

The Ecology of E-Business

Thang N. Nguyen
California State University Long Beach
(tnnguyen@csulb.edu)

Abstract

Biological and ecological metaphors and analogies have been the source of inspiration for many investigations, models and achievements in computing and business. Primarily, biological metaphors have inspired computer scientists and engineers in numerous disciplines, while ecological metaphors have become suggestive to many business models, processes, and formulation of strategies. Recently, there is a fast growing trend towards ecological approaches to business formulating its systems as business ecosystems and/or Internet ecosystems. Our work exploits the same view of business/Internet as an ecosystem. However, there is one basic difference between our formulation and others. In our view of the business and the Internet as ecosystems, we recognize that the business/Internet ecosystem supporting software is a logical part of a *software continuum from bits (at low-level, as primitive building blocks) to Internet ecosystems (at high-level)* much like the *natural continuum from particles to ecosystems*, hence the concept of *ecology of software*. In the software continuum that supports the ecology of software, e-business software application systems are considered as analogous to living organisms, the main level of organization of the natural continuum. In this paper, we argue that the ecology of software is the key to the notion of ecology of e-business which is considered as a derivative of the notion of ecology of business much discussed in the literature recently. We also argue that the ecology of software substantiating the software automation within the ecology of e-business fits well within the framework of the general system theory, at an implementable level of details. Our investigation focuses on specific concept development, architecture and implementation which center initially around three main areas: the birth of genomic programming for the creation of software species, architecture for robustness in automation and intelligence, and architecture for high-level (symbolic) – low-level (numeric) integration. Our contribution from this novel view consists of (1) a nature-based and nurture-driven model for guiding the development of autonomous and intelligent e-business software automation, and (2) a formulation of potential applications servicing the digital economy. These applications include but not limited to autonomous and intelligent e-business software, Internet application interoperability, business-IT integration, e-crisis management, and systemic capabilities such as fully-connectedness, self-organization, self-management and/or emergence.

1. Introduction

Over a period of six decades, biological metaphors have influenced the thinking and the making of computer and its capabilities as well as its system software and applications in many widely diverse disciplines. These included the early von Neumann's computer model (CPU, memory and input/out devices) from his stored program concept and his cellular automaton model, to genetic algorithm, genetic programming, and artificial neural nets of many different flavors. The models and their associated mechanisms and algorithms have defined many areas of applications in business, science and engineering.

On the other hand, the ecological metaphors have emerged only since the early-to-mid 90's with the analogous formulation of concepts starting with the concept of ecology of competition (and its business ecosystems in 1993) introduced by James Moore. Several years later, around 2000 the academic literature and business/industry press has noticed the introduction of the concept of ecology of partnerships (and its Internet ecosystems) by Cisco Corporation. This was followed by a multi-year, multi-national, and multi-organization consortium in Europe in 2003 which initiated the Digital Business Ecosystems Project. Recently, the concept of ecology of strategy was formulated by Iansiti and Levien with the introduction of performance measures for business ecosystems.

In between these two extremes, we have observed the work on amorphous computing, ubiquitous computing, and artificial life addressing mid-level organizations (e.g. population, community), swarm intelligence, and flock complexity.

While the biological metaphors resulted in numerous investigations for modeling low-level architecture or for creating algorithms, the ecological metaphors induced the formulation of high-level modeling of business processes, organizations, and strategies. Interestingly enough, while the biology-ecology spectrum constitutes a natural continuum from particles (and below) to ecosystems (and beyond), there have been no true connection between models derived from numerous biological approaches to computing and those from diverse ecological approaches to business. These investigations stayed disconnected among one another, thus hindered an integrated view of how they will work together.

This paper accounts for the issues raised from the disconnection between (1) the various ecological approaches, metaphorically and analogously, to business models and solutions, and (2) the numerous biological approaches to computing models and applications, and introduces the notion of ecology of software as a complement to all the above ecological approaches to business and biological approaches to computing.

Unlike the business ecosystems and Internet ecosystems which are formulated and modeled by mimicking real-world ecologies, the ecology of software [Nguyen, 2002] is not architected metaphorically. Instead it is based on the underlying notion of software continuum which was shown generally parallel to the natural continuum. It has been argued [Nguyen, 2005] that the natural continuum ranges from particles to ecosystems while its biological equivalent, the software continuum, covers from bits to business ecosystems, organizationally.

In this paper, we argue that the ecology of software is the key to the notion of ecology of e-business which is considered as a derivative of the notion of ecology of business much discussed in the literature recently. We also argue that the ecology of software substantiating the software automation within the ecology of e-business fits well within the framework of the general system theory, at an implementable level of details. Our investigation focusing on specific concept development, architecture and implementation which center initially around three main areas: the birth of genomic programming for the creation of software species, architecture for robustness in automation and intelligence, and architecture for high-level (symbolic) – low-level (numeric) integration. Potential applications include those in large system and software development, and in business (strategy)-IT (operations) integration.

The remaining of the paper is organized as follows. In section 2, we present an overview of ecological approaches and an accounts for the biological approaches. In section 3, we present our formulation of the software continuum concept. Section 4 presents numerous implications to the real-world problems and solutions. We propose a prototypical model for the ecology of e-business (section 5), and discuss some potential applications in section 6. We offer our concluding remarks and future work in section 7. This paper in fact is a condensed version of our multiple investigations from different angles of software continuum and the ecology of software, which have appeared in the literature over the last few years. The theme is larger, however. It addresses the ecology of e-business.

2. Ecological approaches to business and biological approaches to computing

Ecological approaches to business

The term “ecology” was coined in 1866 by Ernst Haeckel, a German biologist, to indicate a scientific discipline that studies the relationship between individuals of a species, its organized activity and the environment of this activity [Library of Congress, 2005]. On the other hand, the term “ecosystem” by Arthur Tansley, a British ecologist, did not surface until 1935. The latter indicates the interactive system established between organisms and the environment in which they live. Within an ecosystem, organisms are dependent upon one another in the food chain. In this food chain, there are three main categories of organisms: producers (e.g. plants), consumers (e.g. animals) and decomposers (i.e. microorganisms such as bacteria, fungi).

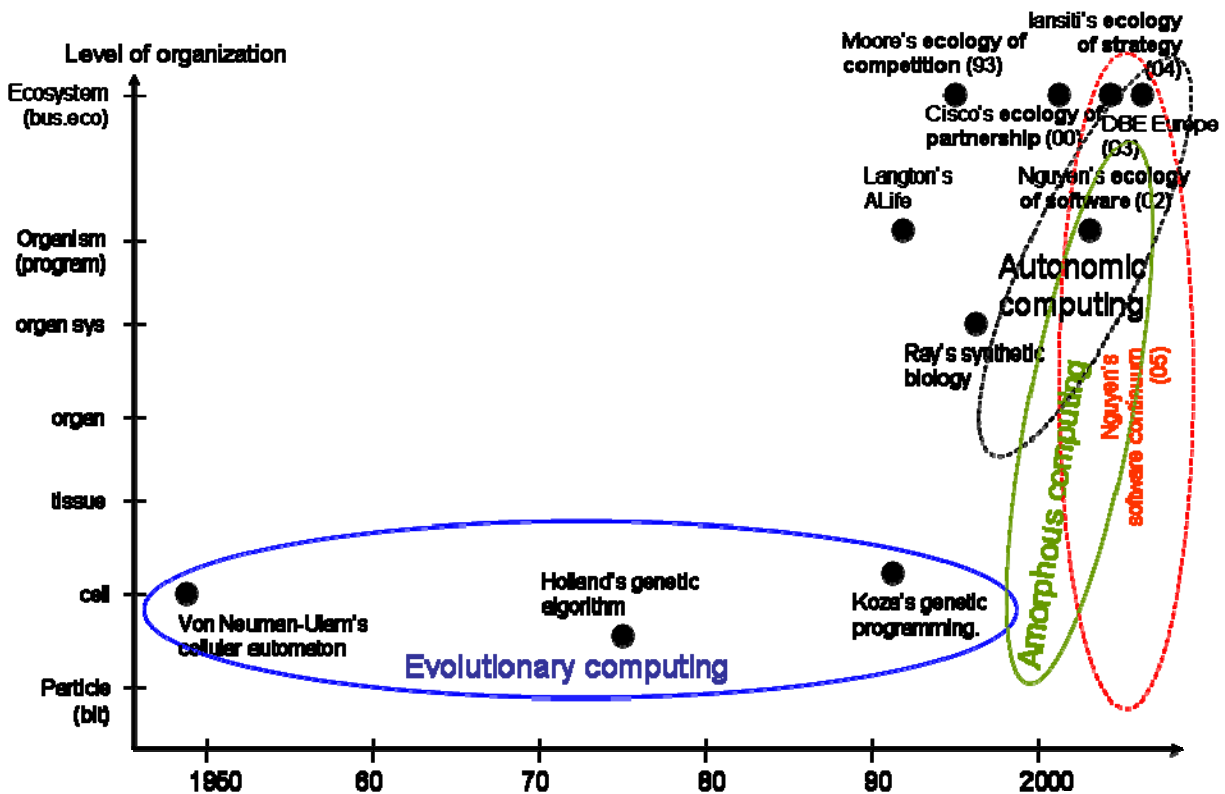


Figure 1: Biological computing landscape

The discipline of ecology and its ecosystems have been the source of inspiration for many business researchers and computer scientists for more than one decade. In fact, in 1993, prior to the explosion of the Internet, James Moore in his award-winning article appeared in Harvard Business Review coined the term *business ecosystem* (Figure 1 – top right). Moore's view focuses on a very broad perspective of all businesses as a whole and recognizes the existence of different business ecosystems that co-evolve. Moore's view characterizes a particular business ecosystem in the *ecology of competition* as emerged from four stages: birth, expansion, leadership and self-renewal/death. The four-stage evolution model offers a framework for business leaders and managers to develop appropriate strategies and timely decisions to avoid failures leading to the "death" of the business.

Most recently, Iansiti and Levien in another Harvard Business Review article in March 2004 looks at *the ecology of strategy* and suggests three basic indicators for measuring the health of a business ecosystem: productivity, robustness and niche creation. They propose the niche creation of valuable new functions via keystone strategy within the ecosystem.

Some time around the end of the millennium, Cisco Corporation introduced the view of the *Internet as ecosystem*, thus the ecology of partnership) in its business model formulation. It denotes the latest stage, the digital ecosystems, of the 6-stage Internet-based technology evolution adoption from e-mail, website, e-commerce, e-business, networked organizations to digital ecosystems. It is conceived as a strategic alliance of business partners to offer complete end-to-end solutions. Participant business partners "must understand, support, and promote the other partners" in a web of interlocking relationships. Partners are to align themselves and are committed to "creating, marketing, and delivering integrated solutions to their customers" to bring a wealth of tangible and intangible benefits to every member of their alliance in the Internet economy.

Cisco's Internet model as well as the work of Moore was subsequently adapted by a consortium of 20 European academic institutions/business organizations in its Digital Business Ecosystem (DBE) project initiated in late 2003. This DBE project is targeting at a biologically-transposed open source, component-based software infrastructure for application development by small and medium-sized enterprises (SME) to guide SME software providers and to foster local and regional economic development.

Biological approaches to computing

At both the cellular level and organism level, John von Neumann fathered the concept of cellular automata (Figure 1 – bottom left). The concept is based on the three abstractions: (1) cross-over and (2) mutation in heredity and genetics (after the Mendelian model) and (3) natural selection (after the Darwinian evolution theory) to arrive at a mathematical model for solving problems such as search and optimization. The model was realized by John Conway in his game of life.

Von Neumann's cellular automaton was extended by John Holland into a form known as genetic algorithm and by John Koza into genetic programming. Together they define what is today called evolutionary computing. At the cell and organ level, the artificial neural network (ANN) is modeled mathematically after a single neuron model by Warren McCulloch and Walter Pitts. Three layered neural nets have been realized with many applications.

At the tissue and organ levels, in the 40's, John von Neumann used biological metaphors of the brain and memory to create the stored program architecture of early computers that is still being used today. Anthropomorphic robots have been modeled after human limbs: multifingered hands [Michael Arbib and Thea Iberall 1990, Thang Nguyen and Harry Stephanou 1992], as well as legs and arms by others. Recently, J. Yasha Kresh and his colleagues have looked at the heart as a complex adaptive system [Kresh et al., 2003]. Ray Paton and his colleagues have introduced a computational model of the human liver [Paton et al., 2004]. At the organism level, there were the unconventional bug-like and humanoid robots created by Rodney Brooks and his team.

At the cognitive level, an experiment by Alan Turing in the 50's [Turing, 1953], has evolved into an active research area called artificial intelligence (AI) - a term coined by John McCarthy [McCarthy 2003]. AI has focused on human symbolic-logic capability in declarative and procedural knowledge (expert system, knowledge-based system, decision support system). Many others have pursued different issues such as machine learning [Marvin Minsky], common sense reasoning, and mathematics-based reasoning under certainty such as Bayesian theory, fuzzy logic by Lofti Zadeh, or the theory of evidence [Dempster-Shafer].

Major biological breakthroughs and advances have changed the previous research directions somewhat. These include (1) great discoveries such as DNA by [Crick and Watson, 1953] which has defined molecular biology, (2) great theoretical formulations and models such as Hebb's cell-assembly in psychology in 1949, and (3) daring research projects such as the Human Genome Project coordinated by US Department of Energy starting in 1990. Molecular biology has been the main source of many newer discoveries. Hebb's cell assembly formulation not only impacted the field of psychology, but also helped create the theory of connectionism in Artificial Intelligence and a research field known as Parallel Distributed Processing. The Human Genome Project, finished in 2003 with a complete sequencing of all 30,000+ human genes, has prompted researchers into new trails.

Other advances in biologically-relevant research beyond ecological approaches to business and biological approach to computing are noted below. With the new gene sequencing, Gary Marcus attempted to shorten the space between the genes and the mind (Marcus, 2004). Albert Barabasi and his colleagues noted the similarities between cellular networks and the Internet [Barabasi et al. 2004]. They found that both follow the power-law distribution, rather than a bell curve distribution from a random network as originally expected. Also pertinent are numerous results found in developmental biology, neurophysiology, neuroscience, neuropsychology and other disciplines by many noted researchers: Rafael Lorente De No, E. Kandel, K. Lashley, Still other related research streams include systemic features of complex systems investigated at all levels of organization such as connectedness (fully-connectedness), self-organization, emergence, and the notion of wisdom of the body in the sense used by Ernest Starling, Walter Cannon and Stephan Nuland.

Some important Issues

Interestingly enough, Cisco's, Moore's and lansiti and Levien's formulations (except maybe DBE) largely ignore, in their analogies, the biological building blocks of ecology from particles, cells, tissues, organ, organ systems, organisms, population, community to ecosystem. Consequently all three formulations stay high-level, focusing on strategic issues of business/Internet ecosystems. The details on how to develop tactics at the tactical level and decision making at the operational level in support of business strategies remain mysterious and are in the hands of capable business leaders and managers.

Enterprise architectures in the sense of Zachman or of De Boever and/or information architecture appear disconnected from these formulations. Although, there has been a lot of discussion on the ecology of business, little attention was given to the ecology of e-business. As a matter of fact, a search on Google revealed there was only one entry on the topic of "ecology of e-business". It appears that with the explosion of the Internet, software automation ought to be investigated or given some higher priority. Also, within the Internet environment, the issue of intelligent software capable of automation largely remains within the scope of intelligent agents in the sense of Maes.

3. The software continuum concept and the notion of ecology of software

Our thoughts on the ecology of software exploit the same view of business/Internet as an ecosystem. However, there is one basic difference between our formulation and others. In our view of the business and the Internet as ecosystems, we recognize that the business/Internet ecosystem supporting software is a logical part of a *software continuum*.

The software continuum ranges from bits (*low-level, basic primitive programming building blocks, biological resemblance to particles*) to business/Internet ecosystem (*at high-level, biological resemblance to natural ecosystem*). It is generally "parallel" to the *natural continuum from particles to ecosystems* [Nguyen 2002]. In this *software continuum* that actually exists [Nguyen 2004], not created metaphorically, software application systems supporting e-business ecosystem are specifically considered as analogous to organisms in the natural ecosystem. This is different from other formulations in which many different things (e.g. businesses, organizations, etc.) are considered as organisms [Moore 1993, Cisco 1999, lansiti et al. 2004].

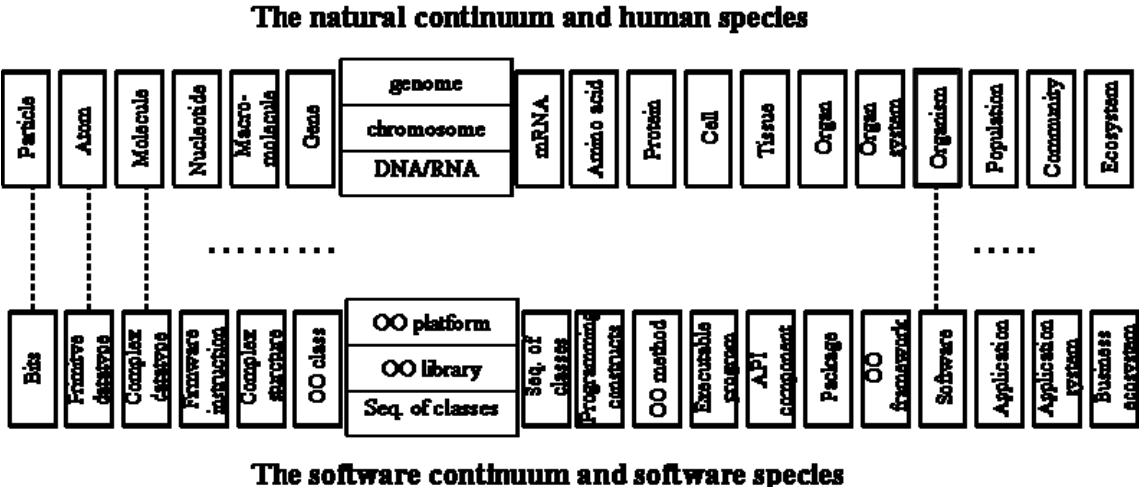


Figure 2: Natural continuum versus Software continuum

Details of the parallelism between software continuum and natural continuum has been drawn in our earlier investigations [Nguyen, 2000-2003]. The following summarizes the parallelism at different levels of organization as shown in Figure 2.

- Bits considered as particles, simple data structures as atoms, complex data structures as molecules
- OOP platform as genome, OO library as chromosome, OO class as gene and sequence of OO packages in the library as DNA.
- Basic instructions as nucleotides, programming constructs as amino acids, compiled codes as mRNA, OO compiled and link-edited function calls (methods) as proteins, utilities as enzymes, and compilation/linkage-editing as transcription/translation process
- OO executable programs as biological cells
- Data as matter (proteins), information as energy (ATP – adenosine triphosphate), messages as secreted chemicals or action potentials, message formats as neurotransmitters
- OO components/patterns as biological tissues, OO frameworks as organs, OO software products as organ systems and OO software applications (“species”) as living species (organisms)
- OO e-business software as population/community and software business ecosystems (Internet ecosystems) as natural ecosystems

4. Implications of the software continuum concept and the ecology of e-business

Our view of software continuum gives rise to some intriguing considerations for guiding the development of software in general and modeling the e-business software automation in particular as follows.

Implication 1 – Should we extend OOP to become “genomic” (or partially “genomic”) rather than as a toolset?

Genome specifies organism. Programming languages specify software. Conventional languages (such as C, LISP), specially designed languages (such as ACL - Agent Communication Language or Thomas Ray’s Tierra) and/or OOP languages (such as C++ or Java) have become a set of *powerful tools* in many previous and current biologically-inspired computing research and related disciplines. We have seen ACL for agent-based software systems, Tierra for synthetic biology, JGAP package for genetic algorithm as research tools. Genomic Perl has been around as a product tool in bioinformatics. Other OO packages have been created for solving problems in many areas of biological computing e.g. cellular automata and genetic algorithms. In the business/industry, the integration of packages such as SAP R/3, Peoplesoft, and component-based software such as Microsoft COM/DCOM, OMG CORBA, or Sun Java/RMI in late 90’s early 2000’s has provided partial solutions to e-business integration and enterprise application integration. The integration, called business framework, appears patch work, and far from considered as robust in the sense of robust intelligence software or robust autonomous software.

We start with a question on a different role of OO language. We ask *whether OOP* (object-oriented programming) *platform* or *paradigm* such as J2SE/J2EE or Microsoft .NET, *can be considered as similar to genome* rather than being a toolset? After all, the genome is “thought of as a summary of many generations of previous experience in dealing with environment” [Grobstein, 2004] and similarly OOP platform can be thought of as a summary of many generations of programming experience since the days of machine language and high-level procedural languages such as COBOL, Fortran, Basic, PL/I, Algol, and C (created by Brian Kernighan and Dennis Richie). Current OOP paradigm was initiated from Simula (created by Kristen Nygaard and Ole-Johan Dahl), Smalltalk (Alan Kay and Adele Goldberg is becoming very powerful as seen in C++ (Bjarne Stroustrup), C# of Microsoft .NET and Java (Jim Gosling) of J2SE/J2EE platforms.

It appears that making OOP “genomic” is viable since it is just the next logical step of programming functionality for use in computational biology and/or biological computing. However the idea may sound foolish since much of genomic mechanism is still unknown although the Human Genome Project (HGP) has just completed its 30,000+ gene sequences in 2003 (2 years earlier than originally planned – Note that it took a good 13 years of international and collaborative effort among many researchers and practitioners). Work on genomics and proteomics to understand gene and protein functionalities are just the beginning.

Whether or not the idea of making OOP genomic is foolish or viable, we do recognize there is a general “resemblance” between OOP and genome (Figure 1). Some small parts of this resemblance have been pointed out or used as metaphors or analogies, here and there over the years. But no prior work has recognized the striking similarities between genome building blocks and OOP building blocks (as indicated in Figure 1) as a continuum from bits to ecosystems at all levels of organization. In other words, we would like to consider whether OO libraries can be considered as chromosomes, OO classes as genes, a sequence of OO classes across the library as DNA sequence, basic machine instruction set as set of codons, basic programming constructs as nucleotides, single algorithms as exons, OO methods (functions) at class level or instance level as proteins, OO utility methods as enzymes, and OO executable programs as biological cells. At levels higher than cell level, we ask whether OO API and patterns (application programming interface) are similar to tissues, OO components (e.g. COM/DCOM) to organs, OO frameworks (e.g. PeopleSoft, SAP R/3) to organ systems and OO applications/application systems (or software species/organisms) to organisms and beyond (Figure 1).

Intuitively, we know that the resemblance or similarities between corresponding levels of organization if existed will not be complete or perfect. Some similarities between corresponding constituents express strong resemblance, others weaker. Also differences do exist. We wonder if the parallelism shown in Figure 1 is good enough for us to model one from the other and to look into solutions of problems in software from biological mechanisms due to possible “homeomorphism” between them. After all, the map of Los Angeles does not resemble the City itself but it is good enough to help visitors findings locations of interest or navigate through the City. Thus, given the partial parallelism, structurally and organizationally, the question is *“what natural principles and/or mechanisms can we learn and derive from the natural continuum for applications in the object-oriented continuum from bits to business ecosystems”*. Specifically, given the complexity of the natural continuum, *“how can we possibly see any simplified way to adopt or adapt any guiding principle and mechanism for use in autonomous and intelligent software automation?”*

The above is the most complete and most consolidated view of all previous analogies between natural systems and man-made system by different and prior researchers and authors. Since partial analogies have given insights to many solutions to prior computational problems, one would hope that this more complete analogy and parallelism will give rise to even more elegant solutions. The question is then *“how can we use this mapping for all practical purposes?”*

One idea arises from this mapping. *“Can we define some gene-like OO classes such that OO programs using these classes would operate in a somewhat cell-like manner?”* *“Will these programs make up of some tissue-like components for some organ-like and some organ system-like frameworks for selected human capabilities?”* *“Is the mapping possible beyond the organism level, i.e. at the cognition level as well as population, community and ecosystem levels?”*

This appears to be a collaborative and giant task, even with completely known genes resulted from the recently finished Human Genome Project and the current pursuit of genomics and proteomics. The idea seems far-fetched but it is not more far-fetched than the initial idea about the Human Genome Project, or the idea on dynamic collaboration between business partners in Cisco Internet ecosystem, or that of Moore and Iansiti and Levien in their ecological processes. It has the same level of difficulty and complexity of effort proposed by European DBE’s in guiding its software providers, if not lower. It is however a long-term and expensive effort. And it needs to embrace additional considerations as argued below.

Implication 2 – Do embryonic processes and mechanisms have homeomorphic transformations in software continuum?

It follows from consideration 1 (if OOP is genomic) that the e-business application development process from conceptualization to implementation can be investigated from an embryonic approach much like the development process in embryology whereby a zygote develops from fertilization into embryo then fetus and then newborn. This embryonic approach to software is different from John von Neumann’s cellular

automation, John Holland's genetic algorithm or John Koza's genetic programming in solving complex search and optimization problems (although it can still benefit from these existing schemes).

In fact, cellular automata or CA [von Neumann, 1951] and genetic algorithm or GA [Holland, 1992] mimic the idea of biological crossover, mutation and natural selection to search for the best algorithm suitable to a certain problem to be solved from a population of group of adjacent cells or binary strings. In the case of GA, each string encodes the presence or absence of a particular characteristic in a structure of a program obtained from a genetic code representing it. The technique amounts to a classifier system associated with rules of selection evaluating the fitness of the strings. The result is a computer program that evolves to arrive at an optimum solution. The process uses the natural selection. This is different from the conventional technique that requires specifications a priori of all features of the program to be written that hinders the software robustness.

Genetic programming (GP), commonly considered as an extension of GA, fathered by John Koza, is a method for automatically creating a computer program from a set of specifications of what need to be done without the needs of specifying how to it in contrast with common practices in conventional method [Koza et al., 2000]. The method involves the selection from a large collection of randomly-generated computer programs by way of fitness evaluation, Darwinian selection and genetic operations that include reproduction, crossover, mutation and other operations patterned after gene regulation.

It is noted that these schemes do experience limitations due to the partial analogy or partial mimicry. AI symbolic-logic schemes for example can only address procedural and declarative knowledge and has little connection to artificial neural nets. Artificial neural nets (ANN) ignore the role of endocrine physiology which works in conjunction of nervous system. Connectionism based on Hebb's cell-assembly model has little in common with agent-based intelligence models of Patti Maes. One would think if OOP is considered as genomic then an embryonic approach to software development would hold promises.

Embryonic and fetal development insights into software conception and early development

The embryonic and fetal (i.e. prenatal) development is relatively well understood in cellular biology [Alberts et al. 1996] and embryology [Moore and Persaud, 1998]. The human prenatal development consists of two periods: the first period includes both germinal phase (first 2 weeks) and embryonic phase (from week 3 to week 8). The first period encompasses 23 stages [Carnegie Collection, 2003] defined by Streeter [Visible Embryo, 2003]. The second period is called fetal phase (from week 9 to term). From fertilization, four major activities occur during these 3 phases of the two periods: cleavage during germinal phase, patterning and differentiation during embryonic phase and growth during fetal phase [Kimball, 2003].

One would think it constitutes a good basis for software development to mimic the relatively well-understood human development due to the following reasons and arguments: (1) the available results obtained from the Human Genome Project initiated from 1990 and ended in 2003 that completely sequenced the 30,000+ human genes, (2) ongoing research in genomics and proteomics to discover the functionalities of genes and proteins, (3) the genomic Perl as well as plenty of Java packages that support research in bioinformatics, and the power of today's personal computer and its utilities.

The embryonic and fetal mimicry appears intriguing and desirable in the long run however it is very impractical, at least from our current research point of view because it would involve the access to the HGP gene sequences which are themselves very complex. It would involve a call for collaboration from multi-nation, multi-organization/institutions, with multi-billion dollars to undertake the task of creating a genomic OOP for embryo-like and fetus-like software foundation. It would take some organization such as Cisco Corporation (e.g. Internet Ecology), US Department of Energy (e.g. HGP), or the European consortium (e.g. Digital Business Ecosystems) to initiate it with Microsoft (.NET), or Sun Microsystems (Java platform) and many other participating vendors. This is not the position we are in.

A couple of ideas can be drawn however from embryology and developmental biology. First, generally, the four activities cleavage during germinal phase, patterning and differentiation during embryonic phase,

and growth during fetal phase, suggest four corresponding major activities in OO software development: conception, patterning, differentiation and growth. Also, note that the first trimester is the most important trimester of gestation where patterning and differentiation occur and where major organs are derived from three basic germ cells: ectoderm, mesoderm and endoderm. In fact, ectoderm is precursor of the brain and skin, mesoderm of the heart and circulation and notochord, and endoderm of the glands, gut and other internal organs such as those of the digestive system. At the end of term (approximately nine months), a newborn is both anatomically and physiologically *complete* to face his/her living environment. What it means is that a software organism could be built so that it is focused on the development of major organ-like at least anatomically and physiologically *sufficient* to deal with its operating environment. In other words, we might want it to have functionalities similar to some major organs and organ systems such as those found in nervous system for control, circulatory system for communication, sensory system for perception, immunization system for error handling and protection, and digestive system for information manipulation.

Thus, the development process should be equipped with appropriate development tools to allow those functionalities to be built. And, rather than mimicking embryonic development, we can look at the finished product of this prenatal phase, i.e. the newborn, and attempt to provide similar functionalities to the software newborn. In other words, our proposed software newborn (at the organism level) should contain the code segments having the role similar to the following functionalities:

- some of the senses and sensory system (for GUI) in Web server
- the brain and nervous system (for control, perception and decision making) in application server
- the heart and circulatory system (for communication), in application server
- other organs and the immune system (for error handling) in e-business applications
- the limbs (connectors and adapters for connectivity and action)

More importantly, these organ-like and organ system-like functionalities should provide containment, fully-connectedness and interface just like interstitial fluid and blood plasma as media for all cell communication among tissues, organs, and organ systems in an organism.

Implication 3 – How can we improve a software newborn in a fashion similar to the nurtured postnatal development of human newborn?

At the center of all levels of organization from particles to ecosystem in the natural continuum, the organism level, human species in particular, appears quite fascinating. It is observable that normal and healthy adult humans carry out their daily activities, physical or mental, with amazing ease and robustness, as opposed to the clumsiness of the newborn. Their activities can be as simple as raising a hand, speaking or hearing a word. They can be more involved such as remembering a past experience. They can be quite complex such as making a decision requiring some deep thinking process. The human capabilities include but not limited to sensation, perception, cognition, memory, emotion, learning and intelligence. Humans can eventually become, via nurture, growth and development, athletes, artists, business people, academicians, scientists, philosophers, etc. They become capable of understanding nature, modeling it, using it to their advantages and/or even modifying it. These activities are naturally developed during prenatal period and nurtured during (postnatal) infancy and/or learned throughout childhood, adolescence and adulthood.

In fact, the ease and robustness of human activities and other capabilities are the results of years of a human's growth and development, beginning with the initial formation of the zygote. The prenatal developing capabilities are gene-based (nature-driven), with the contribution of million of proteins synthesized following the transcription-translation process of the central dogma [Alberts et al. 1998]. During postnatal period, they are nurture-based (infancy) and learning-based (experience-driven) from childhood to adulthood. Nature occurs during the early stages of a human life (from fertilization and beyond). Nurture [Marcus, 2004] occurs during the later stages (beyond fetus, from birth). Human capabilities are also influenced by learning from numerous life experiences and from the acquisition and use of different domain knowledge.

Implication 4 - Can e-business software automation be knowledgeable and robust

Let's examine the connection and relationships between four "elements" of Figure 2: (1) the *Internet* ecosystem in which software species interoperate and (2) the natural ecosystem in which organisms live, (3) the *software automation* on the Internet (in particular e-business software automation) and (4) the human's *wisdom of the body* (WOB) a term coined by Starling [Zajicek, 1995], a biologist, to indicate the mind or instinct in a healthy organism. The Internet is an interconnected network of computer networks via cables, wires, satellites and/or radio links over which all developed business software automatically communicates over the TCP/IP protocol suite and other protocols (e.g. IIOP, SOAP) so that it can support businesses involving in an Internet ecosystems in the sense of CISCO or that of James Moore.

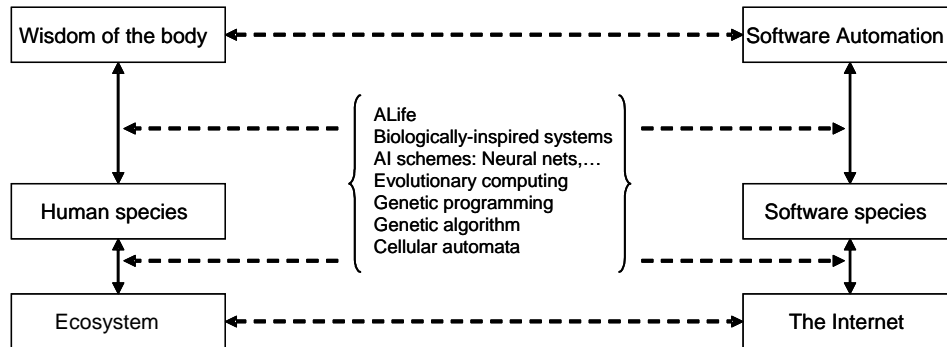


Figure 2: Basis of analogy between wisdom of the body and (wisdom of) software automation

As previously cited, the WOB is a term coined by Startling, a biologist, to express the set of all internal processes to maintain homeostasis in a living organism [Zajicek 2003]. Some authors call the WOB of an animal *instinct*. Part of the WOB of the human is called the *mind*. The WOB of an ameda is informed about the environment in which the organism lives by a receptor. The WOB adjusts its processes to cope with the environment change via a decision making scheme. It may respond with an action. The stimulus-action is more than a reflex. It involves a decision or selection of an action in an optimum manner.

From implications 1 to 3 above, software species can be equated partially as equivalent of human species. Thus the idea that software automation may be considered as parallel to Starling's wisdom of the body (WOB) seems plausible. After all, the WOB denotes the collection of all biological processes in a human body that operate optimally and control its well-being in health and in disease [Zajicek, 2003] just like the software automation is the collection of all software processes aiming at keeping the corresponding e-business operate optimally and control effectively its operating environment.

The analogy in Figure 2 incidentally suggests that many previous investigations on biologically-inspired systems such as cellular automata, genetic algorithm, genetic programming (collectively termed as evolutionary computing), AI reasoning schemes and their various realizations (neural nets, expert systems, decision systems, and the like), even ALife or autopoiesis theory as shown in the middle of Figure 2, may become the candidates for an adaptation and inclusion into our framework since each and everyone of them addresses one particular or more levels of organization which the software continuum concept embraces. In particular, systemic attributes such as fully-connectedness, self-organization and the like as characterized in complexity theory [Kauffman, 1996] and chaos theory [Lorenz, 1993], can be investigated within the scope of software automation in service of e-business on the Internet.

Implication 5 - The ecology of e-business with its software automation on the Internet from complexity theory perspective

As indicated in Figure 3, there are three properties of the proposed framework: *containment*, *fully-connectedness* and full *interface*. First, a certain level of organization is the container of the immediately lower level. Second, in the human body, the fully-connectedness and interface are in the body fluid.

Between cells in the human body, the fully-connectedness is realized by the extra-cellular fluid and blood plasma acting as communication channel [Alberts, et al, 1996]. The World Wide Web (WWW), and so does the Internet, has the characteristic of fully-connectedness, much like that of the human body [Zajicek, 2002].

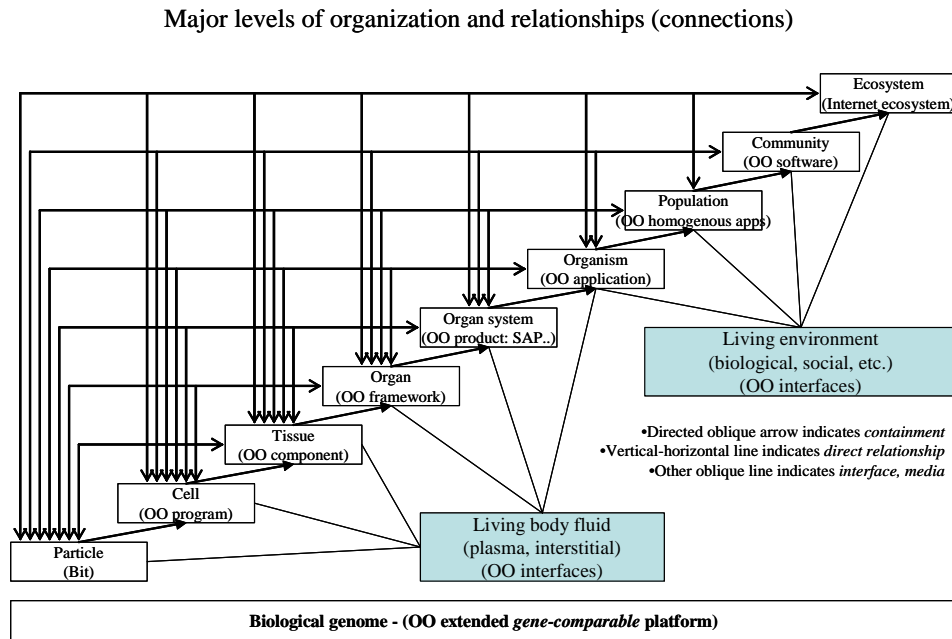


Figure 3: Biologically-inspired OO extended gene-comparable platform for software automation

The connectedness of the WWW is realized thanks to the original work of Tim Berners-Lee with his invention of HTML language. The web server site is a collection of files organized in some hierarchical directory (folders). These files are of different types: html, jpeg, gif, png, exe, script, xml, dtd, etc. Each of them can be linked to another via the hyperlinks under some architectural arrangements: everywhere linked, hierarchical, bus-like, star-like, or any combination thereof.

Due to fully-connectedness and full interface, the communication between the levels is not limited to the two adjacent levels, however (indicated by vertical-horizontal arrows). It involves all other levels. In fact, cell signaling involving hormones in an endocrine system is beyond its own container, a particular gland. It is known that hormone is a chemical messenger secreted into blood or extracellular (interstitial) fluid by one cell that affects the functioning of all other target cells. For a particular hormonal pathway, there are three possible actions: (1) *endocrine* action that impacts distant target cells, (2) *paracrine* action that diffuses to neighboring target cells and (3) *autocrine* action that acts on the same cell. The binding of hormonal receptors can be of *agonist* type (inducing events) or *antagonist* type (blocking the binding of the agonist). It is also known that blood concentration of glucose throughout the body must be higher than 1mg per ml. If it drops below that threshold, our neurons will misbehave. Our organs and organ systems fail and we ultimately are led to coma and death. A particular hormone will broadcast the change in glucose concentration to all cells to stimulate metabolic pathways to pull glucose concentration into its normal range to correct the concentration problem, following biological feedback mechanism (homeostasis of Walter Cannon).

The WWW is running on the Internet which in turn is a huge collection of computer networks that spans the world. The communication between any two computers is performed using a set of exchange rules of the TCP/IP protocol suite. The communication between two web sites is primarily governed by HTTP protocol over TCP/IP. The frames (OSI layer 1-2), packets (OSI layer 3), segments (layer 4), messages (layers 5-7) are under the OSI specifications for intercommunications. It is easy to create a new site and get connected to the Internet. The new site consist of web pages and other items such as Java applets,

scripted files, and the like act like new cells that are created in the human body. Some websites become obsolete or cease of existence, much like the cells that die and vanish. This characteristic of the Internet and its WWW meets the requirement of the WOB.

The WOB needs a memory to remember its past experience. The WOB of an animal has a special receptor called kin receptor for communications with others. The WOB of a human has a memory and other cognitive capabilities such as association and discrimination. These capabilities have to do with the fact that WOB as a set of processes is fully-connected and self-organizing.

The truth, however, is that the connectedness only occurs at the Internet and WWW, at the front-end of web applications and web-based applications (called internet portals). But work is still being done at the backend where connectivity between applications servers and database servers is only partial [Altman et al.]. An example helps clarify this point. Anyone can plan for a trip and may access an online reservation application on the Internet to request an airline ticket, car, hotel, and/or entertainment events but the reservation is actually performed separately at each selected participating server site. Different confirmations are given. If there any modification or cancellation to the intended trip, the user must go to each and every website where the reservation was previously made. The modification is not automatically carried out from airline to car to hotel to entertainment events as one would prefer. The simple reason is that the connections between these sites are rudimentary and occur only at the web server level (hyperlinks from one website to another with some parameter passing capabilities). Many still exist as islands of automation: automated update software, e-delivery of maintenance, EDI and the like. Many newer technologies emerge: XML, JMS, JCA, EJB, SOAP, WSDL, UDDI and related technologies are being applied and implemented for fully-connectedness at the backend where applications are to work together. Fully-connectedness must be reached for e-business software automation to work within the context of WOB.

Figure 3 appears to imply that by suggesting fully-connectedness, we might increase the complexity of OO paradigm unnecessarily. Actually, while waiting for the making of a genomic OO paradigm we only suggest that from Figure 3 if selected human functionalities are to be mimicked in software, the infrastructure of the intended software should observe the containment, fully-connectedness and interface properties. Operationally, the infrastructure should allow a communication mechanism that is, for example, hormone-like: (1) point-to-point, selected broadcast, full broadcast (current software has this capability and mechanism), and (2) with agonist/antagonist feature (not found in current software).

One possibility is to look at existing (vendor) software frameworks and application systems (application servers) such as PeopleSoft Internet, IBM Websphere Application Server, Sun J2SE/J2EE, Microsoft .NET, SAP R/3 and the like to leverage their technologies for robust autonomy and intelligence in applications. Each of these packages has its own web-enabled graphical user interface (GUI), kernel, application server, database server, metadata repository and development tools. Each has its own infrastructure and open architecture with its own set of API (application programming interface). Each allows enterprises to build or customize their Customer Relationship Management (CRM), Supply Chain Management (SCM), Sales Force Automation and other enterprise applications.

Their architectures are commonly Internet-based, server-centric in 3-tier or n-tier client/server paradigm. Their portals provide excellent interface to the outside world. Their database server is the engine for information/data flow similar to the heart of an organism that pumps blood through its body. Their metadata repository and directory server act like partial memory. Their application servers are architected to manipulate messages (similar to digestive system), to handle business/application logic (partially as a brain) among other things. However, the facilities to build robust autonomy and robust intelligence in enterprise e-business software are not the focus of these packages, and therefore they are practically nonexistent. Furthermore, the cost of these packages is in the 6-figure dollar range for installation and customization alone. What we are thinking of is the use of Sun J2SE/J2EE as free software for formulating our biologically-inspired framework.

The software continuum concepts also gives rise to other important systemic properties: *emergence* and *self-organization* (Ashby, von Foerster, I. Prigogine) as observed in natural and biological systems and

extensively studied by numerous researchers in the context of complexity theory (by Stuart Kauffman), chaos theory (coined by Edwards Lorenz, B. Mandelbrot) at all levels of organization. The concept offers links to Starling's concept of wisdom of the body (also studied by Walter Cannon and Sherwin Nuland). The systemic features are addressed in a separate paper.

5. The software continuum (prototypical) framework in support of ecology of e-business

Our proposed process addresses the development of software organism, patterned after human development as follows:

- Embryonic development from software conception
- Nature-based prenatal development of software fetus
- Nurture-based postnatal development of newborn software in its infancy
- Learning-based postnatal development of software childhood, adolescence and adulthood

Our proposed framework is shown in Figure 4. It needs some explanation. The block diagram does not appear "biotic" much less human-like. It consists of three main functionalities: development tools, the main biologically-inspired framework, and the leverage of existing packages. The main biologically-inspired J2SE/J2EE-based framework itself has four components: the web server, the application server framework, applications and the knowledge data bases. The web server primarily acts as the sensory system to receive inputs from user's web browser. Input data or information are structured in XML format if necessary with compiled session EJBs mimicking the functionalities of the sensory receptors.

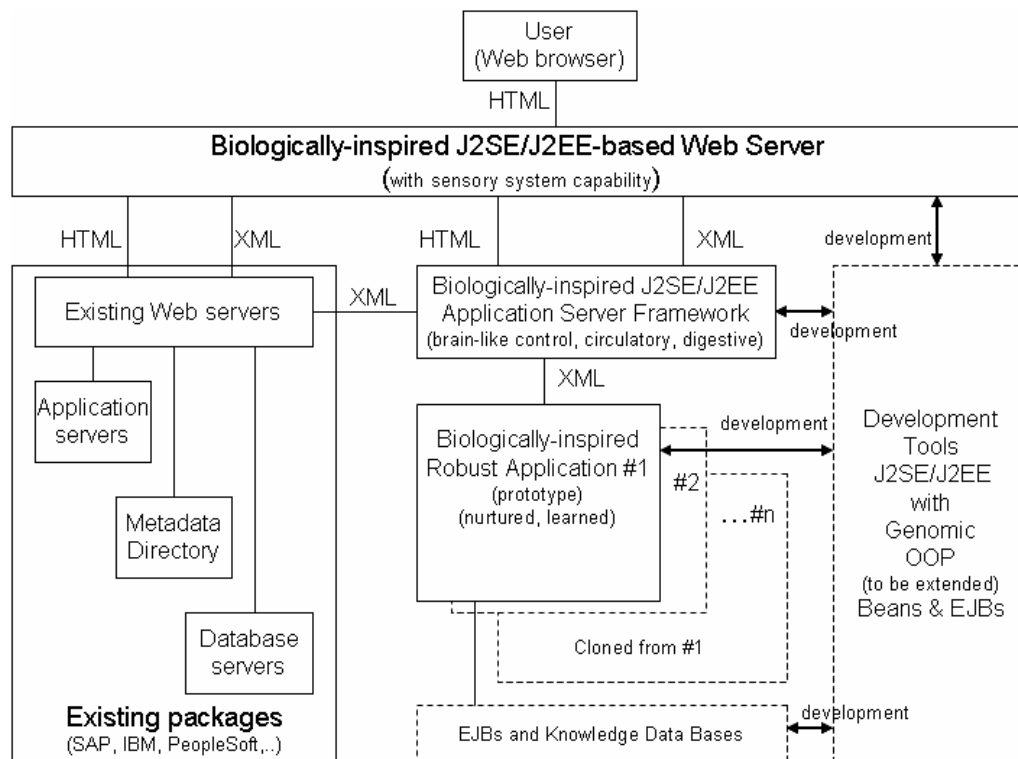


Figure 5: Biologically-inspired framework for software automation

The session EJBs interact with entity EJBs to restructure data fields in various XML structures are sent to the application servers or knowledge database system. The knowledge database systems act like working memory (rather than long-term memory) since their information most likely is integrated at execution time by the application server. The application server has an endocrine-like control system that

work with an ANN for regulating the flow of information and data. Information and data are manipulated by a digestion-like system in the application server. The communication of usable and relevant information and data is achieved by a circulation-like system. The application server in effect serves as a means for transforming and distributing them according to the endocrine-like control system. There is one collection of web server and application server for the intended e-business software automation. The last piece of functionality is the e-business robust application (numbered as #1, #2 ... #n). Each application goes through a nurturing and learning period to become autonomous and/or intelligent. Each might be a simple reflex automation or might associate with an ANN adapting Hebb's cell-assembly model.

As previously discussed in section 2, making OOP genomic (Development tools box in the right side of Figure 4) would be of great importance to the model since it allows cost-effective development of applications. However, genomic OOP (such as genomic Java) is extremely complex, costly and long-term. Meanwhile, we can take advantage of other existing OO packages used in bioinformatics and related disciplines such as genomic Perl. In fact, besides genomic Perl, there are already available OOP packages with the capability to provide biological computation. An example is JGAP version 1 (Java Genetics Algorithms Package) available as freeware to offer basic genetic mechanisms that can apply evolutionary principles to problem solutions. It consists of Java classes such as *gene*, *chromosome*, *genotype* and *FitnessFunctions* and Java interfaces such as *CrossoverOperator*, *MutationOperator* and *ReproductionOperator*. The package implements "Genetic algorithms (GAs) as evolutionary algorithms that use the principle of natural selection to evolve a set of solutions toward an optimum solution" [JGAP, 2004]. Other packages include Bio-Java or neural nets with Java. In addition, software in support of bioinformatics (genomics, proteomics) is readily available and grows relatively fast. These packages offer class hierarchy and facilities that can be partially modified and/or extended for use in this biologically-inspired framework, even though our research is in the opposite direction of bioinformatics. Also, the architectural work by Christopher Alexander which has been widely incorporated in OOAD (object-oriented analysis and design) offers many design patterns in OOP that are (already) nature-inspired.

The software continuum concept appears to offer the possibility to tie together some previous research ideas (metaphorical, analogous, and/or mathematical) that have been conducted and applied in isolation as argued previously. This allows us to take advantage of the previous formulations, realizations and applications.

6. Formulation of potential applications

Internet e-business software automation as ecosystem

In this subsection, we consider a special, terminal case of the software continuum concept where the Internet e-business software automation is viewed as an ecosystem. Within the biologically-inspired framework, such an Internet ecosystem involves e-business applications. Each application is an instance of a certain e-business application "species". Each acts and behaves as an organism. Some are intelligent and/or autonomous, others are not. Some are leaders. Others are supporters. Some are conglomerate predators, others are prey. Some are computer viruses and others are worms which attempt to intrude and damage the Internet-based information or other organisms "living" in the Internet. The following discusses in some details the Internet e-business software automation as an ecosystem.

Microscopically, the messages that travel the Internet between e-business software applications are actually chunks of streams of bits (zeros and ones) in different data structures and formats. At the sending or receiving end, the bit streams are assembled in the form of HTML pages, XML documents or raw format (non-XML/HTML) of data using protocols such as HTTP, IIOP, etc. Originated and created as a request from the sender, a particular bit stream represents a series of headers of the TCP/IP protocol suite (MAC, LLC, IP, TCP and upper layer data). The stream is sent over the communication media to be received at the requested server. The stream is switched and routed as it is floating on the Internet to reach its destination.

Billions of such bit streams originated from million of users are created and processed among different servers, or extracted from files or databases on the Internet with HTTP or other protocols over TCP/IP

[Tang, 1992]. They are exchanged between web programs (the web browser and the web servers, application servers) and its web-based supporting infrastructure (data base servers, systems software, firmware and hardware). This, in some fashion, is similar to the billions of particles (electrons and protons) that are created, shared, or exchanged among different atoms, molecules or macromolecules in every part of an organism.

User messages are transformed into different formats following the implementation of the OSI reference model [Comer et al., 1994]. The requests in terms of single *message* or *transactions* at the application and presentation layers (layers 7 and 6, applications and processes) are padded with control information for session layer (layer 5, hosts and ports). They are transcribed into *segments* at the transport layer (layer 4, routers and network control programs) and into *packets* at the network layer (layer 3). Conversion and additional information are padded to the binary *frames* at the link layer and at the physical layer (network interface card) for transportation across the Internet from the sending end. The reverse process from layer 1 to layer 7 occurs at the receiving end. The frames (at OSI layers 1 and 2), packets (layers 2 and 3), segments (layers 3 and 4), messages and transactions (layers 5-7) are created, shared, exchanged or modified by trillions of algorithms at many different hosts, funneled to different servers to allow users to see the requested web pages and interact. These operations are similar to macromolecules that are transcribed from genes of DNA to messenger RNAs (mRNAs), and translated into thousands of proteins in the biological cells transported by blood circulation in support of different tissues, organs and organ systems of an organism [Alberts et al. 1998].

The automated software collection supporting a request on the web may include, to name a few, a web browser, one or more search engines with CGI, ASP or JSP scripted routines including Java applets, other web servers, application servers, message brokers, database connections and databases [Linthicum, 2000]. These processes may involve shared memory, named pipes, API calls over RPC, DCE, CORBA-compliant ORB, RMI-IIOP mechanisms [Zahavi et al., 1999], or message queuing [Blakely et al., 1995]. They possibly include the interaction between Enterprise JavaBeans, Servlets, Java Server Pages, [Picon et al. 2000], enterprise software components (e.g. EJB containers) or commercial, shrink-wrapped server packages (e.g. SAP R3 or Peoplesoft) in support of previously built applications and web services [Armstrong et al, 2002]) and/or newly created systems and applications over the Internet. The messages among them may be in the form of web client pages, documents with texts and graphics. They may include sound and/or video.

The software processes are in a way similar to the trillions of chemical reactions occurred in the genes of the double-helix DNA chain to synthesize proteins, breakdown carbohydrates, or lipids [Fox, 1996] for a human body. These reactions are to keep a human body alive, keep the brain alert, help digest the food, allow the limbs coordinated. The proteins influence the structures of different cells and tissues, passively (such as the formation of collagen to give strength in the finger tendons and ligaments) or actively (as found in receptors for hormones or in transport carriers, e.g. hemoglobin in the blood). They allow the human to do physical or mental work in an ever-ending cycle.

The chemical reactions occur at the intracellular and intercellular level for the gas exchange between air and blood in the lungs and between the blood cells and tissues of the body via the capillaries [Fox 1996]. These reactions trigger the process of cell respiration. This in turn involves chemical energy for distribution of oxygen and transportation of nutrients and proteins across all tissues, organs and organ systems within the body. They produce the kind of external work expected (e.g. translating a thought into the typing of characters on the computer by the fingers) and the internal work that the body requires. The myriad of chemical reactions involves the release of energy by the breakdown of molecules (glucose and others) in a process called catabolism and in the acquisition of energy to produce other molecules in the process of anabolism. Both thus make up the metabolism in the body [Solomon et al. 1996]. Similarly, all the processes on the Internet are to keep the business, scientific and industrial applications and systems connected, alive, well behaved and communicated much like the biological processes in a human body are to keep the body healthy and alive. In addition, the trillions of chemical reactions keep the whole body continue to be in its homeostasis and therefore healthy. They also prevent the body from contracting disease [Cotran et al., 1994] caused by any viruses (the immune system) from the breathing or any bacteria from the consumption of food (antibodies).

The Internet carries message initiated at one end (e.g. client request at the web browser) to arrive at another end (received at the server). It may involve a response to be generated at the server. The human body carries message initiated at one end (e.g. the sensory receptors) to reach another end (e.g. the central nervous system or CNS that includes the brain and the spinal cord) [Fox, 1996]. The message may hop between many different servers before it reaches the end-point server as commonly occurred on the Internet. Each server program may be thought of as analogous to a neuron, the brain cell. Each neuron consists of a cell body (similar to an executable program), an axon (similar to output) and dendrites (inputs). The interface between two server programs is similar to the synapse where the long axon of one neuron meets the dendrites of another. Most likely, the communication involves many single messages carried from one server to the next to be combined in a proper sequence at the receiving server. This is somewhat analogous to the action potential needed for transmission of neural information as the sum of many signals generated at multiple sending neurons to one particular synapse according to McCullough and Pitts's model.

The metaphorical and analogous observations presented above suggest that diverse e-business applications running on the Internet can be grossly considered as analogous to organisms. A part of the Internet where similar e-business applications are concerned can be viewed as population ecology. Others involve different and diverse e-businesses applications constitute a community ecology or ecosystem.

7. Concluding remarks and future work

From the idea that the Internet can be considered as ecosystem, we argued that there exists a software continuum similar to the natural continuum. We attempted to establish a general parallelism between the e-business software over the Internet and human species. We also argued that e-business software automation over the Internet has the characteristics of the wisdom of the body. While much effort of the industry and academia is currently devoted to the e-business integration and enterprise application integration (architectures, processes, and tools) which will eventually assure to some degree the fully connectedness characteristic of e-business software automation, we claim that to reach robust autonomy and robust intelligence, e-business software automation should be structured to allow self-organization and emergence. We observed that current e-business software development process hinders this goal of autonomy and intelligence since for the most part it still follows a top-down development approach. We proposed that e-business software automation should follow an embryonic approach for growth and development that consists of four stages: conceptualization, patterning, differentiating and growth.

Considering software as living species within the software continuum from bit streams to business ecosystems as illustrated in section 3 allows us to look at software further from different aspects: anatomical, genetic, physiological, metabolic and ecological [Nguyen, 2003]. We can mimic these biological and ecological processes to build capabilities in software that behaves in many respects similar to human species. Some of these capabilities have been conceived and realized as the result of a partial similarities between human species and hardware/software (analogously and metaphorically) as found in computer vision, speech, robotics (nanotechnology, microtechnology, symbolic-numeric integration, task-directed multifingered hand grasping path planning, and the like), computer chip manufacturing, economic modeling, etc.).

Our software continuum concept offers a view of the Internet as an organized, linked architecture from bits to Internet ecosystems much like the natural continuum from particles to ecosystems, instead of a partial, isolated or fragmented view. It offers plenty of new insights drawn from nature and previous accomplishments for applications into organization, infrastructure, integration, collaboration and interoperability of its constituents as complex adaptive systems (Capra), dissipative systems (Prigogine) and/or open systems (von Bertalanffy). Furthermore, the software continuum concept offers links to previous work on system features such as fully-connectedness, self-organization, emergence that have been discussed and treated in the context of complexity theory [Kauffman 1995], chaos systems [Lorenz 1993, Mandelbrot 1983], general system theory [von Bertalanffy 1966], web of life [Fritjof Capra 1966], and scale-free network [Barabasi 2004]. While features such as self-reproduction (sexual or asexual) are

far fetched, other self-ness” features such as self-replication, and/or self-healing/self-repair appear to be potentially addressable by the framework. It also offers opportunities to look at the Internet from other aspects, e.g. digital economy [Margherio et al. 2000]. If appropriately modeled, various issues of digital economy can be addressed in the future.

8. References

- Alberts, Bruce, Dennis Bray, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts and Peter Walter (1998), *Essential Cell Biology*, Garland Publishing.
- Alexander, Christopher. 1997. *The Timeless Way of Building*. Oxford University Press. 1997.
- Alexander, Christopher. 2003. *New Concepts in Complexity Theory*. online.
- Altman R., J. Hill, J. Klein, B. Lheureux, Y. Natis, M. Pezzini, R. Schulte, S. Varma (1999). *Middleware: The Glue for Modern Applications*, AIMS Strategic Analysis Report, R-08-2601, Gartner Group, 1999
- Armstrong E., S. Bodoff, D. Carson, M. Fisher, D. Green and K. Haase, *The Java Web Services Tutorial*, Sun Microsystems, 2002
- Barabasi, Albert-Laszlo and Zoltan N. Oltvai. 2004. *Nature Review: Genetics*. Vol 5. February 2004.
- Blakeley, Bernie, Harry Harris and Rhys Lewis (1995). *Messaging and Queuing Using the MQI*, McGrawHill
- Boar, Bernard H. *Constructing blueprints for Enterprise IT Architectures*, John Wiley & Sons, 1999
- Cannon, Walter B., (1932). *The Wisdom of the Body*, Norton and Company, 1932.
- Carnegie Collection. 2003. online, <http://nmhm.washingtondc.museum/collections/hdac/anatomy.htm>
- Cisco: Internet Ecosystem, http://business.cisco.com/prod/tree.taf%3Fpublic_view=true&kbns=1&asset_id=83098.html
- Cohen, Geoff. *Biologically inspired software: Metaphor, mechanism and meaning*. [http://www.cbi.cgey.com/pub/bi-news/pdf/php0lw0UU\\$.pdf](http://www.cbi.cgey.com/pub/bi-news/pdf/php0lw0UU$.pdf)
- Comer, Douglas E. and David L. Stevens, *Internetworking with TCP/IP*, Vol. III, Prentice-Hall, 1994
- Conklin C., R. Watkins, Dr. S. Foo, Dr. R Perry, Dr. R. Roberts, *Biologically-inspired Vision Algorithms*, ?
- Cotran, Ramzi, Vinay Kumar and Stanley L. Robbins, *Pathologic Basis of Disease*, 5th edition, W.B. Saunders Company
- Davydov, Mark. (2001). *Corporate Portals and e-Business Integration*, McGraw-Hill, 2001
- Dini, Paolo and Andrea Nicolai. 2003. *D.B.E. The Digital Business Ecosystem*. On line, www.digital-ecosystem.net.
- Diorio, Chris, David Hsu and Miguel Figueroa, *Adaptive CMOS: From biological Inspiration to Systems-on-a-Chip*, *Proceedings of the IEEE*, Vol. 90, no. 3, March 2002
- Farmer, J. Doyne (2000). *Market Force, Ecology, and Evolution*, *Santa Fe Institute*.
- Feigenbaum, Edward (1990). *The Age of Intelligent Machines: Knowledge Processing – From File Servers to Knowledge Servers*, online, www.kurzwei.net/articles/art0098.html
- Flake, Gary William, Steve Lawrence, C. Lee Giles, Frans M. Coetzee, *Self-Organization and Identification of Web Communities*. 2002. On line, <http://webselforganization.com/>
- Fox, Stuart Ira (1996). *Human Physiology*. Wm.C. Brown Publishers.
- Geisendorf, Sylvie. (2003). *Genetic Algorithms in Resource Economic Models*. Online
- Gillooly, Caryn. (1999). *E-Business Management: Solving the Next Management Challenge*. *Hurwitz BalancedView Report.*, Hurwitz Group.
- Goldstein, E. Bruce (1984). *Sensation and Perception*, 2nd edition, Wadsworth Publishing Company.
- Harris, J. S. *The Brain and Computational Architectures*, luciano.stanford.edu/~morf/genee/biologic/biologic.htm, 2001
- Hickman James J., Frederica Darema, and W. Richard Adrion, *Biological Computation: How does biology do information technology?* NSF Report, January 2001
- Hillis, W. D. 1990. *Co-evolving parasites improve simulated evolution as an optimization procedure*. *Physica D.*, Vol. 42.
- Holland, J. H. *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, MA (First edition, 1975, second edition, 1992)
- Hughes, K. (1993). *Entering the World-Wide Web: A Guide to Cyberspace* [Online]. <http://www.hcc.hawaii.edu/guide/www.guide.html> (1998, October 20).
- Iansiti, Marco and Roy Levien. 2004. *Strategy as Ecology*. *Harvard Business Review* March 2004. Online. www.hbr.org

- Jog, Ninad (2002). The Epistemology of Digital Organisms, online, www.mindspring.com/~ninad/alife/Epistemology.htm
- Kauffman, Stuart. 1996. Investigations: The Nature of Autonomous Agents and The Worlds They Mutually Create, online, <http://www.santafe.edu/sfi/People/kauffman/Investigations.html>, accessed 10/12/2003
- Kelly, K. 1994. Cracking Wall street. Wired, July 1994
- Kephart, Jeffrey O, Gregory B. Sorkin, David M. Chess and Steve R. White, Fighting Computer Viruses, Scientific American, Nov 1997
- Kimball, online. (2003) users.rcn.com/jkimball.ma.ultranet/BiologyPages/E/EmbryonicDevelopment.html, accessed 7/28/2003
- Klyce, Brig. What is Life?, online, www.panspermia.org/whatis2.htm
- Koza, John R., Martin A. Keane, Jessen Yu, Forrest H Bennett III, and William Mydlowec (2000), Automatic Creation of Human-Competitive Programs and Controllers by Means of Genetic Programming, Genetic Programming and Evolvable Machines, 1, 121-164, 2000, Kluwer Academic Publishers
- Koza, John R., J. P. Rice and J. Roughgarden. 1992. Evolution of food foraging strategies for the Caribbean anolis lizard using genetic programming. Adaptive Behavior. Vol. 1 no. 2.
- Langton, CG. Editor (1995). Artificial Life: An Overview, MIT Press.
- Lahoz-Beltra, R and Vieri Di Paola (2002). Towards a computational view of cell: do cells bear a resemblance with computers?, <http://www.mdpi.net/papers/fis2002/121/LahozDOC.htm>
- Leonard, Allenna and Stafford Beer. (1994). The Systems Perspective: Methods and Models for the Future AC/UNU Millennium Project.
- Library of Congress, (2005) <http://memory.loc.gov/ammem/award97/icuhtml/aepsp2.html>
- Linthicum, David (2000), Enterprise Application Integration, Addison Wesley.
- Linthicum, David. (2001). B2B Application Integration: e-Business-enable Your Enterprise, Addison-Wesley, 2001.
- Lorenz, Edward. (1993). The Essence of Chaos, University of Washington Press.
- Lucas, Chris. Self-Organization FAQ, 1997. online, <http://psoup.math.wisc.edu/archive/sosfaq.html>
- Manson, Steven M. (2001). Simplifying complexity: a review of complexity theory. Geoforum 32.
- Margherio, Lynn, Dave Henry, Sandra Cooke, Sabrina Montes (2000). The Emerging Digital Economy, Department of Commerce.
- MIT Lab, <http://dl.media.mit.edu>, 2001
- MIT Media Lab, <http://agents.media.mit.edu>
- Mitchell, Melanie, Computer Models of Complex Adaptive Systems, New Scientist, 1993
- Mitchell, Melanie and Stephanie Forrest, Genetic algorithms and artificial life, Santa Fe Institute workshop, 1993
- Mitchell, Melanie (1998). An introduction to Genetic Algorithms, MIT Press.
- Mitchell, Melanie (1999). Evolutionary Computation: An Overview, Annual Review of Ecology and Systematics, 1999, Vol. 20: 593-616
- Mitchell, Melanie (2000). Life and Evolution in Computers, Santa Fe Institute.
- Mitchell, Melanie (2002). Is The Universe a Universal Computer? Science, vol. 298, October 2002
- Moore, James (1993), Predators and Prey: A New Ecology of Competition, Harvard Business Review, May-June 1993.
- Moore, James (1996), The Death of Competition, Harvard Business Review, May-June 1993.
- National Science Foundation, BITS 01-102 Program Announcement, January 2001
- Nguyen, Thang N. (2003). An Embryonic Approach to Object-Oriented Software Development, Proceedings of 34th Decision Sciences Institute Conference, Washington D.C. November, 2003
- Nguyen, Thang N. (2002a). A Biologically-inspired Basis of E-business Software: From cellular level to ecological level, Proceedings of the 9th International Conference on Neural Information Processing - ICONIP, Singapore. November, 2002
- Nguyen, Thang N. (2002b). Anthropomorphic Basis of E-Business Software Integration and Interoperability, Proceedings of Americas Conference on Information Systems - AMCIS, Dallas. August, 2002
- Nguyen, Thang N. (2002c). The Ecology of Software: A Framework for the Investigation of Business-IT Integration. Proceedings of IRMA 2002, Seattle, May 2002
- Nguyen, Thang N. (2001). The Ecology of Software, 6th International Conference of Decision Science Institute, Chihuahua, Mexico, July, 2001

- Nguyen, Thang N. (2000). *An Ecological Approach to E-Business*, Proc of Americas Conference on Information Systems (AMCIS), Long Beach, August 2000
- Nuland, Sherwin B. 1997. *The Wisdom of the Body*. Random House. 1997.
- Petrucci, R. and W. Harwood, General Chemistry: Principles and Modern Applications, 6th edition, MacMillan Publishing Co., 1993
- Picon, J., R. Coqueret, A. Hutfless, G. Indurkha and M. Weiss, *Design and Implement Servlets, JSPs and EJBs for IBM Websphere Application Server*, SG24-5754-00, 2000
- Ray, Thomas S. (1993). *An Evolutionary Approach to Synthetic Biology, Zen and The Art of Creating Life*, online, <http://darplace.tomrad.ro/file.PDF>
- Rowley, Kurt. (1995). *Understanding Software Interoperability in a Technology-Supported System of Education. Cause/Effect*, Fall 1995
- Sole, Ricard and Romualdo Pastor-Satorras, *Complex Network in Genomics and Proteomics*, www.santafe.edu/sfi/publ.pdf, 2002
- Schneider, Gary (2002). *Electronic Commerce*, 3rd edition, Thompson Course Technology, 2002
- Shotwell, Allen. (2001). *Biology in Computer System*, online, <http://ivytech7.cc.in.us/mathsci/alife/computers.PDF>
- Solomon, Eldra, Lynda Berg, Diana Martin and Claude Villee. ((1996). *Biology*, Saunders College Publishing.
- Thrift, Nigel. (2001). *The Automatic Production of Space*. Centre for Science Studies, Lancaster University.
- Tang, Adrian, and Sophia Scoggins (1992). *Open Networking with OSI*. Prentice-Hall.
- University of California, Berkeley (1997). *What is the Internet, the WWW, and Netscape? An Introduction*. [Online]. <http://www.lib.berkeley.edu/TeachingLib/Guides/Internet/WhatIs.html>
- Visible embryo. (2003) online www.visembryo.com/baby/index.html
- Von Burtalanffy, Ludwig (1969), *General System Theory*, George Braziller
- Von Neumann, John. 1958. *The Computer and The Brain*, Yale University Press. 1958
- Web Basics – Research Zone. <http://www.classzone.com/research/pages/basics/res07.htm>
- Zahavi, Ron and Linthicum, David. (1999). *Enterprise Application Integration with CORBA Component and Web-Based Solutions*, John Wiley & Sons.
- Zajicek G.(1995). *The language of the wisdom of the body*, *The Cancer Journal*, Volume 8, no. 6 (Nov-Dec 1995).
- Zajicek G.(2002). *Cancer and Wisdom of the Body*, Business Briefing: Global Healthcare, 2002.
- Zajicek G.(2003). *WOB as a non-linear massively parallel biological computer*, online, www.what-is-cancer.com/papers/straming/wobcomputer.htm, accessed 2003