

Article

Not peer-reviewed version

Machine Intelligence with Associative Memory and Event-Driven Transaction History

[Rao Mikkilineni](#)* and W. Patrick Kelly

Posted Date: 19 April 2024

doi: 10.20944/preprints202404.1298.v1

Keywords: Machine Intelligence; Associative Memory; Event-Driven Transaction History; Self-Regulation; Digital Genome; Autopoiesis; Cognition.)



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Machine Intelligence with Associative Memory and Event-Driven Transaction History

Rao Mikkilineni ^{1,*}, and W. Patrick Kelly ²

¹ Opos.ai, Golden Gate University, Dominican University of California, USA

² Opos.ai; pkelly@opos.ai

* Correspondence: rao@opos.ai

Abstract: Digital machine intelligence has evolved from its inception in the form of computation of numbers to AI, which is centered around performing cognitive tasks that humans can perform, such as predictive reasoning or complex calculations. The state of the art includes tasks that are 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., and also tasks that are easy to do “intuitively”, but are hard to describe formally or as a sequence of event-driven actions such as recognizing spoken words or faces. While these tasks are impressive, they fall short in applying common sense reasoning to new situations, filling in information gaps, or understanding and applying unspoken rules or norms. Human intelligence uses both associative memory and event-driven transaction history to make sense of what they are observing fast enough to do something about it while they are still observing it. In addition to this cognitive ability, all biological systems exhibit autopoiesis and self-regulation. In this paper, we demonstrate how machine intelligence can be enhanced to include both associative memory and event-driven transaction history to create a new class of knowledge-based assistants to augment human intelligence. The digital assistants use global knowledge derived from the Large Language Models to bridge the knowledge gap between various participants interacting with each other. We use the general theory of information and schema-based knowledge representation to create the memory and history of various transactions involved in the interactions.

Keywords: Machine Intelligence; Associative Memory; Event-Driven Transaction History; Self-Regulation; Digital Genome; Autopoiesis; Cognition.)

1. Introduction

Machine intelligence, also called artificial intelligence (AI), refers to the capability of digital machines or computer systems to perform tasks that typically require human intelligence. These tasks include reasoning, problem-solving, learning, perception, language understanding, and decision-making. Alan Turing [1] defined digital computers by analogy to a human, computing numbers following fixed rules (known as the Turing Machine). John von Neumann's [2] stored program control implementation of the Turing Machine uses a sequence of symbols (1s and 0s) to execute the task that defines operations as fixed rules on the data, that is also represented as sequences of symbols, consisting of the information about the state of some entities and their relationships. In the case of computing numbers, the entities are the numbers and their relationships which are defined by a combination of the operators "+" and "-". The digital computer thus consists of a store of information (memory), an executive unit (CPU) that performs individual operations defined by the program on the data, and a control mechanism ensuring instructions follow rules and order.

All biological systems derive their knowledge of life processes from their genomes enabling them to build, operate, and manage their physical and mental structures to process information, make sense of what they are observing fast enough to do something about it while they are still observing it. Information processing, communication, cognition which is the ability to update and use knowledge, and consciousness which is the ability to perceive “self” and identity from a system point

of view with self-regulation of various autonomous components participating in the execution of life processes [3,4].

On the other hand, machine intelligence evolved from computation as calculation or operations undertaken on numbers to today's AI which has seen explosive growth, particularly in generative AI (gen AI) which creates new content based on what it has learned from existing content. GenAI produces various types of content, images, text, audio, and synthetic data. The machine intelligence is derived from two distinct computational processes. The first process executes tasks defined as algorithms operating on data structures. These tasks allow us to not only build systems that communicate, collaborate, and conduct commerce at scale but also interface with the physical world using analog sensors and actuators, thus augmenting our bodily functions to interact with the external world. The second process uses an algorithm that mimics the neural network structures in our brain which process information received from various senses. These "deep learning" neural networks allow us to convert audio, images, video, and natural language into knowledge that can be accessed through queries. Generative AI augments our knowledge by processing information at a scale and speed that is not possible to process by any single individual [5].

While these structures have been very successful in implementing process automation and data-driven intelligent decision-making, they fall short of mimicking living organisms in two significant ways.

First, biological systems are autopoietic and use encoded process information to organize and manage their components which are interacting with each other and their environment. The self-organizing and self-regulating patterns sense and counteract fluctuations to maintain their stability [3]. On the other hand, current digital computing structures when deployed as a distributed system with asynchronous communication between them, are not stable under large fluctuations in their resource availability or demand [6–8]. An application often is executed by several distributed software components using computing resources often owned and managed by different providers and the assurance of end-to-end process sustenance with adequate resources, its stability, safety, security, and compliance with global requirements requires a complex layer of additional processes that increase complexity leading to 'who manages the managers' conundrum. In addition, the CAP theorem [9] limitations force, in the event of a network partition, a choice between consistency and availability. Database systems designed with traditional ACID guarantees [10], such as RDBMS, choose consistency over availability, whereas systems designed around the BASE philosophy, common in the NoSQL movement, choose availability over consistency. In this paper, we discuss how to implement autopoietic digital automata using the tools derived from the General Theory of Information (GTI) [11–14] which provides self-regulation of the system and its components and overcomes the current limitations.

Second, biological cognition consists of not only processing information and converting it into knowledge structures from the various senses using its cortical columns, (neurons that fire together wire together), but also uses higher-level knowledge representation in the form of associative memory and event-driven transaction history [15] of all the entities, their relationships, and behaviors the system has encountered to make sense of its current state and take appropriate action based on its experience and common sense to evolve its state. In the context of cognitive behaviors, "common sense" here refers to the innate ability to perceive, understand, and judge situations in a way that is consistent with the external reality and common knowledge of many people. The deep learning algorithms [16] currently provide knowledge representation in the form of hidden parameters (known as black box representation of knowledge) and the knowledge is accessed through prompts or queries. Higher-level reasoning is required to integrate the knowledge obtained through several neural networks and the higher-level knowledge about systemic goals, constraints, and the demand for and the availability of resources to make sense of what is being observed and take action based on experience, wisdom, and common sense while balancing various factors that affect the future state of the system in real-time. In this paper, we discuss how to implement associative memory and the event-driven transaction history that integrates knowledge from deep learning networks and systems goals defined in the form of functional and non-functional requirements and experience-based best

practice policies for self-regulation. We define a digital genome that is based on the system's functional and non-functional requirements and experience-based best practice policies for self-regulation.

In Section 2, we introduce information processing structures derived from the General Theory of Information and describe the knowledge representation that integrates the knowledge from both algorithmic computations and deep learning. In Section 3, we describe how autopoietic and cognitive behaviors are implemented. In Section 4, we demonstrate the use of digital associative memory and event-driven transaction history with a simple example application. In Section 5, we discuss the implications of the digital genome design and its impact on the evolution of digital machine intelligence and its use to augment human intelligence. In section 6, we discuss the key differences between our approach and the current state-of-the-art AI. In Section 7, we suggest some future directions based on our experience in implementing a digital genome proof of concept (PoC) to reduce the knowledge gap between a doctor and a patient in the early diagnosis process using the medical knowledge derived from the Large Language models and other sources.

2. Information Processing and Knowledge Networks

The General Theory of Information (GTI) and the Burgin Mikkilineni thesis provide a framework for understanding information processing and knowledge structures. The GTI, developed by Mark Burgin [8,11,17], is an approach that offers powerful tools for all areas of information studies. It 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. The second class explains how to measure information.

The Burgin Mikkilineni thesis [13,17] focuses on the autopoietic and cognitive behavior of artificial systems. The ontological part of the thesis states that these behaviors must function on three levels of information processing systems and be based on triadic automata. The axiological part of the thesis states that efficient autopoietic and cognitive behavior has to employ structural machines. Ontological and axiological principles [11,17,18], are two fundamental concepts in philosophy, each addressing different aspects of knowledge and reality. Ontological principles deal with the nature of reality and being. For instance, in the context of GTI, ontological principles reveal general properties and regularities of information and its functioning. In the context of GTI, axiological principles explain what measures of information are, the nature of reality, and what constitutes knowledge, and guide us in understanding the value of that knowledge and how it should be interpreted and evaluated.

In essence, these theories provide a robust foundation for understanding how information is processed and how knowledge structures are derived, particularly in the context of artificial systems. They offer insights into the essence of information, the principles underlying its functioning, and the mechanisms for its measurement. They also shed light on the autopoietic and cognitive behaviors of artificial systems [17], emphasizing the importance of triadic automata and structural machines. In this section, we will summarize the basic tools derived from GTI used to infuse autopoietic and cognitive behaviors into digital automata.

2.1. Structural Machines

Structural machines are a new model of computation derived from the General Theory of Information (GTI). They can process information in the form of structures, not just sequences of symbols. These machines represent knowledge structures as schema and perform operations on them to evolve information changes in the system from one instant to another when any of the attributes of any of the objects change.

In terms of designing autopoietic and cognitive digital automata [13,14,17], the super-symbolic computing structures suggested by GTI provide an overlay over symbolic and sub-symbolic computing structures creating a common knowledge representation. This approach allows us to not only model but also implement digital automata with autopoietic and cognitive behaviors. Autopoiesis enables digital automata to use the specifications in their digital genomes to instantiate

themselves using matter and energy transformations. Cognition allows 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. A digital genome is a model that specifies the operational knowledge of algorithms executing software life processes [8,17]. It is derived from our understanding of genomics, neuroscience, and GTI. The digital genome provides the operational knowledge to implement software life processes with specific purposes using software replication and hardware metabolism (using CPU, memory, and energy) providing computing resources. This concept is inspired by the biological genome, which provides the operational knowledge for biological systems to build, operate, and manage life processes using cells capable of replication and metabolism.

Super-symbolic computing differs from neuro-symbolic computing suggested for implementing Artificial General Intelligence (AGI) in its methodology and underlying principles. Neuro-symbolic computing (NeSy) [19] is an active research area in AI that aims to integrate the symbolic and statistical paradigms of cognition. It combines traditional rule-based AI approaches with modern deep learning techniques. The symbolic systems in NeSy can make explicit use of expert knowledge and are to a high extent self-explanatory. The neural aspect of computation dominates the symbolic part in cases where they are clearly, separable. NeSy shows promise in reconciling the advantages of reasoning and interpretability of symbolic representation and robust learning in neural networks. However, both current symbolic and sub-symbolic computing structures are software algorithms implemented using sequences of symbols (programs) operating on another sequence of symbols (data structures) and are subject to Church-Turing Thesis [20] limitations dealing with large fluctuations in resource demand and availability supporting a distributed autonomous software components communicating with asynchronous communication.

Conversely, super-symbolic computing, derived from the General Theory of Information (GTI), provides an overlay over symbolic and sub-symbolic computing structures with a common knowledge representation. It allows us to not only model but also implement digital automata with autopoietic and cognitive behaviors. Super-symbolic computing structures represent knowledge structures as schema and perform operations on them to evolve state changes in the system. Structural machines overcome the limitations of the Church Turing Thesis and provide a path to implementing autopoietic and cognitive behaviors in digital automata. In addition, the structural machines and the digital genome provide a unique identity to the software system composed of autonomous entities deployed in distributed geographies utilizing different computing resources communicating asynchronously with shared knowledge and provide a schema that supports the design, development, deployment, and operation of both the computer and the computed. The digital genome specifies the life processes that specify and execute functional and non-functional requirements and the best-practice policies based on experience to maintain stability, safety, security, and survival, just as the biological genome does.

2.2. Knowledge Structures and the Knowledge Network

Knowledge Structures derived from the General Theory of Information (GTI) are representations of operational knowledge that evolve the state of a system. They are derived from our understanding of genomics, neuroscience, and GTI. Knowledge structures represent the operational knowledge that changes the state of the system from one instant to another when any of the attributes of any of the objects change. They are used to model a digital genome specifying the operational knowledge of algorithms executing the software life processes with specific purposes using software replication and hardware metabolism.

A Knowledge Network, on the other hand, is a network of these knowledge structures. The information accumulated through experience as domain knowledge is encoded into knowledge to create the digital genome which contains the knowledge network defining the function, structure, and autopoietic and cognitive processes to build and evolve the system while managing both deterministic and non-deterministic fluctuations in the interactions.

In terms of designing autopoietic and cognitive digital automata [8,17], knowledge structures and networks provide a common knowledge representation from existing symbolic and sub-symbolic computing structures to implement autopoiesis and cognitive behaviors. Autopoiesis enables digital automata to use the specifications in their digital genomes to instantiate themselves using matter and energy transformations. Cognition allows 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. Therefore, if machines are to mimic living organisms, they must be infused with autopoietic and cognitive behaviors.

2.3. Cognizing Oracles

Cognizing Oracles is a concept derived from GTI and used in the construction of autopoietic and cognitive digital machines. They contribute to super-symbolic computing along with knowledge structures and structural machines.

They are different from the Turing Oracle Machines [8,17] which are abstract machines used to study decision problems. They can be visualized as a Turing machine with a black box, called an oracle, which can solve certain problems in a single operation. The problem can be of any complexity class. Even undecidable problems, such as the halting problem, can be used. The key difference between cognizing oracles and Turing oracle machines lies in their purpose and the problems they are designed to solve. While Turing oracle machines are theoretical constructs used to study decision problems, including undecidable ones, cognizing oracles are used in practical applications to change the behavior of software components. Cognizing oracles, along with knowledge structures and structural machines, are used to model a digital genome [8,17] specifying the operational knowledge of algorithms executing software life processes. This results in a digital software system with a super-symbolic computing structure exhibiting autopoietic and cognitive behaviors that biological systems also exhibit. In essence, cognizing oracles play a crucial role in the practical applications of GTI, particularly in the evolution of machine intelligence.

3. Autopoietic and Cognitive Behaviors

Autopoietic and cognitive behaviors are key aspects of living organisms that have been studied for their potential to be infused into digital automata. Autopoiesis refers to the behavior of a system that replicates itself and maintains identity and stability while facing fluctuations caused by external influences. It enables living beings to use the specification in their genomes to instantiate themselves using matter and energy transformations. Cognitive behaviors [21], on the other hand, model the system's state, sense internal and external changes, and analyze, predict, and take action to mitigate any risk to its functional fulfillment. Cognition allows living beings to process information into knowledge and use it to manage their interactions between various constituent parts within the system and their interaction with the environment. In essence, the genome is a blueprint that prescribes the processes to execute functional and nonfunctional requirements of the organism and the best practice policies that maintain stability, safety, security, and survival of the system based on past experiences and knowledge transmitted by the survivor to the successor.

In the context of distributed software application development, deployment, operation, and life-cycle management, a digital genome is a comprehensive digital representation of a software system's operational knowledge. It specifies the "life" processes of a distributed application where the functions, structure, and means for dealing with fluctuations are encoded. In terms of software development, the digital genome specifies the knowledge to execute various tasks that implement functional requirements. It provides the operational knowledge of algorithms executing the software life processes with specific purposes using replication of software and metabolism with hardware to obtain the required computing resources [8,17].

4. Associative Memory and Event-Driven Transaction History

Associative Memory [22–24] refers to the ability to learn and remember the relationship between unrelated items. This type of memory deals specifically with the relationship between different objects or concepts. For example, remembering the name of someone or the aroma of a particular perfume. In humans, this relates to sensory 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). Associative memory is thought to be mediated by the medial temporal lobe of the brain.

Event-Driven Transaction History [25–27] is a concept that comes from the field of cognitive science and artificial intelligence. It refers to the idea that humans accomplish event segmentation as a side effect of event anticipation. This means that humans are constantly predicting what will happen next, and when something unexpected happens, it triggers a new event segment. This process is thought to be crucial for understanding complex activities and making sense of the world.

Both associative memory and event-driven transaction history play significant roles in human memory, knowledge representation, and intelligence [28]. Associative memory is a key component of human memory, allowing us to form connections between different pieces of information and recall them later. Event-driven transaction history, on the other hand, helps us segment our experiences into meaningful events, which can then be stored and recalled as needed. Associative memory allows us to form and represent knowledge as a network of interconnected concepts. This is crucial for understanding complex ideas and solving problems. Event-driven transaction history [29] helps us understand and represent the temporal structure of our experiences, which is important for tasks like planning and decision-making. Both associative memory and event-driven transaction history contribute to our ability to learn from our experiences, make predictions, and adapt our behavior based on the outcomes of past actions. These abilities are central to human intelligence.

In this paper, we demonstrate using a textbook example of machine intelligence, how a digital genome and the implementation of associative memory and event-driven transaction history improve the reasoning and predictive processes with common knowledge representation derived from multiple information sources.

5. Digital Genome and Machine Intelligence: Illustration with a Textbook Exercise

This exercise will demonstrate the process of converting a common data storage source with historical data into an event-based transaction history and associative memory. This implementation of the General Theory of Information creates a distributed cloud architecture with a Digital Genome that dictates the functional and non-functional requirements as well as the best practices for the operation of these applications. By providing a structured data format and leveraging LLM (Large Language Models) technology it has become possible to identify the entities, relationships, and behaviors of the acting agents within the structured data. From this information, it becomes possible to isolate the functional operations for the interaction of the entities within the data. These functional operations can be expressed through computer code and executed in individualized cloud environments. These isolated functions communicate across a network through a series of API connections that knit these simple functions into a tapestry of complex action that can be tracked and logged through time with the utilization of a graph database system to represent the actions and interactions of.

Given data from a CSV file we can isolate the column heading labels as separate entities. These entities can be interpreted, based on the context of the data, to define the relationships that are based on the interaction of these entities. As these relationships interact the behavior of these entities emerges from the transformation of data over time.

Through the practice of the conversion of data tables into event-driven transaction history trees, we can generate an associative memory of the state of data through time [30,31]. This allows us to traverse the event transactions from any historical state to gain a granular perspective on the behavioral interactions and their effect on the greater knowledge. In this example of an event-driven transaction transformation we used the data file from the text “Data Science Projects with Python” [32] provided through the GitHub repository [33].

This data contains credit card customer information over six months. Over this time several variables are correlated to predict the likelihood of whether any given customer would default on the credit card payment for the seventh month. We are using data that has been cleaned by the original author but still contains some inconsistencies and inaccuracies.

Within our CSV file, we have 26664 instances for a variety of customers. There are 29 columns of variables that contain information related to the monthly account activity for each customer. Through the context of this data, we can interpret this information through a series of months labeled 6 through 1, with the heading 'PAY_AMT6' representing the amount the customer paid in April and 'PAY_AMT1' representing the amount the customer paid in September. The data provided in this file contains columns on customer information (ID, Sex, Education, Marriage, Age), billing information (Bill_Amt), payment information (Pay_Amt, Pay), and account information (Limit_Bal, default payment next month). With the provided information we can begin incorporating the ontology of the knowledge domain to determine the relationships and behaviors of the system.

By consulting a variety of traditional ontological information sources [34], along with the aid of LLM's, we can gain information on the specific knowledge domain to derive schema definitions for our entities, relationships, and their behaviors. From the text [32, p 33] we can find an explanation of our data provided by the client shown below in Table 1.

Table 1. The Dataset information.

EDUCATION	Education level (1 = Grad School; 2 = University; 3 = Highschool)
MARRIAGE	Marital Status (1 = Married; 2 = Single, 3 = others)
AGE	Age (years)
PAY_1 – PAY_6	Past payments (PAY_6 = April; PAY_5 = MAY)
BILL_AMT1 – BILL_AMT6	Bill statement amount (BILL_AMT6 = April)
PAY_AMT1 – PAY_AMT6	Previous payment amount (PAY_AMT6 = April)
default payment next month	Predicted default for next month (1 = yes; 0 = no)

From the client's explanation, we can see that we have been provided data with a mix of numerical and categorical values. The categorical values are expressed numerically throughout the data which necessitates further explanation to provide the proper context to interpret the meaning of these values. The provided explanation dictates the categories that are expressed in PAY_1 – PAY_6 is in a range from -2 to 8. Each numerical value represents a different payment status of the customer's account whereas -2 represents an account that started the month with a zero balance and there was no account activity in that month. -1 represents an account where the last month's balance was paid in full. 0 represents an account where the minimum payment was made, but the entire balance was not paid, resulting in the positive balance being carried over to the next month.

1 through 8 indicates the number of months the account is past due on payment. By providing the ontological context with the categorical definitions and an example of the data in CSV format to an LLM we can leverage external processes to aid in entity identification.

Table 2. The schema definition.

<u>Entities</u>	<u>Attributes</u>
Month	Month name
Billing	Bill Amount

Customer	Education, Age, Sex, Marriage
Account	Limit Balance, Default Status
Payment	Pay, Pay Amount

Through this process, we can derive the relationships between these entities along with their behaviors. This information can then be used to define a schema for a graph database system with each entity interpreted as a vertex object containing the attributes that apply to each specific entity. The edge connections between the vertices dictate the relationships and behaviors between the entities.

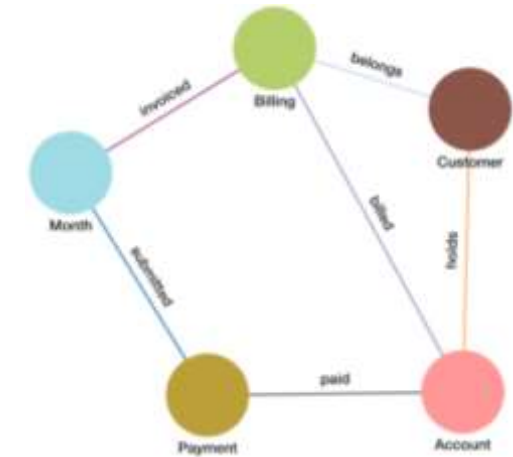


Figure 1. The schema definition.

These changes are logged within the event entities as a series of sensors and actuators that dictate the causal relationship between the sending and receiving entities. These event entities are provided a timestamp when they are instantiated. This process creates a knowledge network based on the actions and reactions of the system [32] from which we can create specific queries to recall past states of the data graph for time series analysis.

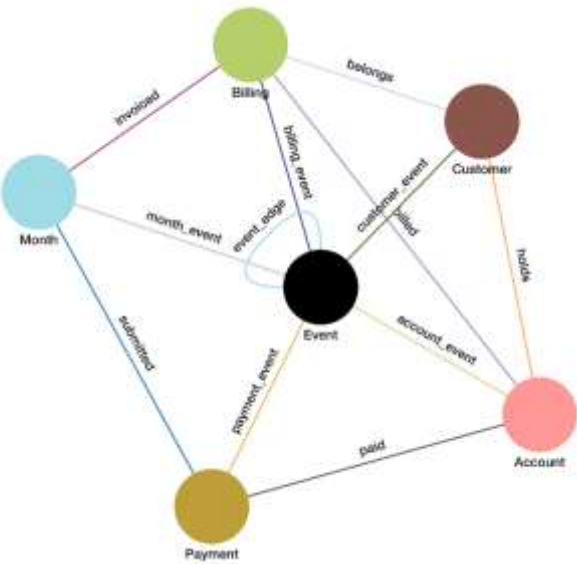


Figure 2. Schema Definition with Event as an Entity with time sequence.

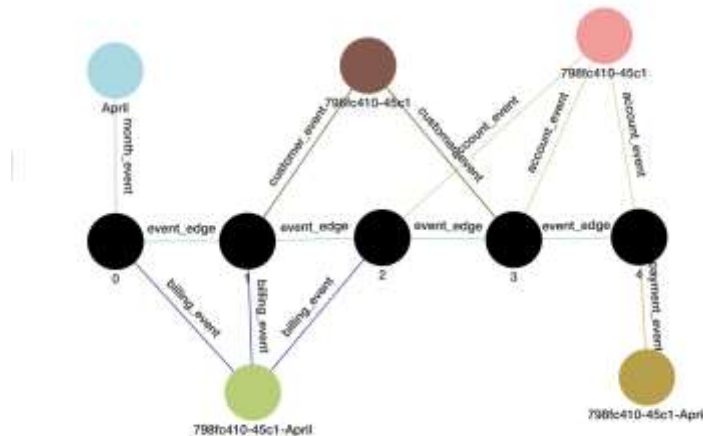


Figure 3. One instance of a single user through events.

From our schema definition, we can define each entity's functional requirements as a process to perform its specific task. These processes are described, executed, and transmitted across the network through a series of API interactions with every other entity node. Each entity process is provided through a server node. Each node is containerized through Docker [35] and deployed to a cloud environment that provides memory, computing power, and other non-functional requirements to maintain the environment in which the functional requirements are executed. This server node network is managed through the dictated policies of a Kubernetes [36] layer that handles the resource maintenance for each cloud environment along with the security to balance the consistency, availability, and partition tolerance throughout the network.

This Kubernetes layer is part of the Autopoietic Manager whose function is to set up the non-functional requirements in the cloud environment and populate the functional nodes of our system within their respective environments. Each functional layer is prescribed a management policy that provisions computing resources to optimize performance and fault management across a distributed infrastructure detecting, isolating, and correcting malfunctions in the network. The Autopoietic Manager includes activities such as monitoring the distributed application components independent of the infrastructure management systems that manage the infrastructure components, receiving notifications of faults, and troubleshooting to resolve issues. With a system of configuration management within the Autopoietic Manager, the changes in the application network components are controlled and monitored to restrict and restructure the state of our distributed application depending on the best-practice-based policies defined in the digital genome. Accounting management is dictated within these policies to manage the resource distribution of CPU utilization of various application components to optimize the desired performance state under any circumstances (i.e., auto-failover, auto-scaling, live-migration, etc.) are managed at the application component layer independent of infrastructure on which they are executed. Optimization performance metrics are used to manage automated actions that improve performance, for example, based on end-to-end response time. The security of our system is also dictated through these policies to protect against unauthorized access and data breaches through the implementation of an intrusion detection system and access control policies managing the application component communications. All application component edges are connected to communicate using private and public keys.

Each distributed application component node stands as its own independent cloud application (as a service) whose non-functional requirements are managed by the Autopoietic Manager [8,17,37]. The API connections between these nodes create a network of wired actions that fire together to perform specific tasks through their defined functional requirements. The Cognitive Network Manager (CNM) [8,17,37] is an intermediary API server within its own cloud environment that manages the connections between these functional operation nodes. Acting as a switchboard operator the CNM connects the signals for communication without needing to know the explicit messages for

the entities it is communicating with across the network. This allows for the reassignment of addresses as multiple node instances are dynamically deployed depending on the demand of the system at any given time. The autopoietic manager and the cognitive network manager work together to implement the functional and non-functional requirements specified in the digital genome using the policies prescribed.

A user interacts with this system through a front-end user interface application. As that front-end application establishes its connection to the network the only node it can communicate with is the CNM. The API request signal from the frontend application informs the CNM of the functional requirements involved in performing the specific task and responds with the respective service URLs of each functional entity that is required to complete the request. This same action is true for each subsequent task throughout the network. For any task that is dependent on the cooperation of two or more independent node functions, the communication path between these acting nodes is routed through the CNM. This allows for a subtractive approach when passing data between these entities. Through the CNM the data that is sent to each functional process can be stripped of sensitive or unnecessary information to maintain a policy of data sharing on a need-to-know basis. This hierarchy of communication ensures not only efficient data transmission but also enhances security by controlling the flow of information within the network.

By centralizing the management of inter-node communication the CNM serves as a gatekeeper facilitating seamless interaction while maintaining strict control over data access and distribution. Moreover, the abstraction provided by routing communication through the CNM allows for a more modular and scalable network architecture, where nodes can be added or modified without disrupting the overall system functionality. In essence, the CNM acts as a crucial intermediary, orchestrating the flow of data and ensuring the integrity and confidentiality of information exchange across the network.

6. Discussion

In this paper, we have used knowledge from multiple domains to formulate a new approach to model a system's behavior using digital machines and use it to derive various results just as human beings do using their mental models of the system's behavior that includes themselves and their interactions with the external world. Autopoietic and cognitive behaviors are examined in depth, emphasizing their importance for replicating a living organism's ability to maintain process stability, information workflow, and adapt to the environment, and meet challenging fluctuations in resource demand and availability. The concept of a digital genome is introduced as a comprehensive representation of a software system's operational knowledge, guiding the execution of functional requirements, non-functional requirements, and best practices. Associative memory and event-driven transaction history are explored as essential components of human memory and intelligence, with implications for machine learning and decision-making. In this paper, we demonstrate how to build them using domain models with entities, relationships, and their behaviors as events change the state of the system. The methodology uses structural machines, cognizing oracles, and knowledge structures. A detailed illustration using a textbook example demonstrates how these concepts can be applied to improve reasoning and predictive processes in machine intelligence.

We have used knowledge from multiple domains to formulate a new approach to model a system's behavior using the digital genome and use it to build, deploy, operate, and manage the distributed application using various infrastructure resources to derive various results just as human beings do using their mental models of the system's behavior which includes themselves and their interactions with the external world. Associative memory is the human ability to learn and remember the relationship between unrelated items which is a key component of human memory, allowing us to form connections between different pieces of information and recall them later. Event-driven transaction history is the concept from cognitive science that refers to the idea that humans accomplish event segmentation as a side effect of event anticipation. It helps us segment our experiences into meaningful events, which can then be stored and recalled as needed. Both associative memory and event-driven transaction history play significant roles in human memory,

knowledge representation, and intelligence. They contribute to our ability to learn from our experiences, make predictions, and adapt our behavior based on the outcomes of past actions. The integration of associative memory and event-driven transaction history into machine intelligence reflects a deeper understanding of human cognitive processes. These mechanisms not only contribute to human memory but also play crucial roles in knowledge representation and intelligence. By leveraging these concepts, machines can emulate human-like learning from experiences, anticipate future events, and adjust behavior accordingly. This holistic approach bridges the gap between artificial and human intelligence, paving the way for more sophisticated and adaptive machines.

We discuss how we can design a digital genome, a concept derived from the general theory of information and Burgin-Mikkilineni thesis, and implement associative memory and event-driven transaction history to improve the reasoning and predictive processes in machine intelligence. We highlight the use of common knowledge representation derived from multiple information sources in the implementation of associative memory and event-driven transaction history in machine intelligence. We also discuss structural machines as a model of computation derived from GTI and highlight their role in implementing autopoietic and cognitive behaviors in digital automata. By harnessing this framework, we illustrate how associative memory and event-driven transaction history can be seamlessly integrated into the fabric of machine intelligence.

We use a machine learning example discussed in a textbook to contrast the conventional approach and the new approach highlighting the contrast and benefits. We also discuss how this approach is different from the current proposals of neuro-symbolic computing and emphasize that the super-symbolic approach derived from the general theory of information addresses some of the shortcomings of Church-Turing thesis boundaries and the computational limits of symbolic and sub-symbolic algorithms that depend on the operations on a sequence of symbols (data structures) using another sequence of symbols (software programs). The structural machines, cognizing oracles, and knowledge structures derived from the general theory of information provide new knowledge representation as a network that integrates the knowledge from symbolic and sub-symbolic computing structures and a systemic view of the state and its evolution in the form of associative memory and the even-based transaction history. By moving beyond the sequential symbol manipulation, our framework fosters a more holistic and nuanced understanding of complex systems using schema-based.

While this paper focuses on a textbook example to illustrate the approach that is not generally known to the current AI practitioners, proofs of concept are being developed to demonstrate its validity, an example of which is a medical-knowledge-driven digital assistant helping to reduce the knowledge gap between the patient, doctors, and various other actors participating in the early disease diagnostic process.

7. Conclusions

We have presented an approach using associative memory and event-driven transaction history to augment current AI using symbolic and sub-symbolic computing structures. The new approach is derived from the general theory of information and uses super-symbolic computing with structural machines which are discussed in several papers and books referenced in this paper. The focus of the study is on the data that belongs to the individuals and the behavioral patterns discerned using the schema derived from the domain knowledge. Each pattern is unique depending on many factors that contribute to the behavior. For example, credit default can occur for many reasons and each case may be different depending on the circumstances and external events. For example, a temporary change in the financial condition may force the default to occur or a serial non-payment may trigger a credit risk. The patterns derived from the associative memory and the transaction history identify the anomalies in individual cases and provide more knowledge to address the issue at hand. In addition, the group data can be derived from the individual histories and the usual machine learning algorithms can be applied to derive the group patterns. It helps to differentiate those who are chronically at high risk of defaulting, those who are temporarily unable to pay, and those who are

“good” or preferred customers. This will allow for taking preventive actions and dealing with special cases.

Central to our study is the analysis of individual data and behavioral patterns, which serve as the cornerstone of our approach. By extrapolating insights from domain knowledge, we construct schemas that identify the unique nuances underpinning each behavioral pattern. For instance, credit default occurrences may stem from a myriad of factors, such as transient financial fluctuations or persistent non-payment trends. By delineating these patterns through associative memory and transaction history, our framework excels in identifying anomalies within individual cases, thereby furnishing practitioners with actionable insights to address underlying issues proactively. Our methodology extends beyond individual analyses to encompass group-level insights derived from collective histories. By applying traditional machine learning algorithms to group data, we delineate distinct categories within the customer base—ranging from chronic default risks to preferred clientele. This segmentation empowers stakeholders to institute targeted interventions tailored to specific risk profiles, thus fostering a more proactive and nuanced approach to risk management.

We believe that the schema-based approach with operations on structures (nodes executing processes and edge communicating with other components based on shared knowledge, not only improves the current design, deployment, operation, and management of distributed applications but also allows us to derive reasoning based on a transparent model that provides associative memory and the history of all entities involved. It eliminates the black-box nature of current AI practices. This approach augments conventional AI frameworks by infusing them with a deeper understanding of individual and collective behaviors. By discerning patterns, anomalies, and trends at both micro and macro levels, our methodology equips practitioners with the foresight and adaptability needed to navigate dynamic real-world scenarios effectively. In addition, the autopoiesis of the system eliminates the Church-Turing thesis boundaries and CAP theorem limitations. It also goes beyond the limitation mentioned in “Computation and its Limits” [38] last paragraph of the last chapter by including both the computer and the computed in the schema.

The concept of the universal Turing machine has allowed us to create general-purpose computers and “use them to deterministically model any physical system, of which they are not themselves a part to an arbitrary degree of accuracy. Their logical limits arise when we try to get them to model a part of the world that includes themselves.”

Author Contributions: Both authors are equally involved in designing and implementing the autopoietic and cognitive computing structures derived from the general theory of information.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Acknowledgments: The authors acknowledge various discussions with Max Michaels and Gorda Dodig Crnkovic.

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

Glossary:

Digital Genome – A comprehensive digital representation of a software system's operational knowledge, analogous to the biological genome, which specifies the "life" processes of the system.

General Theory of Information - A theoretical framework developed by Mark Burgin that provides a comprehensive approach to understanding information and its processing.

Autopoietic manager – A management system or entity that exhibits autopoietic behavior. Autopoiesis, a concept originating from biology, describes the self-replicating and self-maintaining nature of living organisms.

Cognitive network manager – A management system or entity that utilizes cognitive capabilities to oversee and optimize networks. This concept integrates principles from cognitive science and artificial intelligence with network management to create intelligent, adaptable, and efficient network management systems.

Structural machine – A computational model derived from the General Theory of Information (GTI) that processes information in the form of structures rather than just sequences of symbols. This concept extends the traditional notion of computation by allowing machines to manipulate and operate on complex structures, such as graphs, networks, or hierarchies, which more accurately represent the relationships and interactions present in real-world systems.

References

1. Turing, A. M. (1936). "On Computable Numbers, with an Application to the Entscheidungsproblem". Proceedings of the London Mathematical Society, Series 2, Vol. 42
2. Von Neumann, J. (1945). "First Draft of a Report on the EDVAC". IEEE Annals of the History of Computing.
3. Maturana, H. R., & Varela, F. J. (1980). "Autopoiesis and cognition: The realization of the living." Springer Science & Business Media.
4. Francisco J. Varela: Principles of Biological Autonomy. The North-Holland Series in General Systems Research, Vol. 2. New York: Elsevier North-Holland, Inc., 1979, 306 pp
5. Radford, A., et al. (2019). "Language Models are Unsupervised Multitask Learners."
6. Lynch, Nancy A., (1996), Distributed Algorithms, Morgan Kaufmann Publishers Inc, San Francisco, CA, USA, ISBN 9780080504704.
7. Chandy, K. M., & Lamport, L. (1985). "Distributed snapshots: Determining global states of distributed systems." ACM Transactions on Computer Systems (TOCS), 3(1), 63-75.
8. Mikkilineni, Rao. 2023. "Mark Burgin's Legacy: The General Theory of Information, the Digital Genome, and the Future of Machine Intelligence" Philosophies 8, no. 6: 107. <https://doi.org/10.3390/philosophies8060107>
9. Brewer, E. A. (2012). CAP twelve years later: How the "rules" have changed. Computer, 45(2), 23-29.
10. Gray, J., & Reuter, A. (1993). Transaction processing: concepts and techniques. Morgan Kaufmann Publishers Inc, San Francisco, CA, USA.
11. Burgin, M. (2010). Theory of Information: Fundamentality, Diversity and Unification. World Scientific Publishing Company.
12. Burgin, M., & Mikkilineni, R. (2014). Autopoietic and Cognitive Behavior of Information Processing Systems. In Structural Information and Communication Complexity (pp. 13-33). Springer.
13. Mikkilineni, Rao. 2022. "A New Class of Autopoietic and Cognitive Machines" Information 13, no. 1: 24. <https://doi.org/10.3390/info13010024>
14. 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>
15. Wang JH, Cui S. Associative memory cells and their working principle in the brain. F1000Res. 2018 Jan 25;7:108. doi: 10.12688/f1000research.13665.1. PMID: 29487741; PMCID: PMC5806053.
16. LeCun, Y., Bengio, Y., & Hinton, G. (2015). "Deep learning." Nature, 521(7553), 436-444.
17. M. Burgin and Mikkilineni R., (2022). Information Theoretical Principles of Software Development, EasyChair Preprint no. 9222, <https://easychair.org/publications/preprint/jnMd> (Accessed on April 15, 2024).
18. Burgin, M. (2016). Theory of Knowledge: Structures and Processes; World Scientific: New York, NY, USA;
19. Luo, X., Li, H. & Lee, S. Bridging the gap: Neuro-Symbolic Computing for advanced AI applications in construction. Front. Eng. Manag. 10, 727–735 (2023). <https://doi.org/10.1007/s42524-023-0266-0>
20. M. Burgin and R. Mikkilineni, "General Theory of Information Paves the Way to a Secure, Service-Oriented Internet Connecting People, Things, and Businesses," in 2022 12th International Congress on Advanced Applied Informatics (IIAI-AAI), Kanazawa, Japan, 2022 pp. 144-149. doi: 10.1109/IIAIAAI55812.2022.00037
21. Mingers, J. (1995). Self-producing systems: Implications and applications of autopoiesis. Springer Science & Business Media.
22. Baars, B. J., & Gage, N. M. (2010). Cognition, Brain, and Consciousness: Introduction to Cognitive Neuroscience. Academic Press.
23. Jablonka, E., & Lamb, M. J. (2005). Evolution in Four Dimensions: Genetic, Epigenetic, Behavioral, and Symbolic Variation in the History of Life. MIT Press.
24. Medvidovic, N., & Taylor, R. N. (2010). Software Architecture: Foundations, Theory, and Practice. John Wiley & Sons.
25. Bass, L., Clements, P., & Kazman, R. (2012). Software Architecture in Practice. Addison-Wesley.
26. Eichenbaum, H., & Cohen, N. J. (2001). From Conditioning to Conscious Recollection: Memory Systems of the Brain. Oxford University Press.
27. Baddeley, A., Eysenck, M. W., & Anderson, M. C. (2009). Memory. Psychology Press.
28. Collins, A. M., & Loftus, E. F. (1975). A spreading-activation theory of semantic processing. Psychological Review, 82(6), 407-428.

29. Zacks, J. M., & Tversky, B. (2001). Event Structure in Perception and Conception. *Psychological Bulletin*, 127(1), 3-21.
30. Kurby, C. A., & Zacks, J. M. (2008). Segregation of the Neural Correlates of Physical and Kinematic Event Segmentation. *Journal of Cognitive Neuroscience*, 20(5), 923-937.
31. Kiefer, M., & Pulvermüller, F. (2012). Conceptual representations in mind and brain: Theoretical developments, current evidence, and future directions. *Cortex*, 48(7), 805-825.
32. Klosterman, Stephen. *Data Science Projects with Python: A case study approach to successful data science projects using Python, pandas, and scikit-learn*. Packt Publishing. Kindle Edition.
33. TrainingByPackt. (n.d.). *Data Science Projects with Python*. GitHub. Retrieved 20/12/2023, from <https://github.com/TrainingByPackt/Data-Science-Projects-with-Python/tree/master/Data>
34. Kitano, H. (2002). Computational systems biology. *Nature*, 420(6912), 206-210
35. Docker. (n.d.). *Get Started, Part 1: Orientation and Setup*. Docker Documentation. Retrieved 20/4/2020, from <https://docs.docker.com/get-started/overview/>
36. Kubernetes. (n.d.). *Services*. Kubernetes Documentation. Retrieved 29/4/2023, from <https://kubernetes.io/docs/concepts/services-networking/>
37. Kelly, W.P.; Cocco, F.; Mikkilineni, R. General Theory of Information, Digital Genome, Large Language Models, and Medical Knowledge-Driven Digital Assistant. *Comput. Sci. Math. Forum* 2023, 8, 70
38. P. Cockshott, L. M. MacKenzie and G. Michaelson, "Computation and Its Limits," Oxford University Press, Oxford, 2012, p. 215

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.