Article

Not peer-reviewed version

A New Class of Intelligent Machines with Self-Regulating, Event-Driven Process Flows for Designing, Deploying, and Managing Distributed Software Applications

Rao Mikkilineni * and W. Patrick Kelly

Posted Date: 16 November 2023

doi: 10.20944/preprints202311.1104.v1

Keywords: Self-Regulation; Event-Driven Architecture; Digital Genome; Autopoiesis; Cognition.

Article

A new Class of Intelligent Machines with Self-Regulating, Event-Driven Process Flows for Designing, Deploying, and Managing Distributed Software Applications

Rao Mikkilineni [1,*] and W. Patrick Kelly [2]

[1] Golden Gate University, Edward S. Ageno School of Business, San Francisco, California 94105, USA; rmikkilineni@ggu.edu
[2] Opos.ai, San Mateo, CA 94401, USA; pkelly@opos.ai
* Correspondence: rmikkilineni@ggu.edu

Abstract: The benefits of event-driven architecture (EDA) derive from how systems and components are loosely coupled, which can facilitate independent development and deployment of systems, improved scaling and fault tolerance, and integration with external systems, especially in comparison to monolithic architectures. With the advent of new technologies such as containers, and microservices, a new generation of distributed event streaming platforms are commonly used in event-driven architecture for efficient event-driven communication. However, the asynchronous and distributed nature of EDA poses several problems that include handling failures, the dependence of an end-to-end transaction on individual component stability, etc. In this paper, we describe a new approach to designing self-regulating distributed applications with autopoietic and cognitive workflow management. This approach is based on the new science of information processing structures derived from the General Theory of Information. Just as a genome enables self-organizing and self-regulating biological structures, a digital genome enables a specific software application with several components, the ability to use distributed resources and self-regulate the evolution of the system based on functional and non-functional requirements, and best-practice policies that maintain the stability, safety, and survival under non-deterministic fluctuations in the demand for resources. In addition, cognitive workflow management assures end-to-end transaction delivery.

Keywords: Self-Regulation; Event-Driven Architecture; Digital Genome; Autopoiesis; Cognition

1. Introduction

As Cameron Hunt [1] points out 'Event-driven architecture (EDA) is a software architecture paradigm promoting the production, detection, consumption of, and reaction to events. It is used in distributed applications where various software components composing a 'Distributed Application' communicate with each other asynchronously. In EDA, communication between the various software components composing a 'Distributed Application' is always Asynchronous: "If there is a response … it's indirect"'

The benefits of event-driven architecture (EDA) derive from how systems and components are loosely coupled, which can facilitate independent development and deployment of systems, improved scaling and fault tolerance, and integration with external systems, especially in comparison to monolithic architectures [2–5]. With the advent of new technologies such as containers, and microservices, a new generation of distributed event streaming platforms are commonly used in event-driven architecture for efficient event-driven communication. An integration of event-driven architecture and service-oriented architecture is discussed by Papazoglou and Van Den Heuvel in [6]. However, the asynchronous and distributed nature of EDA poses several problems [7–10] that include handling failures, the dependence of an end-to-end transaction on individual component stability, etc. In this paper, we describe a new approach to designing self-regulating distributed

 doi:10.20944/preprints202311.1104.v1

applications with autopoietic and cognitive workflow management using the tools derived from the General Theory of Information, GTI [11–14].

Generally speaking, a business process is a set of tasks, undertaken to create output utilizing various inputs. The tasks are designed by people with a purpose to achieve a goal. The tasks are performed by various individuals sharing the knowledge of operations and exchanging information or goods in the form of inputs and outputs. Operations are performed by individuals on the inputs to produce the outputs in the form of goods, information, or services which are specified as tasks in another process. Information and knowledge play a key role in executing various processes. Knowledge is derived from information [11]. Descriptive knowledge is an awareness of facts. Operational knowledge, on the other hand, refers to the capability of a person, tool, or machine (analog or digital) to perform a set of tasks that are specified. While descriptive knowledge can be stored and used in books, mental models, and computers, operational knowledge requires an agent to execute it such as a person, a tool, or a machine. Operational knowledge is associated with processes involving various entities, their interactions, operation methods to achieve specific goals, and best practices for their effective use.

Human intelligence always strives to manage process efficiency and resiliency by monitoring and optimizing the processes using knowledge acquired through experience. With the advent of general-purpose computers, software programs were used to perform tasks that can be easily described by a list of formal, mathematical rules or a sequence of event-driven actions such as modeling, simulation, business workflows, interaction with devices, etc. In essence, process knowledge describes the state of the structure of a system and various operations that describe the dynamics or the evolution of the system caused by events that change the structure. In computers, the state of the system is described by the data structures represented as sequences of symbols. The operational knowledge is contained in a program also in the form of a sequence of symbols. The programs are executed in hardware (consisting of a central processor unit and memory at a minimum using power to execute binary arithmetic) and transform the input state into an output state using the tasks specified. Algorithms specifying the tasks are used by the digital machines to automate processes to improve their efficiency and resiliency. In addition, a special kind of algorithm that mimics the neuron and the neural networks in the human brain is used to process information and perform "tasks that are easy for people to perform, but hard for people to describe formally – problems that we solve intuitively, that feel automatic, like recognizing spoken words or faces in images" [[15], p. 1]. The neural networks have proven to be very useful in processing data in the form of text, voice, images, and videos and provide a knowledge representation derived from the information provided by the data. The large language models (LLMs) are notable for their ability to achieve general-purpose language understanding and generation using massive amounts of data to learn through billions of parameters during training and consuming large computational resources during their training and operation [16]. While the LLMs are thought to acquire embodied knowledge about syntax, semantics, and "ontology" inherent in human language corpora, they also acquire inaccuracies and biases present in the corpora. However, the LLMs have proven to be useful both in implementing process automation and intelligent decision-making using the insights gained from training these models. Figure 1 shows the relationship between people processes and technology used to automate, monitor, and manage to optimize processes.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 November 2023 doi:10.20944/preprints202311.1104.v1

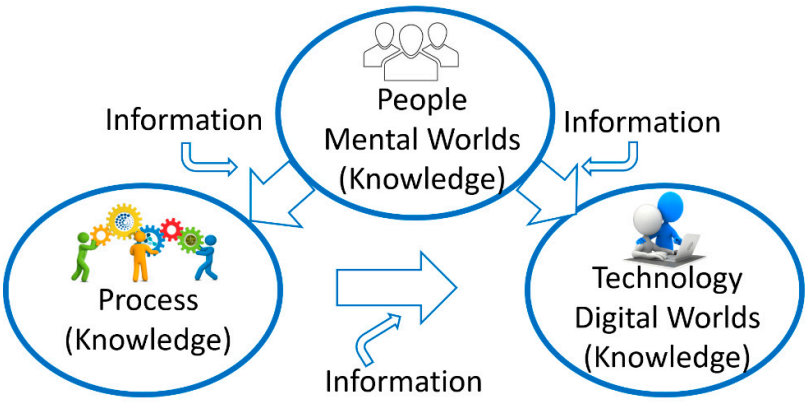

Figure 1. Role of people, processes, and technology in business management.

The genome is a prime example of how knowledge, transmitted by the survivor to the successor through natural selection, is used to build, monitor, and manage structures that not only maintain their stability, safety, and survival but also develop processes to interact with external structures that exist in the material world [17–20]. Intelligence is derived from descriptive and operational knowledge that is both inherited and learned through cognitive processes also enabled by the genome. The cognitive processes are used by all biological systems to receive information from the material world and convert it into knowledge which is used to model and interact with the material world. Human intelligence using knowledge not only enables the processes and their use to improve the efficiency and resiliency of various tasks involved in a business operation using various tools but also enables the use of general-purpose computers to replicate knowledge in digital machines and execute processes to augment human intelligence.

Knowledge acquired through the genome enables both autopoietic and cognitive behaviors, which results in its use to manage the system with self-regulation and interact with the external world. Autopoiesis [21,22] refers to the behavior of a system that replicates itself and maintains identity and stability while its components face fluctuations caused by external influences. As Shettleworth points out time and memory are fundamental to our cognitive processes and our ability to learn, adapt, and make intelligent decisions [23]. Cognitive processes use them to process information into knowledge and use it to manage its interactions between various constituent parts within the system and its interaction with the environment. Cognition uses various mechanisms to gather information from various sources, convert it into knowledge, develop a history through memorizing the transactions, and identify new associations through their analysis. In this paper, we argue that time and memory are also necessary in digital systems to make intelligent decisions.

The General Theory of Information describes the relationship between information and knowledge used by biological systems through their mental structures and provides a bridge between the material structures created by matter and energy transformations in the physical world [9]. The Burgin Mikkilineni thesis discusses the relationship between knowledge representations and the execution of operational knowledge and autopoietic and cognitive behavior of biological and artificial systems [9,14,24]. The thesis states that these behaviors must function on three levels of information processing systems and be based on triadic automata. The thesis aims to guide the infusion of autopoietic and cognitive behaviors into digital machines. It shows how to use the structural machine to build a cognitive reasoning system that integrates knowledge from various digital symbolic and sub-symbolic computations. This approach is analogous to how the neocortex repurposed the reptilian brain, paving the path for digital machines to mimic living organisms using an integrated knowledge representation from different sources. In this paper, we demonstrate the use of GTI and the tools derived from it to design, build, operate, and manage a new class of intelligent machines with self-regulating, event-driven process flows using distributed software components and IaaS (infrastructure as a service) and PaaS (Platform as a Service) infrastructures.

 doi:10.20944/preprints202311.1104.v1

In section 2, we briefly describe GTI and its relevance to designing the new class of intelligent machines used for process automation and intelligent decision-making. In section 3, we describe an application that is used in an early diagnostic process where a healthcare provider interacts with a patient using medical knowledge from a variety of resources including the large language models (LLMs) and early diagnosis medical practices. The purpose is to demonstrate how an event-driven workflow can be deployed with autopoietic and cognitive behaviors to implement a medical-knowledge-based digital assistant used to reduce the knowledge gap between a patient and healthcare providers. In section 4, we discuss the advantages of the new approach from the current state-of-the-art. In section 5, we present some conclusions from our experience and suggest some future applications of this novel approach of using GTI to build a new class of digital automata that break the boundaries of the Church-Turing thesis and provide a path to deploy self-regulating and cognitive distributed applications with domain-specific knowledge with an identity, and a cognitive memory that represents the state evolution and history of the system.

2. The General Theory of Information, Information, Knowledge, and Event-Driven Cognitive Workflows with the Memory of the Past

The General Theory of Information articulated by Mark Burgin, is a comprehensive framework that provides a unified context for existing directions in information studies. GTI has significant implications for computer science. It allows for a comprehensive definition of information, explains the relations between information, data, and knowledge, and demonstrates how different mathematical models of information and information processing structures are related. The GTI is built on an axiomatic base as a system of two classes of principles and their consequences. The first class consists of the ontological principles, which reveal general properties and regularities of information and its functioning. Principles from the second class explain how to measure information [25].

The parametric definition of information utilizes a system parameter called an infological system that plays the role of a parameter that discerns different kinds of information, e.g., social, personal, chemical, biological, genetic, or cognitive, and combines all existing kinds and types of information in one general concept of “information.” An “infological” system is a concept that serves as a system parameter that discerns different kinds of information, such as social, personal, chemical, biological, genetic, or cognitive. This concept allows for the combination of all existing kinds and types of information into one general concept of "information". The concept of an infological system shows that not only do living beings receive and process information, but also it is natural to treat the memory of a computer as an infological system. Then what changes this memory is information for the computer.

In essence, GTI relates the material world, the mental world, and the digital world using information as the bridge [9]. The material structures formed through transformations of matter and energy contain information about their structures (state) and their transformations or dynamics through their interactions. This information (ontological information) is carried by material structures in the form of their state and dynamics. This information (the observed) is perceived by the biological systems (observers) and is dependent on the nature of the cognitive apparatuses and their capabilities. Therefore, the perceived information (epistemic information) and the derived knowledge are different for different observers.

GTI tells us that epistemic information is converted into knowledge as mental structures by the cognitive apparatuses of the observer. GTI provides a common representation for material structures in the physical world and the mental structures in the mental world using the mathematics of ideal structures called named sets or fundamental triads. These fundamental triads consist of various entities, their relationships, and their behaviors as they change their state as they interact with each other. For example, what we observe physically or conceive mentally using mental structures, we give labels and represent them as entities. When we see a dog, we associate a label with the image. This becomes an entity in the mental structure. Entities interact with each other and groups are

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 November 2023 doi:10.20944/preprints202311.1104.v1

composed of entities with relationships. When any entity changes its state, it causes various behaviors that are determined by the nature of the relationship.

This representation of knowledge allows us to model how various autopoietic and cognitive processes are executed as life processes in biological systems. The same model can be used to represent knowledge in digital computers

Figure 2 shows the physical, mental, and digital worlds where ontological information is carried in the material structures, in the form of their states and their evolution as they interact with each other following the laws of transformation of energy and matter, is processed and used. The epistemic information is received by the mental world of biological systems through their cognitive apparatuses and is converted into knowledge which updates the mental structures that execute the life process knowledge inherited through the genome. Just as energy has the potential to create or change material structures interacting with each other through the natural laws of transformation of energy and matter, Information has the potential to create or update knowledge represented in the mental and physical structures of biological systems. GTI asserts that knowledge can be represented using the ideal structures in the form of entities, relationships, and behaviors when events or interactions change their state. These ideal structures are represented by named sets or fundamental triads shown also in Figure 2. The knowledge that is inherited and learned from the genome provides mechanisms to execute life processes that include autopoietic and cognitive behaviors in biological systems.

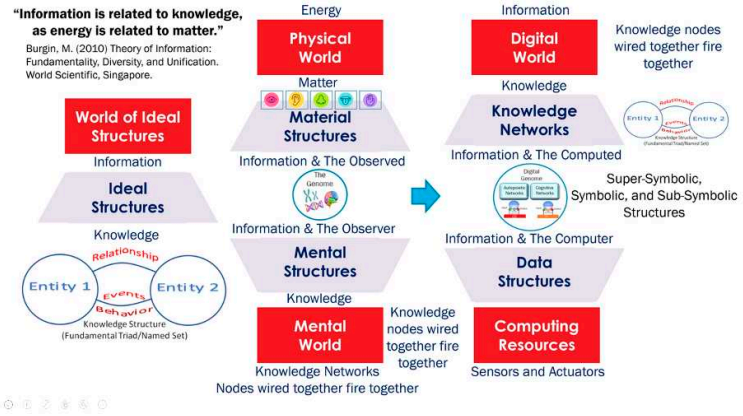

Figure 2. The role of information and knowledge in the material, mental, and digital worlds [9].

In addition, the Burgin-Mikkilineni (BM) thesis [9,14] relates how digital structures can be used to represent knowledge using entities, relationships, and behaviors (named sets or fundamental triads) using information as input. The ontological BM thesis states that the autopoietic and cognitive behavior of artificial systems must function on three levels of information processing systems and be based on triadic automata. The axiological BM thesis states that efficient autopoietic and cognitive behavior has to employ structural machines. This thesis provides a model not only to describe the information processing structures in living organisms but also to model and infuse autopoietic and cognitive behaviors into digital automata. In essence, the BM thesis extends the principles of the GTI to the realm of artificial systems, providing a theoretical foundation for understanding and designing complex, intelligent systems that are autopoietic and cognitive.

Using structural machines, cognizing oracles, and knowledge structures to build autopoietic and cognitive digital automata, [26,27] we have previously implemented a design of the medical-knowledge-based digital assistant [28]. In this paper, we will extend the use of knowledge structures to design and implement an event-driven workflow management that provides a long-term associative memory representing the history of events and their impact on the dynamics of the state of the system. Associative memory allows individuals to make and then recall connections and inferences of various entities, their relationships, and associated behaviors caused by various events. It is used in multilevel memory systems, in which a small fast memory such as a cache may hold

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 November 2023 doi:10.20944/preprints202311.1104.v1

copies of some blocks of a larger memory for rapid access. It is designed to quickly find matching knowledge (entities, relationships, and event-driven behaviors). In humans, this relates to visual and verbal information, such as remembering how two words are related (e.g., man – woman), or seeing an object and its alternate name (e.g., a guitar). When the mind focuses on an individual, a quick memory of events and relationships associated with that individual is recalled to provide the associated knowledge of the experiences. A similar associative model that represents the knowledge (entities, relationships, and their event-driven behaviors) is implemented using the digital genome of the medical knowledge-based digital assistant.

In the next section, we describe our implementation of the associative memory model of the medical knowledge-based digital assistant.

3. The Digital Genome and the Event-Driven Workflow Design, Deployment, Operation, and Management

Data refers to observed or mentally conceived entities with a label and a value. For example, "a horse is a label that refers to an image of a horse." Information is a collection of data that has been processed to identify the relationships between these entities and convert them into data structures. Conventional computers collect data, process it into information by identifying patterns and relationships, and then using algorithms and models to convert this information into knowledge, which can be used to make predictions, decisions, and take actions. Both data structures and programs are represented as sequences of symbols. Intelligence is the ability to take information, convert it into knowledge, and use it to accomplish a purpose or a goal. The degree of intelligence depends on the ability to operate with knowledge. This is true both for natural and artificial systems. At the same time, intelligent systems work not directly with knowledge but employ knowledge representations of different types, because knowledge per se is an abstract structure. One of the most powerful and flexible forms of knowledge representations is a schema. The Turing machine provides a schema on how to represent knowledge in the form of programs (algorithms) and data structures. General Theory of Information (GTI) provides a new schema processing in the form of structural machines, cognizing oracles, and knowledge structures. According to GTI, information is converted into knowledge and represented in the form of named sets or Fundamental Triads that describe not only the entities, and their relationships but also their behaviors when events alter any of the attributes of these entities, the state change fluctuation in their structures modify their interactions. Burgin and Mikkilineni [[27] p. 12] "discuss how the use of new theory about knowledge structures, triadic automata, and structural machines allow us to design a new class of distributed information processing structures that use infware containing hierarchical intelligence to model, manage, and monitor distributed information processing hardware and software components as an active graph representing a network of networked processes. The infware, i.e., the processed information in the form of data structures, uses the operations on the schema representing the hardware, software, and their evolutionary behaviors as dynamic knowledge structures." We refer the reader to this paper for the theory [27] and we focus in this paper on an application of this schema-based approach to designing an autopoietic and cognitive system designed to represent medical knowledge about an early diagnosis process.

3.1. Designing the Digital Software Genome

As Kelly et al., [28] describe the digital software genome "specifies the knowledge to execute various tasks that implement functional requirements, non-functional requirements, and the best practices to assure that the process objectives are achieved. Functional requirements deal with domain-specific process execution, components of knowledge acquisition, model creation, the evolution of knowledge structures, analysis of events and actions, etc. Nonfunctional requirements deal with deploying, operating, and managing the compute resources for the various application components, where to set up IaaS (infrastructure as a service that provides required hardware, operating system), and PaaS (platform as a service that provisions containers, databases, middleware, etc.), deploy auto-failover, auto-scaling, and live-migration policies. Best practices deal with decision-

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 November 2023 doi:10.20944/preprints202311.1104.v1

making by looking at the evolution of the system predicting risk and suggesting remedies for both maintaining functional and non-functional requirement fulfillment."

In the case of the medical-knowledge-based digital assistant, the knowledge is acquired from various sources:

- ChatGPT developed by OpenAI, and various other large language models acquire medical knowledge from a diverse range of sources. Deep learning algorithms are trained on a mixture of licensed data, data created by human trainers, and publicly available data. These sources vary from a wide array of medical textbooks, reputable health websites, and other educational material to human input from experts who provide a wealth of medical knowledge.
- Process knowledge of early diagnosis that involves initial diagnosis assessment using patient details, history, risk factors, etc.,
- Medical knowledge about various symptoms, diseases, the impact of risk factors, etc.,
- Medical knowledge about diseases and their relationships to various specialized disciplines treated by various specialists,
- The knowledge of the patient's history of various events, interactions, and their impact on the diagnosis, and
- Input from various tests, professional evaluations, etc.

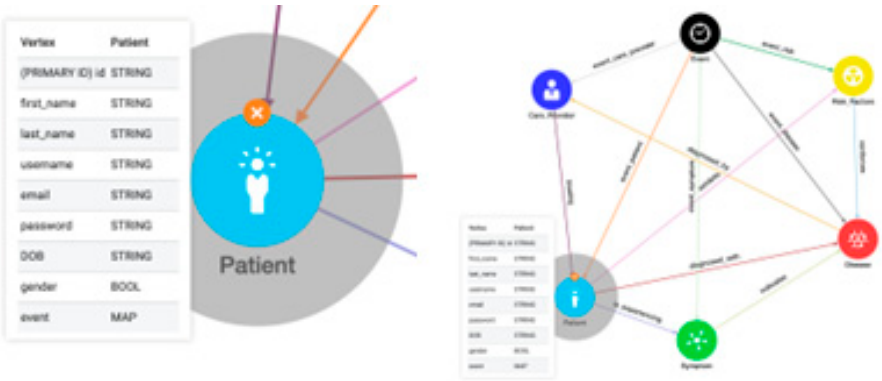

Figure 1. The Digital Genome Schema including entities, relationships, and their event-driven behaviors.

The knowledge provides the input to create a schema as a software digital genome's functional requirements which define various entities, their relationships, and behaviors when events change the state of the system. Figure 1 shows an example of a schema used in the design of the digital assistant genome. The purpose of the genome is to define the life processes of the system that acquires data from various sources and uses medical knowledge to provide model-based transparent recommendations with event-based history visualization as we discuss in this paper.

The picture shows the entities (the patient, healthcare provider, event symptoms, diseases, and risk factors. Each entity is represented as a vertex with process knowledge to execute various functions based on the input received. The output is shared with other entities whose functions are impacted by the information received using shared knowledge defined in the functional requirements of the digital assistant genome. The vertices and the edges that connect various entities that have shared knowledge constitute a knowledge network where nodes that are wired together fire together to execute a transaction spanning the knowledge network with well-defined pre- and post-conditions. The event history provides a long-term associative memory structure which is used to reason about the system behavior using various events and the system's state evolution.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 November 2023 doi:10.20944/preprints202311.1104.v1

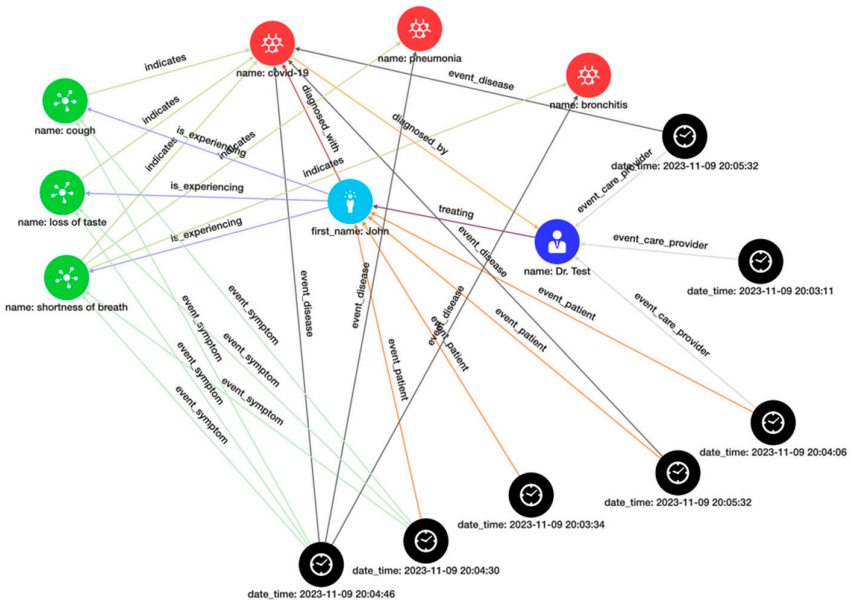

Figure 2. Associative Memory Graph of patient 1 with event history.

The events define an action that changes the state of the system. When a patient or a provider is registered in the system, those instances are created based on the functionality of the vertices designed in the digital software genome. The vertices are defined by their inputs, functions, and outputs connected to other entities through shared knowledge. The symptoms communicated by the patient are given to the symptom vertex which creates these instances and communicates with the knowledge acquisition function to identify various potential diseases, and the associated risk factors, and creates a diagnosis hypothesis by creating various disease instances, risk factors, and weights of the symptom disease edge connections. For example, symptoms such as cough, loss of taste, and shortness of breath produce a hypothesis of various possible diseases such as influenza, COVID-19, common cold, etc., with various weights for the symptom and disease instance connections along with risk factors. The healthcare provider gets the model-based information to consider in taking the next steps. Figure 2 shows the event flow of a simple diagnosis process where ChatGPT is used to get medical knowledge. This knowledge acquisition can be enhanced with information from multiple LLM or other sources for verification and validation. This reduces the self-referential circularity of pieces of knowledge by validating them with multiple sources connected to reality. The validated knowledge then is used in making the final decision about the next steps.

The system state containing the entities, their relationships, and their behaviors is represented as a knowledge network where each node is a knowledge structure that is executed by a structural machine implemented as a Python program.

3.2. Autopoietic and Cognitive Network Managers

To represent entities and their process knowledge, each entity is developed as an individual Flask application where each process can be specified within the code. This application is then containerized with docker and deployed on a cloud environment (to meet the non-functional requirements using autopoiesis). Each containerized entity can communicate information to another entity through an API connection that is established through a URL specified by a cognizing oracle called the Cognitive Network Manager (CNM). This CNM is also a containerized Flask application with the specific task of maintaining network interaction between containerized entities. Each of these containerized entities as well as the CNM are maintained by the Autopoietic Process Manager

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 November 2023 doi:10.20944/preprints202311.1104.v1

(using best practices and policies). The Autopoietic Process Manager is a Kubernetes layer that specifies and manages the container environments.

Kubernetes allows our system to exhibit autopoiesis through a deployment manifest that defines the static components for the non-functional requirements for our container orchestration. The resources and request limits for each container are specified within policies that guide the auto-scaling decisions through Horizontal Pod Auto-scalers to maintain the appropriate metrics and thresholds [29]. This allows Kubernetes to dynamically adjust and automatically recover from failing instances and enhances the failover mechanism. The implementation of a liveness probe detects and automatically recovers failing container instances, while a readiness probe is used to ensure new container instances are only added to the load balancer when they are ready to handle traffic. With the cooperation of the CNM, there is no need for data sharing as each network connection is established through the CNM as a cognizing oracle. Through the utilization of Kubernetes Services to expose the application internally and externally efficient load balancing and traffic distribution are maintained throughout the system. By combining these components Kubernetes can be effectively leveraged to manage and maintain static architecture while dynamically scaling and handling failovers, ensuring a resilient and responsive deployment capable of self-preservation and replication.

This autopoietic process is maintained through a policy that determines the environment, cloud provider, cloud resources in the form of IaaS and containerized PaaS components such as middleware and databases), and replication of each containerized entity process. The autopoietic process allows for a sense of self throughout the network as resources are managed and distributed to maintain an optimal state of operation. New resources can be dynamically added to the system depending on load by automatic deployment of new container instances. This process can split the load between two container instances while maintaining seamless network connectivity for the end user. In essence, the autopoietic manager knows how to acquire the resources required for an end-to-end service deployment, operation, and management. It is responsible for creating new nodes ready for execution with distributed containerized application components and associated data. On the other hand, the cognitive network manager has the knowledge of the application workflow and which nodes are connected to which nodes with shared knowledge and maintains the workflow stability, safety, and survival with policy-based connection management. In other words, the autopoietic network manager provisions and readies the nodes and the cognitive network manager establishes and manages the connection between the nodes that exchange information. That is how the wired nodes fire together to execute a collective process with information exchanged between them.

3.3. Event-Driven Associative Longtt-Term Memory

Each node is a process executed in a Python Program with various instances and their evolution is maintained. A long-term memory representation is stored as a network of networks in a graph database. Figure 3 shows a graphical representation of the knowledge network with a local view related to patient 1 and the associated event history. The long-term memory contains all entity instances, their relationship, and event-based behavior history. Graphical queries allow us to traverse history to discover various insights based on their relationships and behaviors.

With the integration of an event-driven architecture, we can take these real-time interactions and execute a process of retrospective queries to focus on the granular actions and events that evoke a series of reactions. As machine learning processes are implemented each decision layer can be traced based on the weighted edges and decision value pre-scribed by the network algorithm enabling the production of models and simulations. Events occur sequentially through time and act as a log of the changing state throughout the system. The event entity itself is a process that is containerized and integrated into the cloud environment, the same as any other entity. An event consists of three main components, sensors, actuators, and various interacting entities affected by the input information. Sensors cause input to various entities. Actuators are activated by information output from the entities. As entities perform tasks that interact with other entities changing the long-term associative memory of our system, an event when it occurs, is time-stamped and recorded. To log this event, we

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 November 2023 doi:10.20944/preprints202311.1104.v1

need to identify the acting entities. The entity that contained the event trigger acts as the sensor of our event, and the reacting entity acts as the actuator. This information is held within the event entities to facilitate data queries of event trees.

The relationship between entities and events through time can be viewed in two ways. The first view is an 'associated memory view' where there is only a single instance of an entity and its myriad relationships with other entities formed by various events are depicted. Within the associative memory, each entity is a single instance (patient 1 in the figure), and its relationships to all other connected entities with the events that caused state change. This forms a network where every relationship between any entity is logged as an event and related between actors without duplicates or copies.

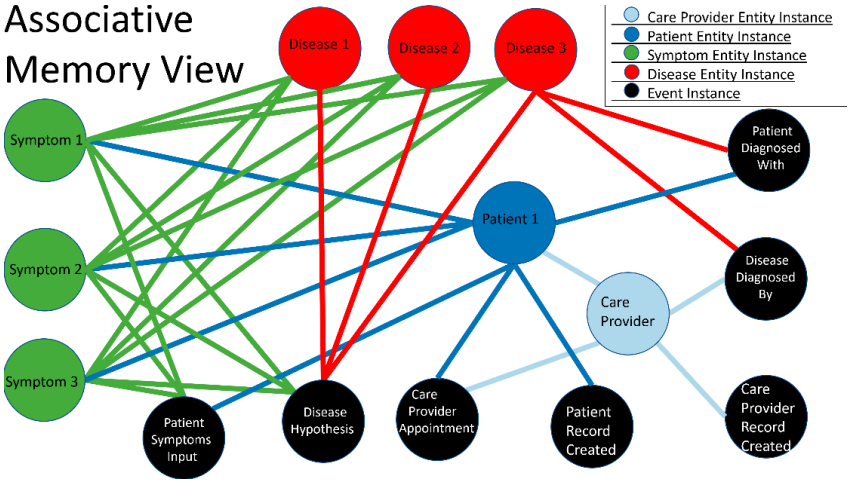

Figure 3. Associative memory view of patient 1 with event history.

In contrast, the second view, the 'event history view' may have multiple instances of a single entity to show the progression of the relationships of our system through time. As behaviors are expressed through the system each event catalogs the sensor, as the entity that instigated the action, and the actuator as the receiving entity to perform a process. The progression of information through event-driven architecture can be explained as a parent-child relationship through successive generations. As the information contained within a parent entity changes a new child instance of the entity is created and related to its parent. This relationship allows for data traversal through time as every state of an entity is logged as an individual instance. As a process changes the state of our associative memory each entity that was affected gets logged into a vertical column along with every other entity that was affected by that event. Edges are created between the parent and child vertices to represent the progression of self through the timeline. Every interaction between event actors is given an edge to represent the data that was changed or resulted in the creation of a new entity instance.

The example shown in Figure 4 has event 0 and event 1 as the instantiation of the first two entities as a care provider and a patient. Event 2 shows the interaction between the patient and the provider during an appointment. In this appointment, the patient expressed that they are experiencing some symptoms. This change in the patient state along with the change or creation of symptom entities and relationships to the patient are logged in event 3. These symptoms are then related to the possible disease hypotheses prediction on event 4. Events 5 and 6 show two successive events showing the new relationship that is formed between the patient and the disease and the relationship between the care provider and the disease upon an official diagnosis. Event 7 shows the care provider's appointment with a new patient as a continuation of the process. As the event history view is populated complex networks are formed between interacting entities. Through the queries on the events, rebuilding the state of data becomes possible. These events can then be traversed through time to examine the emergent behaviors of interacting entities allowing for the granular exploration

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 16 November 2023 doi:10.20944/preprints202311.1104.v1

of the reasoning behind machine learning processes and the changing states that cause variation in predictive analytics.

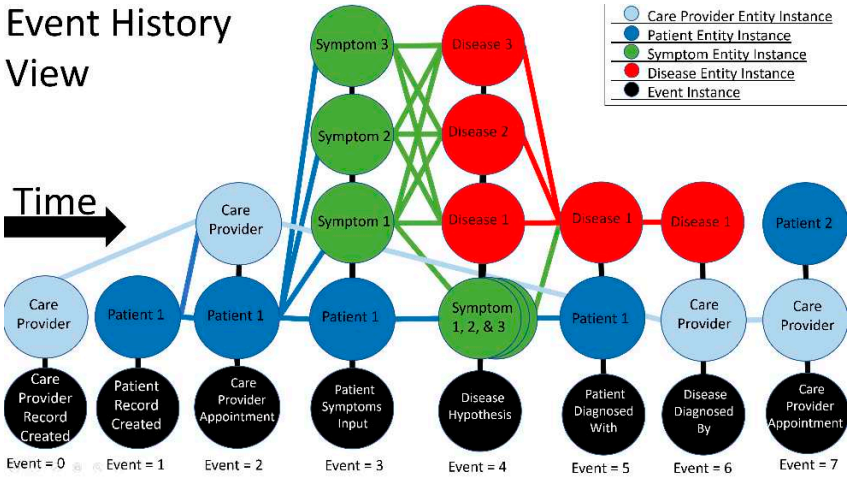

Figure 4. Time flow of events and their impact on the system state and evolution represented as a network with nodes and connections.

3.4. Model-Based Reasoning

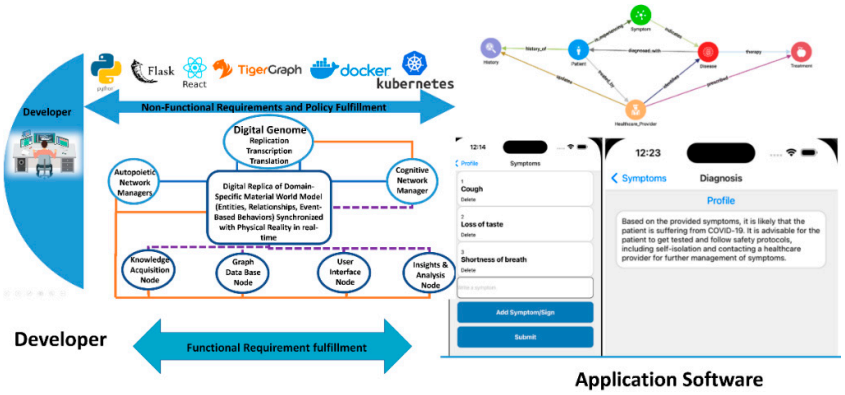

Figure 5. Process architecture of the medical knowledge-based digital assistant.

The approach we have presented here, based on GTI, allows us to represent the system knowledge as a knowledge network schema with operations that are executed using a Structural Machine. The knowledge network consists of nodes wired together with shared knowledge to execute specific behaviors. Each node is a process that has the knowledge to receive information, execute a process, and share outputs with other nodes using shared knowledge designed in the domain-specific digital software genome. There are two types of knowledge structures. The first type is a process that executes specific functions in the node based on the input and produces output shared with other nodes. The second type is cognizing oracles with the knowledge of a sub-network of nodes (deployment, connection management, operation, and non-functional requirement fulfillment). The Autopoietic and cognitive network managers are examples of the second type. Figure 4 shows various modules used in the medical knowledge-based digital assistant.

Various modules depict the implementation of the knowledge network. The long-term associative memory is contained in the knowledge representation stored in the Python Processes and the Graph Database. Various insight analysis nodes use the associative memory to create a local cache and use various processes to verify, validate, and use the knowledge to gain insights through queries, simulations, and other modeling approaches.

 doi:10.20944/preprints202311.1104.v1

We believe that this approach comes as close to human reasoning behavior as we can by integrating current algorithmic and deep learning neural network computing structures with a common transparent domain-specific model-based reasoning.

4. Discussion

In this paper, we have demonstrated the use of GTI to build a new class of digital automata that are infused with both autopoietic and cognitive processes that biological systems use to build themselves and execute life processes that are inherited from their genomes using replication and metabolism. The digital software genome specifies the life processes of a software system. It leverages structural machines and knowledge structures to model the domain knowledge (the medical knowledge required for early diagnosis processes in our implementation). The concept of a digital software genome enables efficient process execution by discovering and utilizing computing resources in the environment. It assembles hardware and cognitive apparatuses, contributing to the development of self-regulating domain-specific software. Recent advances in LLMs have given us a way to tap the wide knowledge pool of medical knowledge and use it to bridge the knowledge gap between various entities participating in the early diagnosis process. For example, a primary care physician, various specialists who treat particular types of diseases, the patient, the nurse who assists the patient, the insurance providers, various laboratories who conduct specialized tests, regulators who oversee the medical practice governance, etc., all bring different levels of domain knowledge that contributes to the overall early diagnosis process. What we have shown here is that we can capture various interactions between various entities as events, and provide a schema to create a knowledge structure and capture the behaviors as interactions change the state of the system. The evolution of the system from an event to event provides the associative memory of the system including all instances participating in the workflow.

5. Conclusions

It is important to emphasize that the general theory of information allows us to extend current computing models based on the stored program computing implementation of the Turing Machine with a super-symbolic computing model [30]. The Church-Turing thesis is concerned with the nature of computable functions. It states that a function on the natural numbers can be calculated by an effective method if and only if it is computable by a Turing machine. This thesis has near-universal acceptance, although it cannot be formally proven, as the concept of effective calculability is only informally defined.

On the other hand, the Burgin Mikkilineni thesis is a more recent proposition that focuses on the autopoietic and cognitive behavior of artificial systems. The ontological BM thesis states that the autopoietic and cognitive behavior of artificial systems must function on three levels of information processing systems and be based on triadic automata. The axiological BM thesis states that efficient autopoietic and cognitive behavior has to employ structural machines.

While the Church-Turing thesis is concerned with what can be computed, the Burgin Mikkilineni thesis is concerned with how artificial systems can mimic the autopoietic and cognitive behaviors found in living systems. Both these contribute to our understanding of computation and artificial intelligence but from different perspectives.

Similarly, while data structures are concerned with how data is organized and stored in a computer for efficient access and manipulation, knowledge structures are concerned with how information is organized and stored in a person's memory for efficient understanding and problem-solving.

Symbolic computing structures including sub-symbolic structures (which also are algorithms implemented using symbolic computing structures) and super-symbolic computing structures are both significant concepts in the field of computation, but they focus on different aspects. Whereas symbolic computing structures are concerned with manipulating mathematical expressions and other mathematical objects, super-symbolic computing structures are concerned with operating big

formal and informal systems of data and knowledge. Both structures contribute to our understanding of computation and artificial intelligence, but from different perspectives.

The common knowledge representation with entities, relationships, and event-driven behaviors provides a transparent model-based reasoning system with both memory and history based on time and associated events that change the system state as we have demonstrated with an example. Hopefully, this will stimulate more discussions and implementations that either validate or refute the general theory of information.

We conclude this paper with a quotation attributed to Charles Spurgeon [31,32]. "Wisdom is the right use of knowledge. To know is not to be wise. Many men know a great deal, and are all the greater fools for it. There is no fool so great a fool as a knowing fool. But to know how to use knowledge is to have wisdom."

6. Patents

We are in the process of applying for various patents based on our implementations of the theory based on specific domain knowledge representation and models.

Author Contributions: Conceptualization and the application of the General Theory of Information to design a digital software genome is carried out by Rao Mikkilineni. The implementation of the digital genome of a medical knowledge-based digital assistant is carried out by W. Patrick Kelly.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Acknowledgments: One of the authors, Rao Mikkilineni is grateful for many discussions and encouragement to pursue this research with Justin Kromelow from Opos.ai, Judith Lee from Golden Gate University, and Gordana Dodig-Crnkovic from Chalmers University.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Cameron Hunt. (2021) Event-Driven Architecture' Manifesto for Distributed Enterprise Applications.
2. https://blogs.sap.com/2021/01/24/event-driven-architecture-manifesto-for-distributed-enterprise-applications/ (Accessed on Nov 13, 2023)
3. P. Goyal and Rao Mikkilineni. Policy-Based Event-Driven Services-Oriented Architecture for Cloud Services Operation & Management. In Cloud Computing, 2009. CLOUD'09. IEEE International Conference on, pages 135–138, Sept 2009.
4. Kevin W. Hamlen, Lalana Kagal, and Murat Kantarcioglu. Policy Enforcement Framework for Cloud Data Management. IEEE Data Eng. Bull., 35(4):39–45, 2012.
5. Stopford, B. (2018). Designing Event-Driven Systems. O'Reilly Media.
6. Engel, Y., & Etzion, O. (2011, July). Towards proactive event-driven computing. In Proceedings of the 5th ACM international conference on Distributed event-based system (pp. 125-136).
7. Papazoglou, M. P., & Van Den Heuvel, W. J. (2007). Service-oriented architectures: approaches, technologies, and research issues. The VLDB Journal, 16, 389-415.
8. Dodig Crnkovic, G. Significance of Models of Computation, from Turing Model to Natural Computation. Minds Mach. 2011, 21, 301–322.
9. Mikkilineni, R.V. (2020). Going beyond Church–Turing Thesis Boundaries: Digital Genes, Digital Neurons and the Future of AI. Proceedings.
10. Mikkilineni R. Mark Burgin's Legacy: The General Theory of Information, the Digital Genome, and the Future of Machine Intelligence. Philosophies. 2023; 8(6):107. https://doi.org/10.3390/philosophies8060107
11. M. Burgin and R. Mikkilineni, "General Theory of Information Paves the Way to a Secure, Service-Oriented Internet Connecting People, Things, and Businesses," 2022 12th International Congress on Advanced Applied Informatics (IIAI-AAI), Kanazawa, Japan, 2022, pp. 144-149, doi: 10.1109/IIAIAAI55812.2022.00037
12. Burgin, M. Theory of Information: Fundamentality, Diversity, and Unification; World Scientific: Singapore, 2010.
13. Burgin, M. Theory of Knowledge: Structures and Processes; World Scientific Books: Singapore, 2016.
14. Burgin, (2012). M. Structural Reality; Nova Science Publishers: New York, NY, USA.
15. Mikkilineni, R. A New Class of Autopoietic and Cognitive Machines. Information 2022, 13, 24.
16. Bengio, Yoshua, Ian Goodfellow, and Aaron Courville. 2017. Deep Learning. Massachusetts: MIT Press.

17. Dodig-Crnkovic, Gordana. 2023. "How GPT Realizes Leibniz's Dream and Passes the Turing Test without Being Conscious" Computer Sciences & Mathematics Forum 8, no. 1: 66. https://doi.org/10.3390/cmsf2023008066
18. Damasio, A. (2010) Self Comes to Mind; Pantheon, a Division of Random House, Inc.: New York, NY, USA.
19. Dehaene, Stanislas. (2020). "How We Learn: Why Brains Learn Better Than Machine … for Now" Penguin Random House Inc., New York, NY.
20. Dehaene, S.; Naccache, L. Towards a cognitive neuroscience of consciousness: Basic evidence and a workspace framework. Cognition 2001, 79, 1–37.
21. Hawkins, J. A Thousand Brains: A New Theory of Intelligence; Basic Books: New York, NY, USA, 2021.
22. Maturana, H. R. & Varela, F. J. (1980). Autopoiesis and cognition: The realization of the living. Dordrecht et al.: Reidel (Boston Studies in the Philosophy and History of Science).
23. Meincke, A.S. Autopoiesis, biological autonomy and the process view of life. Euro Jnl Phil Sci 9, 5 (2019). https://doi.org/10.1007/s13194-018-0228-2
24. Shettleworth, S. J. (2010). Cognition, Evolution, and Behavior (2nd ed.). Oxford University Press
25. Burgin, M.; Mikkilineni, R. On the Autopoietic and Cognitive Behavior. EasyChair Preprint No. 6261, Version 2. 2021. Available online: https://easychair.org/publications/preprint/tkjk (accessed on 13 November 2023)
26. Burgin M. Information Theory: a Multifaceted Model of Information. Entropy. 2003; 5(2):146-160. https://doi.org/10.3390/e5020146
27. Mikkilineni, Rao. 2022. "Infusing Autopoietic and Cognitive Behaviors into Digital Automata to Improve Their Sentience, Resilience, and Intelligence" Big Data and Cognitive Computing 6, no. 1: 7. https://doi.org/10.3390/bdcc6010007
28. Burgin, M.; Mikkilineni, R. From data Processing to Knowledge Processing: Working with Operational Schemas by Autopoietic Machines. Big Data Cogn. Comput. 2021, 5, 13. https://doi.org/10.3390/bdcc5010013
29. Kelly WP, Coccaro F, Mikkilineni R. General Theory of Information, Digital Genome, Large Language Models, and Medical Knowledge-Driven Digital Assistant. Computer Sciences & Mathematics Forum. 2023; 8(1):70. https://doi.org/10.3390/cmsf2023008070
30. Nguyen, Thanh-Tung, Yu-Jin Yeom, Taehong Kim, Dae-Heon Park, and Sehan Kim. 2020. "Horizontal Pod Autoscaling in Kubernetes for Elastic Container Orchestration" Sensors 20, no. 16: 4621. https://doi.org/10.3390/s20164621
31. Burgin, M., & Mikkilineni, R. (2022). Seven Layers of Computation: Methodological Analysis and Mathematical Modeling. Filozofia i Nauka, 10, 11-32 (Mark Burgin & Rao Mikkililineni, Seven Layers of Computation: Methodological Analysis and Mathematical Modeling - PhilPapers, Accessed on November 14, 2023)
32. 130 Best Bible Verses About Wisdom And Knowledge (biblereasons.com) (Accessed on November 14, 2023)
33. You Don't Know What You Don't Know: Knowledge, Understanding, and Wisdom (tifwe.org) (Accessed on November 14, 2023)