter be undetectable.

If God cannot suspend or change the natural laws, then He might operate by introducing imperfections as a *tinkerer* in the mechanics of the Copying Rule shifting the application of the randomizer R to cause accumulated errors in the copying process that would give rise to our world (Figure 4).

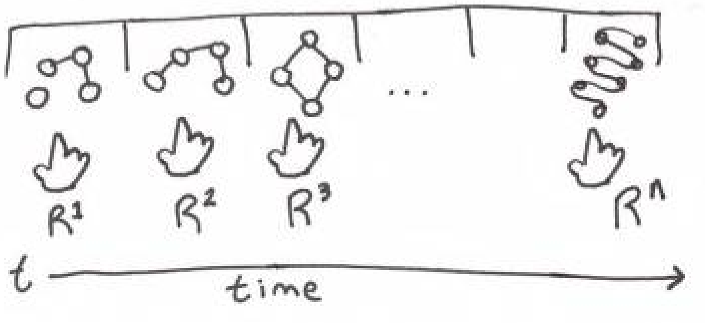


Figure 4: The accumulating effects of R through time.

Perhaps God could decide to permit R to affect the copying mechanism or not, or He could choose to proactively add to or subtract from the influence of R by a large enough amount to "tip the balance" in favor or one copying outcome or the other. In this case the Hand of God should be detectable as localized statistically anomalous behavior in an otherwise uniformly distributed random landscape of R. The monkey wrench in these works is that R itself is by definition unpredictable. If R is governed by some Natural Law or mathematical formula then it would not be R. If God could predict the future value of R and act accordingly then we would have our God the Mechanic back. So God, just like the rest of us, has to live with the unpredictability of R (Figure 5) and would seem to us to operate not by absolute *Will* but by *Whim*. This kind of God would hardly be able to exercise much design upon the universe.

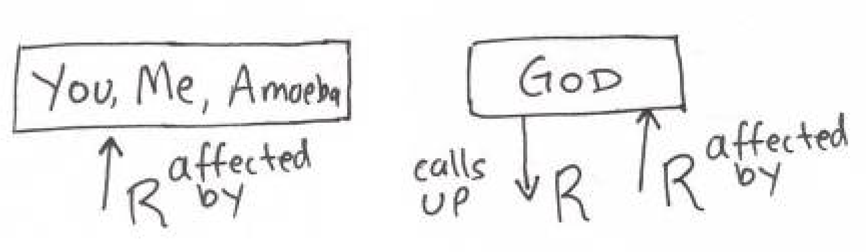


Figure 5: What is affected by R?

The Monk and the Copying Rule

Here is where our thought experiment could use a bit of help from a hypothetical real-world example. Picture a literate monk, working at his table some time in the early years of Christianity. He is given a book written in Hebrew to translate into Greek. In it is a section of a passage:

> "…and Jesus was born to the young girl Mary"

Reaching for his Hebrew-to-Greek dictionary scroll and not finding it, he sighs and makes a guess, translating the phrase to:

> "…and Jesus was born to the virgin Mary"

Perhaps random fluctuations in air molecules contributed to a puff of air that nudged the dictionary scroll from the table, and hence caused the translation of "young girl" to "virgin". Many scholars believe that this translation error actually occurred (Brown 1977, Pagels 2003) and led to the concept to the "virgin birth" or "immaculate conception" in the Catholic tradition. The resulting impact of this was substantial for the future of Christianity, leading to its wider adoption throughout the Mediterranean, where there were existing religious movements that also believed in spiritual power emanating from a virgin birth. The virgin birth idea also led to the suppression of women (whom evidence suggests were treated more equally in the early church) by enforcing male celibacy and sequestering devout and intelligent females away in convents. Male domination of the early church was therefore assured, which eased the integration of the religion into Roman power structures. The supernatural aura of the virgin birth also propelled the character of Jesus along a road that led to his elevation to Godhood following the Council of Nicaea in the fourth century.

Would God have had a hand in this fateful application of R to the translation of early Christian texts? Certainly if it was God's intention as an Intelligent Designer to promote Christianity as a new religious force (at the cost of existing belief systems) then we might say "yes", God influenced the movement of air molecules, at the quantum level, at that critical juncture.

However, God would have to have done more than just arrange for the translation error. God would also have to ensure that the proof-reading senior monk, upon seeing this one term, would not find it erroneous and send it back for correction. In addition, the natural error correcting mechanisms of the followers of the Hebrew version would have to be overcome. In practice, each small change affected through an influence of R (which is by no way guaranteed to work given the unpredictable nature of R) is followed by a virtually uncountable large number of subsequent required adjustments that require almost total foreknowledge of every action. It seems that God's task in influencing history in this way would require a brain that would be large enough to store all possible outcomes while executing perfect adjustments of random effects to guide each step. The size of the required decision-tree for even relatively small scale design changes might exceed the size of the countable particles in the universe. Amazingly, each Monk's brain contains a number of unique pathways through its neurons that already exceed this number. At the finest level of detail, God's brain would have to account for each of these neural pathways and be able to affect the journey of each electron. We are fast approaching an event horizon of total implausibility.

Cultures all over the world attribute "god-like" powers to those who seem to be able to repeatedly "beat the odds" in dice-tossing, in war, in procreation, or in longevity. However, no documented case of the conquest of tremendous odds has ever been produced. Methuselah' 969 year lifespan, and other miracles live strictly in the domain of mythology. It would seem that God is as powerless to affect next toss of the dice as the rest of us.

Many believers might state here that God is a separate, all-knowing, omnipotent actor for whom the universe is a mere toy. In this case then He could choose to be detected or not and we would be powerless to make inquiries about His existence or nature (in which case there is no reason for this book to exist). So let us return to reason and consider God as an actor within the universe subject in some way to its laws, rather than an incalculably large and immeasurable actor separate from everything.

God the Intelligent Adapter

But wait, there is another way to affect the application of R in the Copying Rule, and that is *through adaptation, after the copying is completed.* Every single celled organism in Earth's early seas that suffered an injustice due to physical or chemical fluctuations, heat, cold or an attack had a chance to adapt to the situation and survive to reproduce another day. The machinery of adaptation adjusts for the ravages of R and therefore diminishes and redirects its impact into the future.

So could God in fact be living at "the output end" of the Copying Rule, in the land of adaptation? If so, God's Hand would be busy helping adapt everything from cellular machinery on up to guiding the entire biosphere through the slings and arrows of large scale misfortunes such as meteor impacts.

In human culture, intelligence emerged as a critical adaptation. Might intelligence therefore be a place where the mark of God is the strongest? Would God then not be an Intelligent Designer but instead be a Designer of Intelligence? Would any act of intelligence be an act of God, regardless of the outcome? If God is trying to effect some kind of perfect design upon the universe then influencing outcomes of adaptation might be just as numerically challenging as trying to control individual applications of R. Just as in our monk's brain example, God is again relegated to being an imperfect player, making do with a limited ability to influence adaptations to direct the future of life.

God, Life, the Universe and Everything

So we return to our original question: if God is an actor in the universe and we render the universe down to its simplest organizing principles, then God must have some kind of fundamental relationship with the Copying Rule. We have decided that, for our purposes, we are not considering a God the Mechanic, who simply sets up the initial Laws of Nature and then departs the scene. If our God is actively tinkering then He could only affect the progress of life and culture in two ways: by affecting the unpredictable R value that randomly affects copying machinery, or by working His miracles on the output side of the Copying Rule that adjusts for the influences of R through adaptation.

We concluded that God could not affect any kind of predictive design on the universe by trying to influence the unpredictable effects of R as copying occurs. God's information processing capability would probably have to be many times the size of the universe for even minor adjustments to the future and therefore He could not be an actor in the universe.

This left God only one place to act, as a player in assisting the power of adaptation at the output end of the Copying Rule. Thus, God would not be an Intelligent Designer but instead could be thought of as an *Intelligent Adapter*. If God is indeed operating at the adaptation end of the spectrum, then there is no difference between God's work and the work of evolution through Natural Selection or engineering through human intelligence.

For example, a human technologist using his or her own intelligent genetic engineering skills or the processes of Natural Selection over eons could both create a fish that can live in near-boiling water. To those who did not witness the processes of the engineer or Natural Selection, this fish would be indistinguishable from a miracle from God. Would then believers be forced to conclude that Natural Selection or human genetic engineering must be equivalent to the Hand of God or that God's Hand need not be present at all?

In conclusion, given all the above uncertainties the Copying Rule, when pressed into service as a natural God Detector, is unable to permit us to unambiguously detect any unique sign of the Hand of God.

Where does this leave the believer and the non-believer? Those who still wish to include the presence of a *God the Tinkerer* in the universe could still invoke a vision of God the *Intelligent Adapter*, playing an ongoing (but by no means exclusive or unique) hand in the survival and glorious diversification of life as well as the blossoming richness of human culture and technology. Those who find no need to place an actor like God in the picture can celebrate and seek to better understand the process of billions of years of evolution by cumulative copying and adaptation, made even more astonishing by the very fact that *no hand guided it*. Stuart Kaufmann may show us another way, in which he redefines God "…to mean the vast ceaseless creativity of the… universe" (Kaufmann 2008). If God is embodied in the artful adaptation on the output end of the Copying Rule then He is the agency of the seemingly miraculous processes of Natural Selection and Emergent phenomena.

Afterthought Experiment: Building a God Detector

What if our cumulative technology including computers, networks, robotics, sensors, and Cyberspace, is creating a set of tools which we can use to determine, once and for all, whether God exists? And if so, might we also be able to use these tools to determine God's nature and the exact means by which He manifests in the world? If we as a species could answer the question of the presence of deity in the world it would save untold future strife and focus our intellectual and artistic pursuits like never before.

What if we could "set a trap for God", a place where God could not resist manifesting His Will? What I am proposing is to engage all of the best programmers, artists and philosophers of our generation to create a gigantic network of software and computers, working to create a sort of "Evolution Grid" or "EvoGrid" (Damer 2008). This EvoGrid would start out as *God the Mechanic* (like Karl Sims' creatures) in which we build the simulation, set the initial conditions and then let the artificial ecosystem go from there.

Indeed, such a simulation might satisfy Richard Gordon's challenge in the chapter *Hoyle's Tornado Origin of Artificial Life, A Computer Programming Challenge* found in this volume. The EvoGrid would therefore seek to show that in amongst the vast machinery of the natural laws, and despite the chaos of R, the universe (or God acting within the universe) possesses the innate property to instantiate the Copying Rule and generate us.

However, the EvoGrid could be set up to also embody some of Will Wright's *God the Tinkerer*, with people in the loop. The way this might work is that the creatures of this simulated ecosystem would systematically consume all of human language and culture available to them in the semantic flow of the Internet. Every piece of text, image, music or video, blog, or other cultural artifact would be both the landscape and foodstuffs for the EvoGrid. The creatures of the EvoGrid would continuously adapt to the myriad streams traveling along the growing cyberspace synapses of the collective human mind. The EvoGrid would communicate in its own language which we might be able to understand. If there was ever any medium through which God could speak to us, this would be it. Gerald de Jong claims that artificial life and the EvoGrid might be our way to finely polish a mirror we could then hold up to ourselves (de Jong, 2008). Would we then see the face of God?

Giving Birth to God

In our age old quest to detect and define God, there might be another ultimate outcome in store for us. Over the coming eons, would our own divine creations, such as the EvoGrid, allow us to merge with all living things, and transform and connect all of technological and biological reality? Would we then survive long enough to contact and combine with the EvoGrids of other sentient civilizations? If we never detected God in our own EvoGrid it would no longer matter because in some far distant time all sentient minds, biological bodies, and technological creations would ultimately merge into one total universal life form. If the universe succeeds to birth itself as one conscious entity, everything, including us and all of our past selves, will unify into a single being which we will then call… *God the Universe*.

So perhaps God is nothing more and nothing less than an expression of our hopes and dreams for that distant possibility.

"*God who created all things in the beginning is himself created by all things in the end*" *(Stapledon 1937).*

References

Adams, R. 1998, Is there an Artificial God, in the online proceedings of Digital Biota 2 available on the web at: http://www.biota.org/people/douglasadams/

Barbalet, T. 2007, Biota.org, available on the web at: http://www.biota.org

Brown, R. E. 1977, *The Birth of the Messiah*, Appendix V: The Charge of Illegitimacy, Doubleday & Company.

Damer, B. F. 1997, *Avatars: Exploring and Building Virtual Worlds on the Internet*, PeachPit Press, Berkeley CA.

Damer, B. F. 2008, The EvoGrid, an Evolution Technology Grid, available on the web at: http://www.evogrid.org

Dawkins, R. 1976, *The Selfish Gene*, Oxford University Press.

Dawkins, R. 1986, *The Blind Watchmaker: Why the Evidence of Evolution Reveals a Universe Without Design*, W.W. Norton & Company.

De Jong, G. 2008, from a conversation on the Biota Live Podcast #22, June 20, 2008, *What is the Philosophy of Artificial Life?*, available on the web at: http://www.biota.org/podcast

Dyson, G. 1997, *Darwin Among the Machines: The Evolution of Global Intelligence*, Perseus Books.

Farmer D. J., d'a Belin A. 1991. Artificial Life: The Coming Evolution. In : Langton, C., C. Taylor, J. D. Farmer, & S. Rasmussen [eds], *Artificial Life II, Santa Fe Institute Studies in the Sciences of Complexity*, vol. XI, pp. 815-840, Addison-Wesley, Redwood City, CA.

Gordon, R. 2000, The emergence of emergence: a critique of "Design, observation, surprise!", *Rivista di Biologia /Biology Forum*, 93(2), 349-356.

Wright, W. 2008, From a public conversation with Will Wright, designer of Spore, NASA Ames Research Center, Virtual Worlds Workshop, January 26, 2008.

Kauffman, S. 2008, *Reinventing the Sacred: A New View of Science, Reason, and Religion*, Basic Books.

Kurzweil, R. 2005, *The Singularity Is Near: When Humans Transcend Biology*, Penguin Books.

Langton, C., C. Taylor, J. D. Farmer, & S. Rasmussen [eds], 1991, *Artificial Life II, Santa Fe Institute Studies in the Sciences of Complexity*, vol. XI, pp. 815-840, Addison-Wesley, Redwood City, CA.

Levy, S. 1993, *Artificial Life: The Quest for a New Creation*, Vintage Books.

Lloyd, S. 2006, *Programming the Universe: A Quantum Computer Scientist Takes on the Cosmos*, Alfred Knopf.

Negroponte, N. 1995, *Being Digital*, Alfred Knopf.

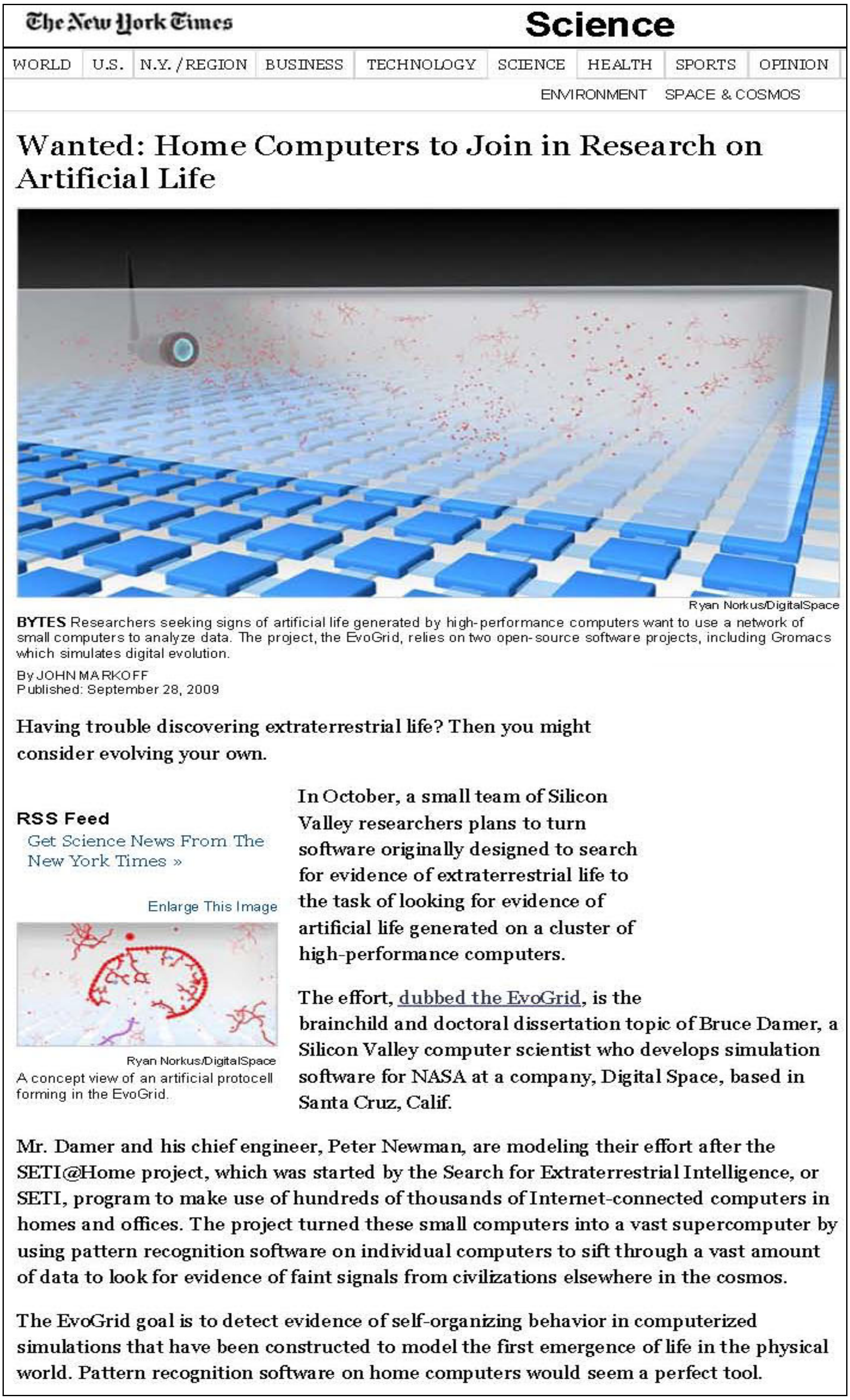
Pagels, E. 2003, *Beyond Belief: The Secret Gospel of Thomas*, Random House.

Sims, K. 1994, Evolving Virtual Creatures, *Computer Graphics SIGGRAPH 1994 Proceedings*, pp. 15-22.

Stapledon, O. 1937, *Star Maker*, Methuen, UK.

A.4 Article on EvoGrid in New York Times, Sept 28, 2009

The following article in the New York Times from September 28, 2009 is reproduced by permission of John Markoff and the New York Times.



Wanted: Home Computers to Join in Research on Artificial Life

Ryan Norkus/DigitalSpace

BYTES Researchers seeking signs of artificial life generated by high-performance computers want to use a network of small computers to analyze data. The project, the EvoGrid, relies on two open-source software projects, including Gromacs which simulates digital evolution.

By JOHN MARKOFF
Published: September 28, 2009

Having trouble discovering extraterrestrial life? Then you might consider evolving your own.

Ryan Norkus/DigitalSpace

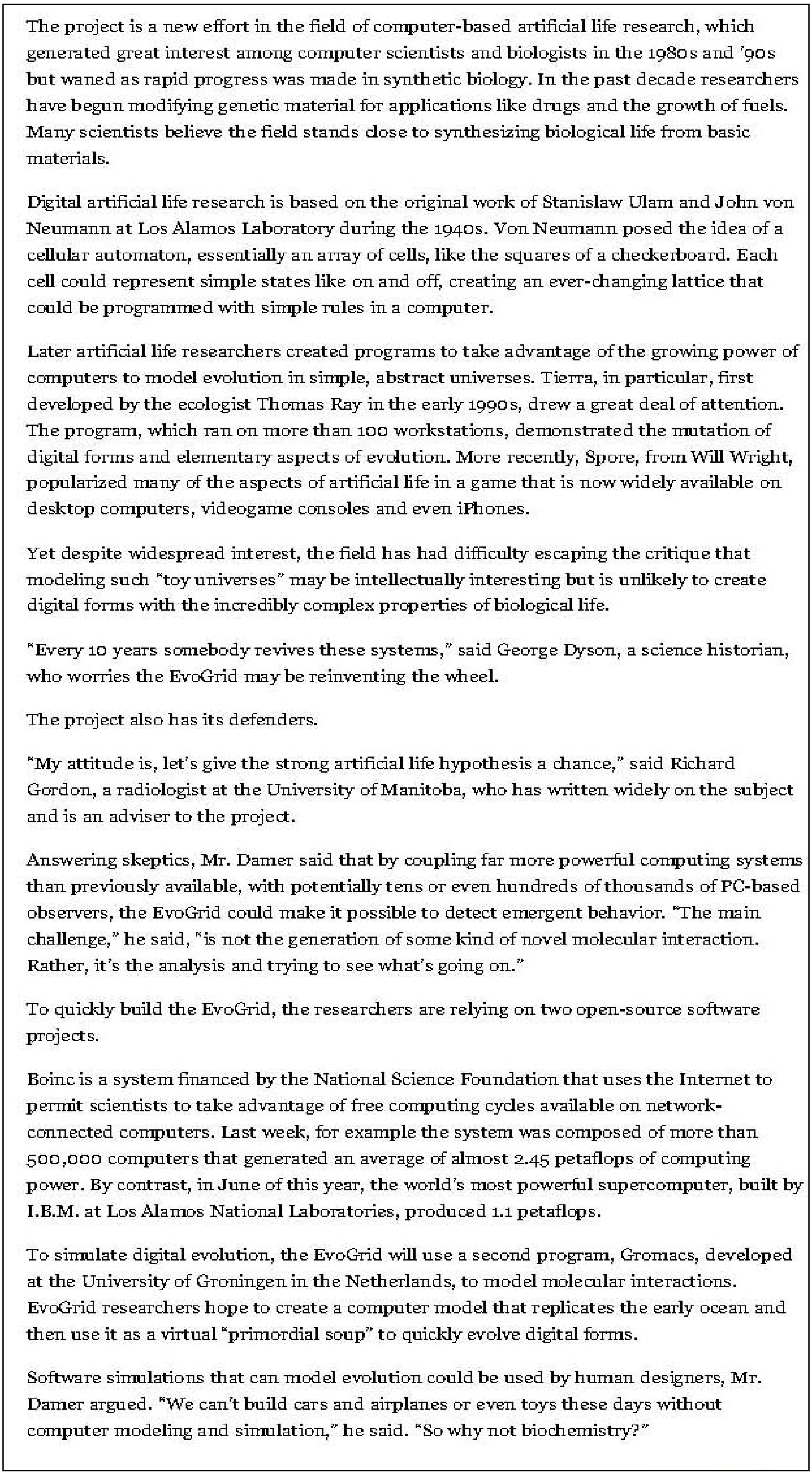
A concept view of an artificial protocell forming in the EvoGrid.

In October, a small team of Silicon Valley researchers plans to turn software originally designed to search for evidence of extraterrestrial life to the task of looking for evidence of artificial life generated on a cluster of high-performance computers.

The effort, dubbed the EvoGrid, is the brainchild and doctoral dissertation topic of Bruce Damer, a Silicon Valley computer scientist who develops simulation software for NASA at a company, Digital Space, based in Santa Cruz, Calif.

Mr. Damer and his chief engineer, Peter Newman, are modeling their effort after the SETI@Home project, which was started by the Search for Extraterrestrial Intelligence, or SETI, program to make use of hundreds of thousands of Internet-connected computers in homes and offices. The project turned these small computers into a vast supercomputer by using pattern recognition software on individual computers to sift through a vast amount of data to look for evidence of faint signals from civilizations elsewhere in the cosmos.

The EvoGrid goal is to detect evidence of self-organizing behavior in computerized simulations that have been constructed to model the first emergence of life in the physical world. Pattern recognition software on home computers would seem a perfect tool.



The project is a new effort in the field of computer-based artificial life research, which generated great interest among computer scientists and biologists in the 1980s and '90s but waned as rapid progress was made in synthetic biology. In the past decade researchers have begun modifying genetic material for applications like drugs and the growth of fuels. Many scientists believe the field stands close to synthesizing biological life from basic materials.

Digital artificial life research is based on the original work of Stanislaw Ulam and John von Neumann at Los Alamos Laboratory during the 1940s. Von Neumann posed the idea of a cellular automaton, essentially an array of cells, like the squares of a checkerboard. Each cell could represent simple states like on and off, creating an ever-changing lattice that could be programmed with simple rules in a computer.

Later artificial life researchers created programs to take advantage of the growing power of computers to model evolution in simple, abstract universes. Tierra, in particular, first developed by the ecologist Thomas Ray in the early 1990s, drew a great deal of attention. The program, which ran on more than 100 workstations, demonstrated the mutation of digital forms and elementary aspects of evolution. More recently, Spore, from Will Wright, popularized many of the aspects of artificial life in a game that is now widely available on desktop computers, videogame consoles and even iPhones.

Yet despite widespread interest, the field has had difficulty escaping the critique that modeling such "toy universes" may be intellectually interesting but is unlikely to create digital forms with the incredibly complex properties of biological life.

"Every 10 years somebody revives these systems," said George Dyson, a science historian, who worries the EvoGrid may be reinventing the wheel.

The project also has its defenders.

"My attitude is, let's give the strong artificial life hypothesis a chance," said Richard Gordon, a radiologist at the University of Manitoba, who has written widely on the subject and is an adviser to the project.

Answering skeptics, Mr. Damer said that by coupling far more powerful computing systems than previously available, with potentially tens or even hundreds of thousands of PC-based observers, the EvoGrid could make it possible to detect emergent behavior. "The main challenge," he said, "is not the generation of some kind of novel molecular interaction. Rather, it's the analysis and trying to see what's going on."

To quickly build the EvoGrid, the researchers are relying on two open-source software projects.

Boinc is a system financed by the National Science Foundation that uses the Internet to permit scientists to take advantage of free computing cycles available on network-connected computers. Last week, for example the system was composed of more than 500,000 computers that generated an average of almost 2.45 petaflops of computing power. By contrast, in June of this year, the world's most powerful supercomputer, built by I.B.M. at Los Alamos National Laboratories, produced 1.1 petaflops.

To simulate digital evolution, the EvoGrid will use a second program, Gromacs, developed at the University of Groningen in the Netherlands, to model molecular interactions. EvoGrid researchers hope to create a computer model that replicates the early ocean and then use it as a virtual "primordial soup" to quickly evolve digital forms.

Software simulations that can model evolution could be used by human designers, Mr. Damer argued. "We can't build cars and airplanes or even toys these days without computer modeling and simulation," he said. "So why not biochemistry?"

Appendix B. Records from the Origin of the EvoGrid Idea

B.1 Summary of Meeting with Richard Dawkins, Jul 10, 2000

Present: Bruce Damer, Stuart Gold, Richard Dawkins, Oxford, U.K.

Report of this meeting presented to members of the Biota.org SIG of the Contact Consortium and others.

I am excited and pleased to tell you that we had a very positive meeting with Richard Dawkins at his home in Oxford today. In short, he has agreed to serve to support our cause, from being part of the advisory board, to recommending other advisory group members (ie: big names), providing leads on possible funders, guidance on criteria and format, and generally being enthusiastic. He even presented a viable method to carry on the annual competition which is: give an annual award to the system exhibiting the most lifelike qualities but don't go for a "holy grail" approach... but reserve that and the ability to evolve the criteria as time goes by. He recognized that this is less of a big attractor but is in fact more practical, preventing us from unduly restricting the criteria, or awarding the prize "too soon" when we are expecting to see better results in subsequent years. This in fact like climbing Mount Improbable from the back side, the gentle slopes, like nature herself. We may ultimately be able to offer a "holy grail" final prize.

He was impressed by our (well my) confidence on being able to raise the ultimate 1$ million prize (which funds an annual endowment). He then mentioned a Jeffrey Epstein of New York who in a recent dinner with him and Lala (his wife). Jeffry is a very wealthy man and had asked Dawkins what might be a good use for some of his money. Dawkins had said that funding a prize would be and they came up with a couple of options (discoverer of life on another world etc). Dawkins thought that this indeed would be something worth approaching Epstein about.

He was very keen on the potential educational benefits produced by all of this... and being able to simulate evolution and life to understand it. He is passionate about any move to give the public a greater understanding of evolution. I suspect he likes anything to fight the effects of the creationists (and he mentioned them by name). He noted that the project could backfire in that if we could show that "alive" systems could actually be instanced in a built world (the computer) that the creationists might take this as evidence for god.

He asked us about possible worries about how "viral" systems of the prize could be mixed up with the idea of malevolent computer viruses. However, we talked about this for some time and concluded that the aliveprize could provide a group of serious practitioners in the field who could develop an immune system or practice against the element of malevolence. For example, if your system was malevolent you would be disqualified from the competition. Systems could run in virtual machines or even feed directly from the nets own streams (a more natural environment).

We talked about other big names to be included on the advisory board level: Dan Dennett, Ted Kaehler, Danny Hillis, Kevin Kelly (without becoming "too

California") and I suspect he would help us recruit these too. I suggested Chris McKay (who is an exobiologist Dawkins knows of), Karl Sims, anthropologists (i mentioned CONTACT COTI) and others. We spoke of Steve Grand and Chris Langton. He had also read a post or two by Larry Yaeger so we talked about him too.

I spoke about our work with NASA on a virtual solar system and he mentioned Carolyn Park or someone on Cassini.

A workable structure emerged from the discussions..

Advisors
****
A board of "big name" advisors lending their credibility who would not have much time to commit but would be happy to come to and present at some of the conferences (at least a few at each). I described what we want to do for the "grand opening" event this fall and he was very keen to participate.

The Testers
****
The advisors might recommend these folks, they might be their colleagues, graduate students etc. These folks would sift and shortlist the submitted environments. Others might come up with testers too. Another group of testers would evaluate the biotic virtual environments against the evolving criteria and perhaps run these by some of the advisors who had time that year.

Practitioners
****
Developers and submitters of systems. These would create their own networks of cooperation within the competition.

Organizers
****
Well you know about these folks!

Other points..
****
He agreed that virtual worlds could be rich environments to evolve and visualize these systems in.

We agreed that each system has to be executable and viewable over the net, accessible by the public (at least some level of it).

He took my book off his shelf and asked questions, showing his interest in the virtual worlds medium and we talked about the social and memetic power of the medium. I am hoping to interest him on the memeflow and evolving properties of inhabited cyberspace for a next book.

He seemed to not disapproved of the name "AlivePrize" but wanted to see what it looked like "in text on the computer"

In summary
****

So after this stream of consciousness, the upshot is that with this one key endorsement, I feel we are GO for this project and that the viability of the end of year announcement (DB4) is very high. Dawkins has agreed to come and address the assembled group. A lot of work remains to be done, and recruitment, and fundraising, but I feel that this one is the next mission for Biota.org and indeed possibly the next frontier for this very creative edge of Humanity.

I look forward to us all taking an amazing journey together!

Best,

Bruce
cc S Gold, J Hauser, J Bowman, L Bowman, G Brandt, L Yaeger

B.2 Summary of Meeting with Freeman Dyson, Institute for Advanced Study, Mar 11, 2009

The following report was written to Arnie Levine, IAS Professor/Member about the meeting between Bruce Damer, Galen Brandt, Freeman Dyson for which Dr. Levine had helped us prepare the night before while staying at the Marquand House.

I had set up the 4pm appointment but Piet took us over to tea to find Freeman earlier, saving him having to rush back to his office. After a brief introduction by Piet (which put me at ease) Freeman walked Galen and I back to his office. We spoke about his passion for nuclear disarmament, shared experiences of the cold war (his wife is from the former E. Germany, I lived and ran software labs in post-Berlin wall Czechoslovakia) and then he invited me to launch into the reason I wanted to see him.

For the most part it was me talking (30-40 minutes) with breaks wherein Freeman conjured up something interesting to add. I was taken aback by his blue eyes peering into mine without a single glance or blink. For a minute I thought he was going to conclude I was mad as a hatter but then during a break he uttered the single word "delightful". I later said that both this idea and I had also been described as "mad" by a British academic, to which he said "oh no, that's not a bad thing at all". I let him know we had shared this concept of ratcheting up complexity with you the night before and you had said it was a "holy grail" and had added some great insight (the changing of the laws at different levels) and this contributed greatly to the quality of the presentation.

After I dared to take the concept far out... all the way to evolving Dyson's Trees (and suggested a new concept: Dyson's Disks) he still wasn't blinking or getting up to leave. We sat on in silence and he came out with more ideas. I mentioned my conference up at the Burgess Shale, a recent visit to Bletchley Park to look at the Colossus Rebuild, my work on asteroid gravity towing, mutual friend Tom Ray, and of getting in touch with his daughter Esther, my relationship with Charles Simonyi (who has supported my computer history project) and of our relationship with his son George (ref Barricelli's Universe, his upcoming book). He reacted positively, in a kind of visceral way, to each of these references. So in some real sense we took a jaunt together around a not insignificant fraction of Freeman's universe and personal contacts. I seemed to be hitting all of the major points.

After sitting in silence further and expecting Freeman to summarily boot us out, he picked up his little disembodied strapless watch, but then came out with more ideas. I then broached the subject that perhaps either me or the EvoGrid research concept (or both) might some day find a home at the Institute (in SNS, or even Princeton University?) as it was truly in the spirit of pure theoretical fundamental Biology a la Baricelli and von Neumann. He then described (with palpable pride) of how Biology had finally come to the IAS (and of course, after he has retired ;). He then suddenly said "could you please add me to your mailing list or something" to which I graciously accepted (Freeman as advisor!). Galen mentioned we were coming back next week (Thursday) to which Freeman said "wonderful" and jumped up to look at his calendar. So we are going to meet with him again on Thursday and you again too if you are around. I know Piet has also expressed a desire to meet with you and us as well and that could be very valuable.

As I promised Freeman we would return I sensed he might want just myself and Galen at our follow-up meeting to clarify the EvoGrid concept and about us as people. I will show him some funny yet provocative visuals (a whimsical EvoGrid movie treatment I made with our NASA animator), early outputs from the simulator and my wild diagrams (vaguely Feinman-esque) of the proposed architecture. He is traveling the following Saturday to Kazakhstan to be with Esther and to watch Charles (or Ester if Charles is somehow bumped from his second trip) launch into space as a tourist. So this may be the last chance to meet Freeman for a while.

I believe Galen and I made a real connection with Freeman. At one of the points when he became silent, I injected "I would like to be able to take some of your ideas forward" and mentioned that I was three years shy of fifty (about forty years younger than him) and hoped I had enough remaining brainpower to do something in the world, to which Freeman replied "don't worry, you can get a great deal done in forty years".

-Bruce

Note: a follow-up communication with Prof. Dyson occurred in June 2011 where the results of the work and the next work on the CREATR model were presented to him. He looked at a short abstract and Springer book chapter and was provided access to the full thesis.

B.3 Biota Podcasts Discussing the EvoGrid

The notes and audio from all of the Biota podcasts, produced and hosted by Tom Barbalet, are available at:
http://www.biota.org/podcast/

Listing of Biota Podcasts where the EvoGrid or its predecessors were discussed, in reverse chronological order, 2010-2006:

Biota Special: Miro Karpis, WebGL and the Artificial Life Forum [October 16, 2010] Tom Barbalet welcomes on Miro Karpis to discuss Miro's introduction to artificial life, his work with WebGL on the EvoGrid and the new artificial life forum (currently in beta).

Biota.org Live #71: Warming Barns the World Over [August 6, 2010] Bruce Damer and Tom Barbalet discuss the EvoGrid, ALIFE XII and the rise of SecondLife compatible artificial life simulations.

Biota.org Live #62: DarwinAtHome and EvoGrid Updates [January 23, 2010] Tom Barbalet is joined by Gerald de Jong and Peter Newman to discuss their respective projects.

Biota.org Live #54: Great Projects [September 18, 2009] Eric Burton, Bruce Damer and Tom Barbalet discuss a number of great open source artificial life projects.

Biota.org Live #53: Post-Singular and Post-Prize [September 4, 2009] Tom Barbalet and Bruce Damer are joined initially by Eric Burton to discuss the Singularity movement. They are then joined by William R. Buckley and briefly Rudolf Penninkhof to discuss prizes and the artificial life community.

Biota.org Live #52: Steve Grand [August 21, 2009] Tom Barbalet is joined by Steve Grand, William R. Buckley, Luke Johnson and Bruce Damer to discuss a wide variety of topics.

Biota.org Live #49: Bankers Beware [July 3, 2009] Peter Newman, Jeffrey Ventrella, Bruce Damer and Tom Barbalet discuss the EvoGrid and artificial life's role in contemporary warfare.

Biota.org Live #48: Value and Peter Newman [June 19, 2009] Gerald de Jong and Tom Barbalet are joined by Peter Newman to discuss the implementation of the EvoGrid, what OpenSim means to artificial life and finding the value in artificial life.

Visions of the EvoGrid #1: Scott Schafer (Part 1) [June 2, 2009] Scott Schafer talks with Tom Barbalet about his particular vision of the EvoGrid.

Visions of the EvoGrid #1: Scott Schafer (Part 2) [June 2, 2009] Scott Schafer talks with Tom Barbalet about his particular vision of the EvoGrid.

Biota.org Live #46: Spheres of Influence [May 22, 2009] Jeffrey Ventrella, Gerald de Jong and Tom Barbalet talk about spheres, the reality of the EvoGrid and global warming vs simulation.

Biota.org Live #45: Surviving Chaos [April 17, 2009] Rudolf Penninkhof, Bruce Damer, Gerald de Jong and Tom Barbalet talk about a variety of topics linking with the ideas of artificial life in chaotic environments.

Biota.org Live #44: Summoning the EvoGrid [March 29, 2009] Dick Gordon and Tom Barbalet talk with Bruce Damer about his recent EvoGrid tour and where the EvoGrid is currently.

Biota.org Special: Bruce Damer at FLinT [February 24, 2009] Bruce Damer presents to FLinT. http://www.greythumb.org/blog/index.php?/archives/369-Bruce-Damer-at-FLinT.html

Biota.org Special: Bruce Damer's PhD Transfer [February 16, 2009] Bruce Damer presents an introduction to his PhD on the EvoGrid at SmartLabs in London.

Biota.org Live #42: Mark Bedau [February 6, 2009] Bruce Damer, Dick Gordon and Tom Barbalet talk with Mark Bedau on wet artificial life and a number of additional topics.

Biota.org Live #40: Larry Yaeger Returns [January 9, 2009] Larry Yaeger, Bruce Damer and Tom Barbalet talk about algorithms and teaching artificial life.

Biota.org Live #39: EvoGrid Broad [December 12, 2008] Bruce Damer and Tom Barbalet discuss the EvoGrid Broad.

Biota.org Live #38: the Hobby of Artificial Life [November 28, 2008] Bruce Damer and Tom Barbalet talk about artificial life as a hobby.

Biota.org Live #37: the Very Long Winter [November 14, 2008] Bruce Damer, Gerald de Jong and Tom Barbalet talk about the EvoGrid and an artificial life winter. This show is concluded with Fien and Mitch's version of Black Hole Sun.

Biota.org Live #36: the Cathedral and the Spider [October 31, 2008] Dick Gordon, Jeffrey Ventrella, Gerald de Jong and Tom Barbalet talk about two dividing topics - religion and spiders.

Biota.org Live #34: Open Source Continued [October 3, 2008] Bruce Damer and Tom Barbalet discuss starting an open source project plus more advanced open source topics.

VideoBiota.org Special: Biota 3 Break-Out Session, 1999 [September 27, 2008] This break-out session from Biota 3 in 1999 features a number of people discussing a future project very similar to the EvoGrid. Thanks to Al Lundell for this fantastic footage!

Biota.org Live #33: How to Promote Your Project [September 19, 2008] Scott Davis, Travis Savo, Bruce Damer and Tom Barbalet discuss project promotion.

Biota.org Live #26: the EvoGrid (August Update) [August 1, 2008] Bruce Damer returns to talk about his trip to the UK and the future of the EvoGrid.

Biota.org: Bruce Damer at GreyThumb London [July 11, 2008] Bruce Damer talks at GreyThumb London 7/28/08

VideoBiota.org Special: Bruce Damer before GreyThumb Silicon Valley [June 24, 2008] Bruce Damer talks about Biota, GreyThumb and the EvoGrid before GreyThumb Silicon Valley. Thanks to Al Lundell for this fantastic footage!

Biota.org Live #22: What is the Philosophy of Artificial Life? [June 20, 2008] Gerald de Jong, Bruce Damer and Tom Barbalet explore the Philosophy of Artificial Life.

Biota.org Live #21: What's the Future for Biota.org? [June 13, 2008] Bruce Damer and Tom Barbalet discuss the many possible directions of Biota.org in the future.

Biota.org Live #20: the EvoGrid (May Update) [May 30, 2008] Scott Schafer, Dick Gordon, Bruce Damer and Tom Barbalet talk about the EvoGrid amongst other things.

Biota.org Live #14: the EvoGrid (April Update) [April 18, 2008] Jeffrey Ventrella, Bruce Damer and Tom Barbalet return to discuss the EvoGrid.

Biota.org Live #13: Where's the Secret Sauce? [April 11, 2008] Jeffrey Ventrella, Bruce Damer and Tom Barbalet begin a thought-experiment rich discussion on whether a closed break-through will benefit the artificial life community.

Biota.org Live #10: the EvoGrid [March 22, 2008] Travis Savo, Adam Ierymenko, Brian Peltonen, Bruce Damer, Justin Lyon, Gerald de Jong and Tom Barbalet talk about the EvoGrid.

Biota.org: Bruce Damer at GreyThumb Boston [March 3, 2008] Bruce Damer introduces the EvoGrid at GreyThumb Boston (with a brief history about his work and Biota.org).

Biota.org Live #4: Surreal and Possible Worlds [February 1, 2008] Justin Lyon, Bruce Damer and Tom Barbalet discuss the environment that surrounds the initial ideas of artificial life development. They discuss visual impact, user interface, the background philosophy and the impact of early education in artificial life.

Biota.org: Questions to Will Wright + Lecture from Rudy Rucker [January 30, 2008] From a space/avatars conference at NASA Ames Research (January 26-27, 2008), questions to Will Wright and a lecture from Rudy Rucker.

Biota.org Live #3: The Ultimate Project (part 2) [January 25, 2008] Tom Barbalet raps solo for forty-two minutes and is then joined by Jeffrey Ventrella who talks about his development experiences in contrast.

Biota.org Live #2: The Ultimate Project [January 18, 2008] Tom Barbalet talks with Jeffrey Ventrella and Bruce Damer about the format of the ultimate artificial life project.

Biota.org Live #1: An Introduction [January 11, 2008] Tom Barbalet talks with Justin Lyon and Bruce Damer about the history and directions in artificial life.

Biota.org Chat: Bruce Damer [October 23, 2007] Space, science books, SecondLife update, graphics and artificial life politics.

Biota.org Chat: Bruce Damer [June 19, 2007] Communicating and nurturing artificial life with discussions on SecondLife, Spore and Breve.

Biota.org Conversation: Is Open Source Good for Artificial Life? [January 20, 2007] Gerald de Jong, Pedro Ferreira, Bruce Damer and Tom Barbalet discuss Open Source in Artificial Life.

Biota.org: Bruce Damer [June 27, 2006] Bruce talks about the Biota conferences, Biota.org and Digital Space.

Appendix C: Detailed Implementation and Source Code Examples

Complete EvoGrid Project Documentation is on the web at:
http://www.evogrid.org/index.php/Category:Prototype2009_Documentation

C.1 Components

Simulation Manager

This component acts as the central data distribution point for the batch processing. This uses HTTP for communication, and provides either human targeted XHTML or machine readable JSON.

The Simulation Manager accepts stores and provides:

- Specification for pending simulation jobs
- Histories of completed simulations, for processing by analyzer functions
- Statistics generation by analyzer functions, for processing by searching functions
- Scores generated by both analyzer functions and searching functions.

Due to the amount of data being stored and transmitted, the hardware requirements for a Simulation Manager include disk for file storage, and database storage.

The Simulation Manager provides a method for daemons to request "pending" variations on data to be processed. This allows the Simulation Manager to choose what order data should be processed in, particularly simulation specifications.

To date, the selection method used is ordering by the "priority" property, then random selection from items with the highest priority.

Statistics and scores are currently accepted in an open manner, in that any statistic or score name can be used, and this will be automatically added to the storage database.

If there are no pending simulation specifications, then the Simulation Manager generates new ones, by providing random parameters. The random parameters include the number of atom types present in the simulation. Currently, this seed generation is the only point capable of varying the number of atom types present. The current search function implementation does not alter this.

Simulator

The Simulator component retrieves pending simulation job specifications from the Simulation Manager, performs these jobs and submits the history back to the Simulation Manager.

This is a multiple stage process.

1. The daemon executable retrieves a JSON formatted specification for the simulation.
2. The daemon generates any data not specified in the JSON specification. This includes atom position, velocity and type, based on information that must be in the JSON specification.
3. The daemon produces a GROMACS format simulation specification file. This is specified in the GROMACS binary TPR format. The Energy Minimization option is specified, which instructs GROMACS to remove any overlapping or impossible placement of atoms, due to random generation.
4. GROMACS is executed, with the TPR file for configuration.
5. The daemon reads the produced TRJ file, merging the energy minimized atom positions into the simulation specification.
6. The daemon produces a GROMACS TPR file, with Molecular Dynamics enabled.
7. GROMACS is executed, with the TPR file for configuration.
8. The daemon reads the produced TRJ file. The atom positions and velocities are extracted.
9. Bond formation is performed by the daemon.
10. History data in EvoGrid format is written to disk.
11. Steps 6 through 10 are repeated, until the number of repetitions specified in the JSON specification has been completed.
12. History data is submitted to the Simulation Manager.

Specific Details

GROMACS 3.3 is used. The TPR file specifies only a single processor to be used, for simplicity.

The run/stop/process usage of GROMACS is inefficient, but was the simplest way to implement the bond formation. This method of bond formation was chosen as the Quantum Mechanics features of GROMACS (which performs the same functionality) was not understood at the time. Part way through development it was realized that using QM would be a much better implementation, however it was decided to continue with implementing the current basic method to allow development to continue on the other parts of the system as soon as possible.

Analysis Daemon

This daemon retrieves simulation histories then performs per-frame analysis on the data. Each analysis produces a single floating point value, specified per frame.

Once per-frame analysis is completed, score analysis is performed by processing the per-frame statistics. Each score analysis produces a single floating point value, that describes the simulation history as a whole.

The per-frame statistics and simulation scores are submitted to the Simulation Manager.

Search Daemon

This daemon retrieves simulation scores and performs analysis on these to produce a single floating point score. This score is relevant to the particular search being performed.

The search daemon submits the specifications of any additional simulations it requires, to be queued for later simulation. The submitted specifications include the “parent simulation” property, which specifies which simulation result was used to produce the new simulation specification, and the “priority” property, which is set to the search score.

C.2 Energy Distribution

Find this section online at:
http://www.evogrid.org/index.php/Prototype2009:_Energy_Distribution

The following are the pseudocode formulae used to derive the velocity of generated particles in generator/molecule. The base formulae are taken from evogrid-analysis/temperature, which is taken from gmxdump:

```
ekin = ((vel.x * vel.x * mass ) / 2) + ((vel.y *
vel.y * mass ) / 2) + ((vel.z * vel.z * mass ) / 2)

temp = (2 * ekin) / ( 3 * BOLTZ ) // 3 * due to 3
axis

2 * ekin = temp * 3 * BOLTZ

ekin = ( temp * 3 * BOLTZ ) / 2

((vel.x * vel.x * mass ) / 2) + ((vel.y * vel.y *
mass ) / 2) + ((vel.z * vel.z * mass ) / 2) = (
temp * 3 * BOLTZ ) / 2

(vel.x * vel.x * mass ) + (vel.y * vel.y * mass ) +
(vel.z * vel.z * mass ) = ( temp * 3 * BOLTZ )

mass * ( (vel.x * vel.x ) + (vel.y * vel.y ) +
(vel.z + vel.z) ) = temp * 3 * BOLTZ

(vel.x * vel.x ) + (vel.y * vel.y ) + (vel.z +
vel.z) = (temp * 3 * BOLTZ / mass)
```

To get a single component, we'll consider as if we've divided both sides by three. We'll use a Maxwell-Boltzman distribution to make this randomly distributed over all three axes:

```
vel.x * vel.x = (DIST * temp * BOLTZ ) / mass

vel.x = sqrt( ( DIST * temp * BOLTZ ) / mass )
```

C.3 Bond Formation

Find this section online at:
http://www.evogrid.org/index.php/Prototype2009:_Bond_Formation

The bond formation code currently in place is primitive. It met the requirement of having bonds be able to form during simulation time and effect the result of the simulation from that point. These requirements allowed us to have working data for developing the other parts of the system. It is our intention to replace this with GROMACS QM, and have bonds be an emergent property from the QM.

The bonds are formed based on the distance between atoms. The maximum distance for atom bonding is specified in the simulation specification. This specification applies to all atom types.

The current algorithm used for bond formation is such:

1. For each atom, find neighbors within the bonding range.
2. If already bonded, Done.
3. If both atoms have less then 4 bonds, form a bond. Done.
4. For each atom, test the separation distance for their existing bonds against the current potential bond. Remove the largest separation bond that is larger then the potential bond.
5. Form the bond.

C.4 Example Process

Find this section online at:
http://www.evogrid.org/index.php/Prototype2009:_Example_Process

Simulation Manager

The Simulation Manager contains a list of future simulation jobs with the following specifications:

- These are neighbors of completed simulations, varied by one option (dimension).
- Each future job has an assigned priority score, built from statistic analysis of neighbor completed simulation.
    - NOTE: Initial simulations are selected randomly, to seed future job generation.

The Simulation Manager contains list of un-analyzed simulation jobs with the following specifications:

- These are simulation runs that do not have the full range of statistics or scores.

Simulator

Simulation system requests new job from Simulation Manager

- Receives initial simulation condition parameters
  - This may or may not include specific particle/atom information
  - If not, atoms are generated
    - Atoms are randomly placed
    - Atoms are energy minimized to remove overlaps from random generation
- Generates GROMACS binary format topology.
- Loops:
  - Passes topology to GROMACS to perform short simulation step (1 second?)
  - Load trajectory (atom motion) data generated by GROMACS
  - Analyze data for bond formation
    - Create new bonds (molecules)
      - This includes atom exchange
  - Update topology
  - Loop until full simulation time has been generated
- Submit trajectory data to Simulation Manager
- Submit basic statistics (bond creation) to Simulation Manager

Analysis

Analysis system requests new analysis job

- Receives simulation trajectory data
- Perform statistical analysis
- Analyze statistics to produce scores
- Submit statistical analysis to Simulation Manager
- Submit scores to Simulation Manager

Simulation Manager

Simulation Manager receives scores

- Generate priority for neighboring simulations
  - Certain score types may emphasis work on specific neighbors
- Submit neighboring simulations to future job queue

C.5 Simulation Manager API

Find this section online at:
http://www.evogrid.org/index.php/Prototype2009:_Simulation_Manager_API

General Design

The Simulation Manager is implemented as JSON transferred over HTTP requests. This uses GET and PUT to retrieve or submit information, with any information not in the URL to be in HTTP headers.

All requests start from a base URL. On the development server, this is

http://wizarth.is-a-geek.org/development/evogrid/

This URL uses Apache mod_rewrite to convert to

http://wizarth.is-a-geek.org/development/evogrid/index.php

and is just shortened for neatness. In the Alpha 10-01 VM Images, the path is

http://192.168.230.2/index.php

When accessed directly, the base URL produces a human readable status page.

The URL used to access a resource follows a pattern of

[base_url]/[simulation id]/[property]

Additionally, [simulation id] can be the magic word "pending". Property can be

- parameters
- history
- statistics
- scores

For example:

http://192.168.230.2/index.php/pending/parameters

Providing a incorrectly formatted simulation id will return a HTTP 400 - Bad Request. Redirect means - Provides a HTTP 302 response, and provides a Location: header. When using a GET request, if the data is not available, returning HTTP 404 is the standard response.

See Prototype2009: Data Formats for the JSON used in these requests.

GET

parameters

- pending - Will redirect to a simulation id that hasn't been simulated
- simulation id - Will provide the parameters for the specific simulation

history

- pending - Will redirect to a simulation with a history but no statistics
- simulation id - Will provide the history for the specific simulation

statistics

- pending - Returns HTTP 400, pending is not valid for statistics

- simulation id - Will provide the statistics for the specific simulation scores

- pending - Requires the X-Search-Score header be included in the request. This will return the scores for a simulation that has scores, but not the X-Search-Score
- simulation id - Will provide the scores for the specific simulation

PUT

Unless specified, use of pending will return HTTP 400.

parameters

- pending - Will submit the simulation parameters as a new simulation

All other parameters (like a simulation id) will return HTTP 400.

history

- simulation id
  - If simulation already has a history, returns HTTP 403 (Forbidden).
  - Otherwise, history is stored statistics

- simulation id
  - If simulation doesn't have a history, returns HTTP 403 (Forbidden)
  - Store statistics, discarding duplicates of already know statistics
    - While in development - If statistic name is not recognized, it is added as a new type of statistic scores

- simulation id
  - If simulation doesn't have a history, returns HTTP 403 (Forbidden)
  - Store scores, discarding duplicates of already known scores

Future Developments

- Require identifiers/accounts for PUT actions, for logging and security (prevent poisoning).
- Use multiple sources for all processing that is farmed out (prevent poisoning).
- Break searches separately from scores.
  - Have searches require sets of scores, specified server side.
  - Concept: /pending/search/search-identifier rather then X-Search-Score ?
- PUT /pending/parameters will include which search it's being submitted for.
- GET /[simulation id]/ (without parameters/history/statistics/scores/etc) will return a human formatted status page for that simulation

- Human readable pages could use a microformats concept to remain machine parsable?

C.6 Scoring and Searching

Find this section online at:
http://www.evogrid.org/index.php/Prototype2009:_Scoring_and_Searching

Statistics are generated for each frame of the simulation, then scores are generated from those. These scores are used to calculate the fitness used for the searching functionality.

The Analysis Daemon is implemented as a modular process. The core executable loads external modules, which register their names and calculation functions on being loaded. The core retrieves the history from the Simulation Manager, parses it and feeds the frames into the calculation functions as they are parsed. After each frame has been processed by the analysis function, the frame data is discarded by the core. Any module that wishes to maintain a state must do so itself.

Once all frames are processed, the results from the calculation functions are passed to any registered scoring functions.

The accumulated per frame statistics and the scores are then submitted to the Simulation Manager.

The currently implemented modules are:

- Temperature
    - Per Frame
        - Temperature Matrix
          Divides the atoms into a grid, calculates the average temperature of each cell. This is calculated from the atoms velocity and mass. Uses a pre-processor to allow it to perform the calculation in one pass, rather then recalculating for each grid cell (which is returned as a single statistic)
        - Simulation Wide Temperature
          Provides the accumulated temperature, across all the cells
    - Score
        - Temperature Mean
          The mean temperature across all frames of the simulation.
        - Temperature Mean Change
          The difference between Temperature Mean and the initial temperature provided in the simulation specifications.
- Errors
    - Score
        - Failed Frames
          A negative score that indicates how many frames of the

        statistics are 0. This indicates that no history was available for that frame, due to the simulator crashing.

- Molecules
    - Per Frame
        - Max Molecule Size
          The size of the largest molecule in this frame.
        - Avg Molecule Size
          The average molecule size.
    - Score
        - Avg Avg Molecule Size
          The mean average of the Avg Molecule Size statistic. This indicates multiple molecules forming and persisting.
        - Avg Max Molecule Size
          The mean average of the Max Molecule Size statistic. This indicates large molecules forming and persisting.
        - Max Avg Molecule Size
          The maximum value of the Avg Molecule Size statistic. This indicates the largest amount of molecules formed, or the formation of few large molecules.
        - Max Max Molecule Size
          The maximum value of the Max Molecule Size statistic. This indicates the size of the largest molecule formed in the simulation.

Searching and branching

The currently implemented search daemon is “search-evogrid-complexity-1”, to indicate that this is a search score, for the EvoGrid project, searching for complexity, and is the first algorithm for this.

This search is simple. If Failed Frames are zero, then all the scores from the Molecule module are added together. This value is used as the fitness score for the searching function. The use of these attributes for the fitness function will search towards production of long chains.

The specifications for the simulation that produced this score is then copied and varied multiple times. Each variation is the same as the original, with one floating point variable altered by 1%, either incrementing or decrementing. Each of these single variations are submitted to the Simulation Manager as a new specification.

The number of variable parameters is dependent on the number of atom types present in this simulation. The parameters are:

- Density
- Temperature
- Bond Outer Threshold
    - Atom Type Radius
    - Atom Type Mass
    - Atom Type Q
        - Atom Interaction Force c6
        - Atom Interaction Force c12
        - Atom Interaction Force rA

- Atom Interaction Force krA
- Atom Interaction Force rB
- Atom Interaction Force krB

Additionally, the ratio of atom types is varied in the same manner, with each value increased or decreased by 1 percent, then normalized so the ratios continue to make up 1.0 .

Due to the interaction forces between each atom type being mutable, as well as the ratios, the number of variations submitted to the Simulation Manager varies from 138 (for 3 atom types) to 1286 (for 10 atom types).

Contiguous Time

As currently implemented, the newly submitted simulation specifications do not include specific initial conditions, such as atom position, velocity or bonding. As such, any molecules formed during the simulation will not carry on to any of the branched submitted simulation specifications. This means the search is performed purely among the initial specification meta data.

Future development involving contiguous time could use many of the currently used variations, however variations to ratios, density and temperature would not be relevant. Additional variations that are possible would include random alterations to atomic velocities, or random removal of existing bonds.

C.7 Source Code Examples for Scoring and Searching

priority_simple used in Experiments 1 & 2

```
/**
Assigns the score as the priority.
This is the simplest priority function, but does not take
any measures to avoid a wide search at a plateau.
*/

float priority_simple( const struct
simulation_parameters* sim_params, float score )
{
   return score;
}

// Copy-paste from evogrid-search-complexity
float get_score( const char* score_name,
igraph_vector_ptr_t* score_names, igraph_vector_t*
score_values )
{
   int i;
   for( i = 0; i <
igraph_vector_ptr_size(score_names);++i)
      if( strcmp( score_name,
char*)VECTOR(*score_names)[i])==0)
         return VECTOR(*score_values)[i];
   return 0;
}
```

priority_generation used in Experiments 4 & 5

```
/**
Assigns a score based on the improvement in score
compared to the previous generation.
Priority is calculated relative to the priority of the
previous generation, allowing a promising line to
"gather" priority that is decreased over generations of
poor improvement.
*/

float priority_generation( const struct
simulation_parameters* sim_params, const char*
score_name, float score )
{
   /*
   Get the simulation_parameters for the parent
simulation
   Get the priority for the parent simulation.
   Get the scores for the parent simulation
   Get the fitness score for the parent simulation
   Calculate the priority from this information.
   */
   float parent_priority = 1, parent_fitness = 0;
   struct simulation_parameters parent_simulation;
   struct score_handle* parent_score_handle;
   float priority;

   if( sim_params->parent_id )
   {
      retrieve_simulation_parameters( sim_params-
>parent_id, &parent_simulation );
      parent_priority = parent_simulation.priority;
      parent_score_handle = init_score_handle();
      register_score_function( parent_score_handle,
score_name, &priority_generation_score_callback );
      set_score_simulation_id( parent_score_handle,
sim_params->parent_id );
      priority_generation_score_callback_ctx.score_name =
score_name;
      priority_generation_score_callback_ctx.score =
&parent_fitness;
      process_score_handle( parent_score_handle );
      clean_simulation_parameters( &parent_simulation );
   }
   priority = parent_priority * ( 0.9 * exp( score
parent_fitness ) );
   return priority;
}
```

C.8 Source Code Examples for Branching

See this lengthier source code online at:

http://www.evogrid.org/index.php/Prototype2009:_Scoring_and_Searching#Searching_and_branching