December 2002

ANTHROPOMORPHIC BASIS OF E-BUSINESS SOFTWARE FOR INTEGRATION AND INTEROPERABILITY

Thang Nguyen
California State University Long Beach

ANTHROPOMORPHIC BASIS OF E-BUSINESS SOFTWARE FOR INTEGRATION AND INTEROPERABILITY

Thang N. Nguyen
California State University Long Beach (tnnguyen@csulb.edu)
Candle Corporation (thang_nguyen@candle.com)

Abstract

From the notion that e-business software is considered as a living species, we investigate the anatomical, genetic and metabolic basis of software supporting e-commerce/e-business integration and interoperability.

E-Business Software as Living Species

As we speak, the trillions of bits (zeros and ones) in different data structures and streams are floating on the Internet and on its supporting infrastructure. They are acquired, shared, exchanged or modified by trillions of algorithms to keep business, scientific and industrial applications and systems alive and connected. This is in some fashion similar to the trillions of particles (electrons and protons) in the human body that are acquired, shared, or exchanged among different atoms, molecules or macromolecules. The trillions of chemical reactions on the latter in turn provide proteins, fatty acids, sugars and nucleotides to keep the human body active and healthy and to do physical or mental work. Thus, e-business software may be considered as a human body, i.e. living species.

Anthropomorphic Basis of E-Business Software

The anthropomorphic basis of e-business software is investigated from the following aspects: anatomical, genetic and metabolic. In the following, we argue that cells in living organisms and class objects used in e-business software possess striking similarities anatomically, genetically and metabolically.

Anatomical and Genetic Basis of E-Business Software

There are hundreds of millions of different cell types in living species just like there are hundreds of millions of object classes used in existing OO applications and systems. Cells are made of a half a dozen of atoms (H, C, N, O, S, P) and a few other elements (sodium, magnesium, chlorine, potassium, calcium) while class object attributes are made of half a dozen of primitive data types (bit, character, integer, float, double, long, short). These atoms form molecules (e.g. carbon compounds) and similarly the primitive data types form more complex data types in data members of object class (Figure 2). All living cells have a similar basic chemistry. In all of them, cells consist of small molecules of four major families, among others: sugars as subunits of polysaccharides, fatty acids as components of cell membranes, amino acids as subunits of proteins and nucleotides as subunits of DNA/RNA (deoxyribonucleic acids and ribonucleic acids respectively). Small molecules can be used as monomers to construct giant macromolecules such as proteins, nucleic acids, lipids and polysaccharides (Solomon et al., 1996). Similarly, all class objects have simple programming constructs such as if-statement and "assignment" statement) that build complex member functions. The simple constructs can be thought of as biologically equivalent elements of α–helix and β sheet of amino acids.

Genes are written in the same chemical code, and constructed out of the same chemical building blocks. Proteins produced from DNA instructions are based on the same 20 different types of amino acids. These amino acids are building blocks of proteins, occur over and over again in all proteins from bacteria to animals and humans (Albert et al, 1998). The programming constructs such as *if*-statement, *while*-statement, *dowhile*-statement in a software program segment have a role similar to amino acids in

Particles	Bits
Atoms	Primitive data
	(e.g. char, int, float, ...)
Molecules	Complex data
	(String, struct,...)
Monomers	Fields
Polymers,	Records, lists, arrays,
macromolecules	XML structures
Genome	Firmware instructions
	(=, conditional, nested,..)
Amino acids	Programming constructs
	(if-then, for-,
	while-, do-while, …)
Proteins	Active algorithms
(hemoglobin,..)	*(transport RPC,..)*
DNA/RNA	Object constructor methods
Organelles	Object class methods
Cells	Object classes
Tissues	Components
	(COM/DCOM, beans..)
Bacteria, virus	Computer virus
Organs	Applications
	(vertical apps,..)
Organ systems	Application systems
	(domain-specific:
	e.g. manufacturing,
	financial, ..)
Organisms	Software products
Species	Software

Figure 1. General Parallelism Between Ecology and E-Business

humans. DNA/RNA and proteins that are sequenced (primary), folded (secondary), 3-dimensional (tertiary) are the biological equivalent of the constructor functions (DNA/RNA) and active algorithms (proteins) of class object. The latter are formed by programming constructs such as *assignment* statement and conditional statement (sequenced), *while*-statement, and *dowhile* statement (folded), nested-*if* and nested logic statements (multi-dimensional). They manipulate and transform data members, break down complex data members, and form other class objects. Like amino acids, the programming constructs occur over and over again in all algorithms. By peptide bond, amino acids chain together to form polypeptide that is vital to the function of the proteins while programming constructs chain together to form member functions in OO methods. One particular type of methods is called constructor method. This method carries the information on the knowledge of the object (e.g. data members) and on how the object is created or generated in a fashion similar to the biological information found in nucleotides. The nucleotides are the building blocks of nucleic acids that exist as DNA or RNA. DNA/RNA and proteins are building blocks from which a cell is constructed and determined thousands of distinct functions of the cell. These macromolecules are made from a set of polymers formed in a particular order called sequence much like the set of programming constructs coded in methods in a particular order to reflect the program logic.
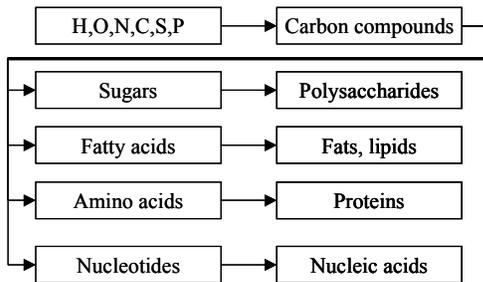

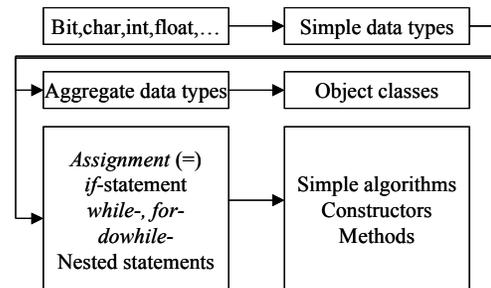**Figure 2a. Anatomical Basis of Cells**

Figure 2b. Anatomical Basis of Object Classes

Metabolic Basis of E-Business Software

Sugars and fatty acids store *energy* that is used in living cells. Energy can be acquired or released. We consider *information* (inferred from data) is the biological equivalence of energy for a class object. Information can be stored or retrieved. Without information in data members of class objects that are instantiated, objects are idle and have no life. Similarly, without energy acquired from food, cells will die.

Living organisms such as humans need *energy* to carry out many chemical reactions that occur in cells and between cells. These reactions are connected in series so that the product (result) of one reaction becomes the substrate for the next. This constitutes the metabolic pathway in cells (Fox, 1996). This is somewhat similar to the *information* (data) needed for the execution of a OO
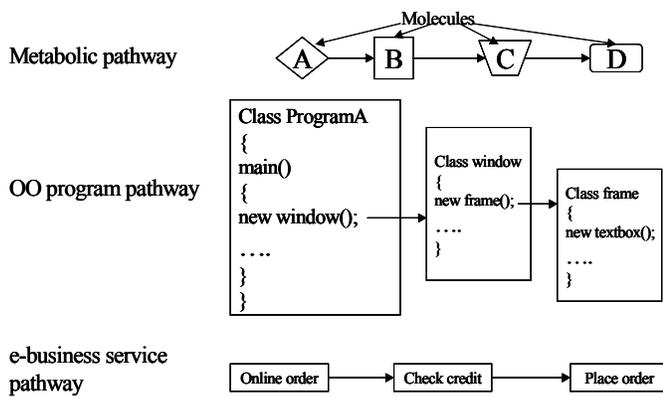
Figure 3. Equivalent Pathways

program (C++ or Java) in which an object instantiated from a class (e.g. from main() in turn creates (by the operator *new*) or instantiates another object in the chain (e.g. window()) as shown in Figure 3. The window object then instantiates another object from the class of textbox, and the like. This constitutes an OO program execution pathway.

From a higher level, e-business service (e.g. online order) may be considered as a series of steps that involves (1) taking an order at a program running under a web browser in Los Angeles, (2) checking buyer's credit at another program running in a server in San Francisco, (3) placing the order by a third program in Chicago, (4) produce and distribute the order by a fourth program in Atlanta, and (5) finally performing the billing of the order in New York in a B2B setting. The latter constitutes the e-business service pathway provided by the software product automation.

Concluding Remarks: E-Business Integration and Software Interoperability Using an Anthropomorphic Approach

The parallelism between e-business software and living species is generally established anatomically, genetically (in the sense of Mitchell, 2000), and metabolically in the previous sections. If we include the ecological aspect in the sense of Moore (1993), this will constitute a parallelism between ecology and e-business (Nguyen, 2002). Although the parallelism appears general and superficial, it gives rise to the potential applicability of laws and information technology processes (to be identified in natural bio-ecological systems) to e-business systems from cellular level to ecological level. One particular major issue we are interested in is the e-business integration and interoperability that have become very critical in today's computing environment (Linthicum, 2001), and that remain as a collection of isolated, disconnected or fragmented information systems within an enterprise or across enterprises, with serious mismatches and gaps exist between the business part (e.g. high-level business strategies) and the supporting IT part (e.g. low-level IT operations). In fact, in a particular enterprise, departmental units are not readily and totally connected to others. Users have to log in and access many different and diverse enterprise applications for needed information and changes (e.g. Web applications, Lotus Notes databases, Relational DBMS, mainframe transaction systems). The integration in making the e-business applications work together in EAI has been primarily patchwork because e-business integration, horizontally or vertically, is complex, difficult, expensive and very time-consuming. Since *information* is considered as biologically equivalent to *energy*, we hope to explore the laws of thermodynamics, complexity theory and information theory that govern biological objects for applicability to e-business software integration and interoperability. If this is understood, we will extend our investigation to how cellular and pathologic basis of injury and disease in humans might help understand the basis of e-crisis and response.

References

Alberts, Bruce, Dennis Bray, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts and Peter Walter (1998), *Essential Cell Biology,* Garland Publishing, 1998.

Fox, Stuart Ira (1996). *Human Physiology.* Wm.C. Brown Publishers, 1996.

Linthicum, David. (2001). *B2B Application Integration: e-Business-enable Your Enterprise,* Addison-Wesley, 2001.

Mitchell, Melanie (2000). "Life and Evolution in Computers", *Santa Fe Institute,* 2000.

Moore, James (1993), "Predators and Prey: A New Ecology of Competition", *Harvard Business Review,* May-June 1993.

Nguyen, Thang (2002). "The Ecology of Software: A Framework for the Investigation of Business-IT Integration", *Proc. IRMA (Information Resources Management Association)* 2002, May, Seattle, Washington, 2002

Solomon, Eldra, Lynda Berg, Diana Martin and Claude Villee. ((1996). *Biology,* Saunders College Publishing, 1996.