Proceedings

Lessons from Biology: Genes, Neurons, Neocortex and the New Computing Model for Cognitive Information Technologies †

Rao Mikkilineni [1,*]

[1] C^3DNA; rao@c3dna.com
* Correspondence: rao@c3dna.com; Tel.: +1-408-406-7639

† Presented at the IS4SI 2017 Summit DIGITALISATION FOR A SUSTAINABLE SOCIETY, Gothenburg, Sweden, 12-16 June 2017.

Published:9 June 2017

Abstract: In this paper we analyze our current understanding of genes, neurons and the neocortex and draw a parallel to current implementations of cognitive computing in Silicon. We argue that current information technologies have evolved from the original stored program control architecture implementing the Turing machines which allowed us to model, configure, monitor and control any physical system using a Universal Turing Machine model. However, large scale of distributed computations and their tolerance requirement to fluctuations have created new challenges. We suggest that a new agent based computing model extension to the current Turing machine meets this challenge and provides a path to cognitive self-learning and self-managing systems.

Keywords: Genes; Neurons; Neocortex; Cognition; DIME Network Architecture; Self-managing computations; Computing models.

1. Introduction

According to Ray Kurzweil [1] the story of human intelligence starts with a universe that is capable of encoding information. This was the enabling factor that allowed evolution to take place. The story of evolution, he observes, unfolds with increasing levels of abstraction that exploits information processing. From atoms to molecules to a very unique and complex molecule DNA, the increasing levels of abstraction of encoding information in reusable form and using it to define a process intent and delivering a means to execute it has resulted in a diversity of self-replicating system.

The next major evolutionary abstractions, the neuron and the neural network allowed embedded, embodied, enacted and extended cognition (known as 4E cognition) in a variety of ways [2] to model, configure, monitor and control the environment using a set of information patterns with a self-identity and optimize the execution of the processes (both of managing its own life and of managing its interactions with the environment). Cognition [3] here is the ability to process information, apply knowledge, and change the circumstance. Cognition is associated with intent and its accomplishment through various information management processes that monitor and control a system and its environment. Cognition is associated with a sense of "self" (the observer) and the systems with which it interacts (the environment or the "observed"). Cognition extensively uses time and history in executing and regulating tasks that constitute a cognitive process.

The Neocortex takes the neural network encoding of information to a higher level by creating a hierarchical temporal memory that not only composes complex information processing workflows that compose downstream specialized networks using neural processing units but also creates predictive and corrective patterns in real time. The neocortex holds the key to the learning processes which provides its cognitive data analytics and their predictive capabilities.

In this paper we examine the evolution of Genes, Neurons and the Neocortex and the evolution of information processing in Silicon based computing systems. Prof. Mark Burgin [4] from UCLA has argued that the first mathematical model employing agent technology (called an inductive Turing machine) serves as a theoretical foundation for a new model, the DIME network architecture (DNA) and allows modelling a network as an extremely large dynamic memory. The DNA agents organize connections of an information processing network in Silicon to provide cognitive computing abstractions and manage the evolution of the computing network behavior.

2. Genes

As George Dyson [5], in his book 'Darwin among the Machines,' observes "The analog of software in the living world is not a self-reproducing organism, but a self-replicating molecule of DNA. Self-replication and self-reproduction have often been confused. Biological organisms, even single-celled organisms, do not replicate themselves; they host the replication of genetic sequences that assist in reproducing an approximate likeness of them. For all but the lowest organisms, there is a lengthy, recursive sequence of nested programs to unfold. An elaborate self-extracting process restores entire directories of compressed genetic programs and reconstructs increasingly complicated levels of hardware on which the operating system runs." Life, it seems, is an executable directed acyclic graph (DAG) and a managed one at that.

According to Richard Dawkins [6], DNA molecule is really a set of plans for building a body and it indirectly supervises the manufacture of a different kind of molecule – protein. "The coded message of the DNA, written in the four-letter nucleotide alphabet, is translated in a simple mechanical way into another alphabet. This is the alphabet of amino acids which spells out protein molecules." The indirect supervision task mentioned by Dawkins is related to the function of the Gene, which is a piece of DNA material that contains all the information needed to "build" specific biological structure.

According to Damasio [7], managing and safe keeping life is the fundamental premise of biological entities. At the base of these features there is the homeostasis process. Homeostasis is the property of a system that regulates its internal environment and tends to maintain a stable, constant condition of properties like temperature or chemical parameters that are essential to its survival. System-wide homeostasis goals are accomplished through a representation of current state, desired state, a comparison process and control mechanisms. Consciousness plays an important role in all these processes; the knowledge about the internal (self) and external (environment) conditions is the key element for understanding which action (or sequence of actions) has to be performed to reach a specific goal given the current situations.

This brings to light the cellular computing model that:

1. Spells out the computational workflow components as a stable sequence of information and information processing patterns in executable form that accomplishes a specific purpose or intent,
2. Implements a parallel management workflow with another sequence of information patterns that assures the successful execution of the system's purpose (the computing network to assure biological value with management and safekeeping)
3. Uses a signaling mechanism that controls the execution of the workflow for gene expression (the regulatory, i.e. management, network) and
4. Assures real-time monitoring and control (homeostasis) to execute the four genetic transactions of replication, repair, recombination and reconfiguration.

3. Neurons

Neurons are special purpose cells created by genes that process information received through various sensor cells and control the environment using myriad motor cells. The neuron is the fundamental unit of the nervous system. The basic purpose of a neuron is to receive incoming information and, based upon that information, send a signal to other neurons, muscles, or glands. Neurons are designed to rapidly send signals across physiologically long distances. A group of neurons thus are programmed to perform specific functions (sensory and motor) that help an organism to process information, detect patterns and take action based on signals received from within or without.

The Hebbian theory of learning summarized as "cells that fire together wire together" has been the premise used to understand "neural nets" and neuronal learning. While Hebbian unit of learning is a neuron, Ray Kurtzweil on the other hand proposes an assembly (Lego-like) of neurons whose function and structure are predetermined by genes as a fundamental unit of learning executing information processing, pattern detection and communication with other units of learning. He goes on to say that the purpose of these modules is to recognize patterns, to remember them, and to predict them based on partial patterns. In essence, groups of neurons act as basic autonomic computing units with sensors and actuators processing information based on genetically programmed pattern recognition mechanisms.

4. Neocortex

The neocortex plays a crucial role in the control of higher perceptual processes, cognitive functions, and intelligent behavior. It acts as higher level information processing mechanism that uses re-composable neural subnetworks to create a hierarchical information processing system with predictive and proactive analytics. In other words, it allows us to learn, adapt and create new modes of behavior.

Ray Kurzweil [1] succinctly summarizes the nature and function of the neocortex. The basic unit of the neocortex is a module of neurons which he estimates at around a hundred. The pattern of connections and synaptic strengths within each module is relatively stable. He emphasizes that it is the connections and the synaptic strengths between the modules that represent learning. The physical connections between these modules are created to represent hierarchical information pattern processing workflows in a repeated and orderly manner. He asserts that the real strength of the neocortex is that the connections are built hierarchically, reflecting the natural hierarchical order of reality.

He goes on to say "the basic algorithm of the neocortical pattern recognition module is equivalent across the neocortex from "low-level" modules, which deal with the most basic sensory patterns, to "high level" modules, which recognize the most abstract concepts." The universal nature of this algorithm provides a stable architecture giving rise to plasticity and predictive management of information processing. The hierarchy is inherently recursive and supports redundancy. The signals go up and down the conceptual hierarchy. A signal going up means "I have detected a pattern" and a signal going down means, "I am expecting your pattern to occur" which is essentially a prediction. "Both upward and downward signals can be excitatory or inhibitory"

These insights into how the biology works are very valuable for us to design solutions that infuse cognition into silicon based information processing systems.

4. The Turing Machine

The evolution of computing seems to follow a similar path to cellular organisms in the sense that it emerged as an individual computing element (von Neumann stored program control (SPC) implementation of the Turing machine) and evolved into today's networks of managed computing elements executing complex workflows that monitor and control external environment. The basic information processing algorithm starts with the abstract Turing machine (derived from the observations of how humans compute numbers) which has three basic steps repeated till the job is complete.

Read; Compute (change state); Write

Turing machine allowed us to deterministically model any physical system, of which they are not themselves a part, to an arbitrary degree of accuracy. In order to execute the algorithm, John von Neumann's stored program implementation of the Turing machine requires a CPU and memory and the time it takes to execute the algorithm depends on the nature of the algorithm, available CPU and memory. If the CPU or memory at any time are not available, the computation fails. In order to maintain these resources, other Turing machines are used to assure that the computation is progressing and the resources are continuously available to complete the computation. Unfortunately this leads to who manages the managers conundrum and increases complexity. The problems become pronounced with increasing fluctuations in either the demand by the computation for resources or the available resource pools.

5. The DIME Computing Model

The DIME computing model recently introduced and discussed in the Turing Centenary Conference [8] addresses the management of the computation and its resources in a unified fashion. As we have seen with the biological systems, the cognitive process execution and management needs to support, use and accommodate, among others, the following:

- Self-identity
- Recursive composition scheme and scale-invariant self-management process structures
- Sequential and parallel process flows,
- Dynamic provisioning of appropriate resources to assure process flow execution,
- Multiple dynamically assignable communication channels,
- Fault tolerance to process, communications or infrastructure failure,
- Process and infrastructure level security, and
- Run-time monitoring and control of resources to meet changing process priorities, latency constraints and workload-fluctuations caused by interactions with the environment.

In short, the cognitive process requires active management of dynamic coupling between various elements of the system, where each change in one element influences some other element's direction of change while its computation is still in progress and must be accounted for in any computational model. In addition, the resulting changes in computational resource requirements caused by the non-deterministic influences of various changes in the overall system have also to be dynamically managed.

The cognitive computing model construct derived from the Turing Oracle design discussed in his thesis, is the DIME basic process, which performs the {read → compute → write} instruction cycle executing an algorithm or its modified version the {interact with external agent → read → compute → interact with external agent → write} instruction cycle. This allows the external agent to influence the further evolution of computation while the computation is still in progress.

6. The DIME Network Architecture

The DIME network architecture accomplishes the management of process evolution by introducing three key functional constructs to enable process design, execution and management and improves both the resiliency and efficiency of distributed computing elements.

Machines with an Oracle: Executing an algorithm, the DIME basic processor performs the {read -> compute -> write} instruction cycle or its modified version the {interact with a network agent -> read -> compute -> interact with a network agent -> write} instruction cycle. This allows the different network agents to influence the further evolution of computation, while the computation is still in progress. It is assumed that a DIME agent knows the goal and intent of the algorithm (along with the context, constraints, communications and control of the algorithm) the DIME basic processor is executing and has the visibility of available resources and the needs of the basic processor as it executes its tasks. In addition, the DIME agent also has the knowledge about alternate courses of action available to facilitate the evolution of the computation to achieve its goal and realize its intent. Thus, every algorithm is associated with a blueprint (analogous to a genetic specification in biology), which provides the knowledge required by the DIME agent to manage the process evolution.

Blue-print or policy managed fault, configuration, accounting, performance and security monitoring and control: The DIME agent, which uses the blueprint to configure, instantiate, and manage the DIME basic processor executing the algorithm uses concurrent DIME basic processors with their own blueprints specifying their evolution to monitor the vital signs of the DIME basic processor and implements various policies to assure non-functional requirements such as availability, performance, security and cost management while the managed DIME basic processor is executing its intent.

DIME network management control overlay over the managed Turing oracle machines: In addition to read/write communication of the DIME basic processor (the data channel), other DIME basic processors communicate with each other using a parallel signaling channel. This allows the external DIME agents to influence the computation of any managed DIME basic processor in progress based on the context and constraints. The external DIME agents are DIMEs themselves. As a result, changes in one computing element could influence the evolution of another computing element at run time. The signaling channel and the network of DIME agents can be programmed to execute a process, the intent of which can be specified in a blueprint. Each DIME basic processor can have its own oracle managing its intent, and groups of managed DIME basic processors can have their own domain managers implementing the domain's intent to execute a process. The management DIME agents specify, configure, and manage the sub-network of DIME units by monitoring and executing policies to optimize the resources while delivering the intent.

7. Conclusion

In this paper, we investigated the parallels between the evolution of genes, neurons and the neocortex and argue that the new DIME network architecture provides the framework for implementing the specification and execution of process evolution based on pre-defined policies and best practice specifications with self-configuring, self-monitoring and self-correcting behavior. An agent architecture that has the global knowledge of the process and its evolution in the form of a blueprint is executed with self-managing properties. The computational process evolution knowledge that incorporates the intent, the environment, available resource pools, and best practices to monitor and cope with deviations when the vital resources fall short are encoded in the blueprint. The blueprint is interpreted to create a network of resources that are used to configure, monitor and control the evolution of computational process. A real-time communication between the agents that monitor and control the computing and the agents that monitor and control the resources. Together the agent network provides the cognitive ability to self-manage the computational process. Notice that the agents only know the resource management requirements and do not know the content of the computations.

We believe that the DNA framework provides a means to infuse cognition into computing and create the beginnings of a self-managing system. It is possible to extend the agent architecture to create a hierarchy of temporal knowledge that can not only manage from pre-defined blueprints but also modify the blueprint with learnings from the history of its evolution [9].

An implementation of DNA evolving IT process is shown in a video https://youtu.be/tu7EpD_bbyk The video demonstrates the implementation where an application availability and performance are managed using the agents discussed in this paper. The ability to self-provision, auto-scale, auto-failover and live migrate computing processes without disturbing the user experience extends the current IT to cognitive IT.

References

1. Ray Kurzweil, How to Create a Mind: The Secret of Human Thought Revealed, Penguin Books, New York, NY, 2013
2. L. Barrett, "Beyond the Brain," Princeton University Press, Princeton, 2011.
3. R. Mikkilineni, "Going beyond Computation and Its Limits: Injecting Cognition into Computing." Applied Mathematics 3 (2012): 1826.

4. Burgin, M., Mikkilineni, R., and Morana, G. (2016) Intelligent Organization of Semantic Networks, DIME Network Architecture and Grid Automata, International Journal of Embedded Systems (IJES), Vol. 8, No. 4.
5. George B. Dyson, Darwin among the Machines: The Evolution of Global Intelligence, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1997
6. Dawkins, Richard. *The Selfish Gene*. New York : Oxford University Press, 1978,
7. Damasio, A. Self Comes to Mind: Constructing the Conscious Brain. New York: Pantheon Books, 2010.
8. Mikkilineni, R.; Comparini, A.; Morana, G. The Turing O-Machine and the DIME Network Architecture: Injecting the Architecture Resiliency into Distributed Computing, Proc. The Turing Centenary Conference, Turing-100, Alan Turing Centenary, EasyChair Proc. in Computing, EPiC vol. 10 (ed. A. Voronkov), Manchester, UK, June 2012, 239-251.
9. Burgin, M. Super-recursive Algorithms, New York: Springer, 2005