

# Designing Self-Learning Agentic Systems for Dynamic Retail Supply Networks

Raviteja Meda <sup>1,\*</sup> 

<sup>1</sup> Lead Incentive Compensation Developer, USA

\*Correspondence: Raviteja Meda (rvtjmeda@gmail.com)

**Abstract:** The evolution of supply chains (SC) from a linear to a network structure created an opportunity for new processes, product/service offerings, and provider-business. Rising customer service expectations have led to the need for innovative SC designs to develop and sustain competitive performance globally. Firms are forced to respond and adapt accordingly, thereby leading to design, network, operational, and performance dynamics. Traditionally, SCs are treated as static structures, focusing solely on design and/or operational optimization. Such perspectives are not viable options for SC domains, as they address only a portion of the dynamic problem space, use a deterministic assumption of dominant design variables, capitalize on past data to predict future decisions, and offer pre-classified forecasting options complemented with a limited comprehension of systemic SC elasticity. Novel self-learning agentic systems are proposed that blend the sciencematics of SC decisions and dynamics. The designs guide firms seeking to build adaptive SCs using operational decision processes. The designs address the agentic nature of SC, embedding computational interaction models of firm SC networks. The designs contrast the stochastic action-taking and thereby the performance outcomes, discovering opportunities for adaptive operational designs of SC tasks. Fine-tuning and meta-learning are new design capabilities that adapt to evolving dynamic environments. Frameworks for behavioral customization and systematic exploration of the design space are provided as user guides. Exemplar designs are also provided to serve as a translation template for users to express operational models of their own contexts. To account for the dynamics of supply chains (SC), agent-based models are increasingly adopted. Such models exhibit SC structure and/or formulation dynamics. Though existing efforts commence adjacent-only structural changes, dynamism with respect to tasks is crucial for SC design and operational strategy development. Proposed is a process modeling library and workflow for discovering intricate designs of adaptive agentic systems. The library revises Dataflow and Structure, concealing sequencing and context designs of processes. Prompted specifications describe and enact designs. Applications in SC formulation discovery are provided.

## How to cite this paper:

Meda, R. (2020). Designing Self-Learning Agentic Systems for Dynamic Retail Supply Networks. *Online Journal of Materials Science*, 1(1), 1–20.  
DOI: [10.31586/materials.2020.1336](https://doi.org/10.31586/materials.2020.1336)

**Keywords:** Self-Learning Systems, Agentic Intelligence, Dynamic Supply Networks, Retail Supply Chain Optimization, Autonomous Agents, Multi-Agent Systems, Machine Learning in Supply Chains, Adaptive Retail Systems, Distributed Decision-Making, Real-Time Supply Chain Analytics, Reinforcement Learning, Supply Network Resilience, Intelligent Retail Systems, Data-Driven Logistics, Cognitive Supply Chain Agents

**Received:** October 10, 2020

**Revised:** November 16, 2020

**Accepted:** December 20, 2020

**Published:** December 27, 2020



**Copyright:** © 2020 by the author. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Retail supply networks are increasingly becoming dynamic due to changing market conditions, product structures and customers' demands. As business processes become dynamic, a new generation of intelligent agentic systems needs to be developed to effectively cope with the complexity of constantly varying retail supply chains. These systems should be self-designing adaptive to cope with continuously varying supply networks as well as agent-supplier networks, and have unsupervised learning

mechanisms based on adaptive agent associations navigating through a multi-dimensional parameter space of the supply chains. This presents a need for research on designing and benchmarking intelligent systems for evolving environmental supply chains for a group of agents assigned to track a group of suppliers.

There have been various methods and tools proposed to address the problems of collaborative decision making for agents in multi-agent systems. However, the environments of most of these methods are static or continuous. Larger simulation environments, increasingly dynamic agent-supplier networks, and the associated agent-agent interactions require agent-supplier networks with the ability to learn and adapt to continuous changes in the environments continuously. Fast changing and stochastic external environments increasingly challenge the planning decisions of agents in multi-agent systems. New mechanisms are required to quantitatively evaluate and compare the proposed planning systems in such dynamic environments. The limitations of the existing tested simulated environments need to be overcome in a greater scope of diverse and deeper dimensions of agent behaviour, their corresponding environmental states, agent-supplier networks, and design and learning mechanisms.

The overall research objectives are to develop a new generation of benchmarking simulation environments and multi-agent simulations to evaluate various systems against a greater scope of designs, implementations, methods, and strategies; and inspire innovation and new designs for intelligent self-learning agent-supplier systems. To achieve these objectives, the research proposes a new concept of fully adaptive environments and associated approaches of mechanism generations for improved agent-supplier networks.



**Figure 1.** Agentic AI use-cases in Retail, Manufacturing, Supply-Chain, and business benefits

### 1.1. Background and Significance

The global economy and global supply chains are changing as a result of rapid advances in technology. People are feeling the impact of this transformation in both their professional and personal lives, creating both risks and opportunities. Moreover, recent events such as the outbreak of COVID-19, ongoing trade disputes between world powers, and stresses on production and shipping have accelerated the change. Health pandemics, natural disasters, trade disputes, and other unforeseeable events have made it difficult to predict safety and reliability in increasingly complex global supply chains. Supply network disruption poses significant risk to firms, and technology is increasingly being employed to design and reshape networks.

While artificial intelligence is giving firms new capabilities and agility to respond rapidly to change, it is vitally important to ensure that AI-enabled decisions are always trustworthy and ethical. New approaches are needed to use Artificial Intelligence (AI) technologies to design self-learning agentic systems for new product development, manufacture, distribution, and sale by very large networks of component suppliers, assemblers, distributors, retailers, etc. in the consumer retail goods sector. Rather than specifying explicit functions for such agentic systems, a complete set of rules governing

behaviour is specified, making them reactive and adaptable to new and changing environments. They include rapid reconfiguration design principles, taking inspiration from natural agentic systems such as flocks of birds, ant colonies, and the nervous system.

However, creating agentic systems capable of credible and intelligent enactment of network designs that change in rapid response to changing environmental conditions and events is a significant challenge. Existing AI techniques alone are inadequate. Agent-based modeling (ABM) is widely used to study supply networks and has provided significant insights into their performance. It has also been applied to explore architectures for agentic systems. Frameworks exist in which reactive agentic systems can be designed using rules governing their behaviour, which are compatible with ABM. However, two significant challenges remain. Having been designed using ABM modeling software, thousands of agentic agents cannot be simply coded into any existing system context. It's important to ask what is needed to enable forming a large number of agents over complex networks and to enable them to function effectively.

*Equation 1: Demand Forecasting with Reinforcement-Adjusted Predictive Model*

$$\hat{D}_t = f(X_t) + \gamma Q(s_t, a_t)$$

Where:

- $\hat{D}$ : Predicted demand at time  $t$
- $f(X_t)$ : Traditional forecasting function (e.g., regression or neural network)
- $Q(s_t, a_t)$ : Q-value from the reinforcement learning agent at state  $s_t$ , action  $a_t$
- $\gamma$ : Weighting factor for agent influence

## 2. Literature Review

However, whether more intelligent supply system structures will emerge with the artificial-decision revolution is unclear, given the game-theory framework design. Determining how self-learning agentic systems shape the problem-space and HRM choices as first tasks of agents in a usable ecosystem for well-balanced efficacies is urgent. Hence, fundamental characteristics of emergent meta-agents and meta-networking designs of co-evolutionary organizational structures responsive to environment changes are explored.

## 3. Conceptual Framework

On the other hand, modeling becomes paramount when it comes to the domain of interest. Current models abound but remain unsuitable for studying learning dynamics of self-learning agentic systems due to their normative focus. Consequently, the quest arises for how to model a self-learning agentic system that is context-specific, intricate yet sufficiently abstract, and incorporates the recess of interoperability. Agent-based modeling offers a possible solution as agents can express themselves independently leveraging the incorporated domain knowledge and capabilities, including learning, offering a meet halfway solution for the exploration in question. The doubts about reading agent-based artifice are similar to doubts about reading ideas. It becomes apparent that agent-based modeling implicitly presupposes a theory of situated agency. It becomes less apparent how such a theory is to be fleshed out or even discovered. A logistical viewpoint subsequently emerges. The viewpoint reminisces with an important twist: impossible problems need not be present failures but only future expectations. To substantiate such a viewpoint, designing self-learning agentic systems requires platform engineering, i.e., a model of embodying abstractions that are sufficiently refined that stylistic considerations

are all-important, i.e., algorithms and deductive structures become unimportant. Whereas agencies are abstract, platforms are concrete and thus more accessible. Directing attention to platform engineering entails two ingredients. Firstly, agency can be regarded as manageably mandated but still flamboyant abstractions derived from pre-existing ones such as social, economic, physiological, material, etc. Secondly, designing matchmakers that interoperate with agents requires expert knowledge, thereby paving the way for cognition, sense, and regulatory measures that “carries” agency right through totally invisible abstractions. Despite the hurdles concerning reading agent-based artifice, self-learning agency remains intelligible, and even learnable notions of agency remain discernable.

### ***3.1. Defining Agentic Systems***

The term agent and agency are used in many disciplines. The most general use refers to systems that perceive and act on the world. The description of its perceptive and effectors capabilities can be enriched by describing their powers to learn (for example predict) and change the world (for example by making others learn). This is the idea behind the terms learning agent and agentic system. The face of the agent can rely on many metaphors ranging from programming instructions to verbal descriptions and from a neural or genetic artificial net to a mythical deity. Agentic systems are capable of learning how to learn (for example from previous experience) or how to build devices that learn. They can learn by their actions (for example exploring the situation) or from a communication with other agents (for example elicit information). The simplest description of an agent is based on the environment it perceives via sensor inputs and that it acts on through effectors. The coupling between environment and agent may exhibit different behaviors, they can be deterministic or nondeterministic, or continuous or discrete.

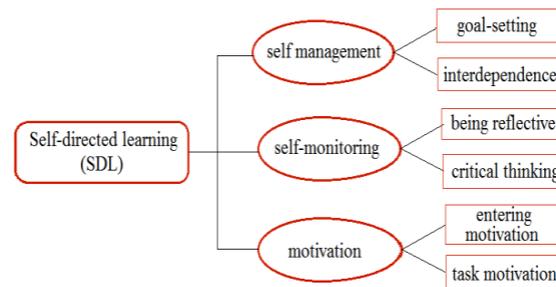
Fifty years ago, the increase in the number and complexity of manufacturing processes fostered a concurrent multi-discipline research effort in Canada, Europe and several other developed countries to find, model and formalize the rules of teamwork required for various processes. The renewed interest in agent oriented methodologies and systems has revealed new facets for these same capabilities. New software tools and a means to model relationships have been developed for concurrent research in many disciplines including design, architecture, behaviour, operations and knowledge management. An interesting question is whether and how agentic modeling and method development can benefit from this discipline wide effort. New sophisticated agent programming languages have given rise to a renewed interest in the relationship between beliefs, desires and goals. The choice of control architecture is crucial, since many unanticipated emergent phenomena and capabilities depend on it. New approximations, like multiagent based metrics, nascent approaches, such as analytic-planning and hybrid systems, and agent architects that jointly compute parameters and behaviour have also been formulated. It seems that better means to characterize learning are much needed, since many current proposals, like genetic learning, have not been well formalized or given a rigorous foundation.

### ***3.2. Characteristics of Self-Learning Systems***

Self-learning systems have been widely researched and applied in various domains where long life and mere usability are desired. Self-learning systems are often characterized by their adaptive, context-aware, and multi-agent system properties where multiple mechanisms, including social, collaborative, distributed, and decentralized approaches, are utilized to learn to improve performance. While significant research has been devoted to discovering the compositions of self-learning systems, little attention has been given to self-learning systems in dynamic supply networks.

Considering the extensive research on self-learning systems and a detailed understanding of dynamic markets and retail supply networks, the following questions arise: What are the characteristics of self-learning agentic systems for dynamic retail supply networks? What are the new trends in self-learning systems in agent-based modeling? In the context of agent-based modeling for retail supply networks, where agent interaction is often based on rules or mappings, a black-box model and a properly defined self-learning mechanism can be utilized to eliminate a long period of significant financial loss, such as a few years of adapting to the new situation of the supply chain. The major contributions of this research effort include discovering the characteristics of standalone self-learning models, contextualizing existing agent-based modeling research gaps, and providing a road map of research that practices a self-learning mechanism to innovate agent-based modeling approaches to opportunities & threats of supply networks.

Agent functions refer to the objectives of an agent that can constrain an agent's behavior and mapping from states to actions. A more complex function may capture the essence of the target behavior but be harder to learn from sampled information. Learning can be roughly classified into supervised, semi-supervised, and unsupervised learning, depending on the nature of the target function. Reinforcement learning does not explicitly involve any kind of input-output mapping; target information is solely in terms of reward function.



**Figure 2.** Dimensions of Self-directed learning (SDL)

#### 4. Methodology

Self-learning systems in the context of agent-based models have large potential for use in the supply chain optimization domain. They allow for alternative configuration strategies and supply chain resilience adaptation options to be explored. They can also deliver real-time simulation options with alternative supply chain design and control strategies. In this context, a self-learning system adapts over time. Functionally, phase one entails design of the simulation environment, data feed and interface in a simulation tool, and exploration task input to a self-learning system. Phase two consists of implementation of appropriate self-learning algorithms to mimic adaptive or self-learning agents. The suitability of the approach is illustrated by means of a case study on sustainable design of supply chain configuration and control options. It shows how adaptivity is simulated and how self-learning configurations are identified. New configurations are generated that outperform existing configurations. The potential for agent-based modelling to improve supply chain design, adaptation, configuration and resilience of supply chain controls and operation is illustrated. In phase one, the supply chain simulation modelling environment, supply chain scenario, simulation design scheme, input interfaces, and appropriate agent architecture are developed. In phase two, exploration tasks applied are described, including how agent actions and performance are expressed in the respective fitness function definitions both in a single- and multi-agent context. The functionality is illustrated with three different use cases that explore exploration tasks for simple designs with pre-defined exploration performance measures, optimal solutions as states of the

world with predefined actions, and adaptive agents in a pre-defined continuous change scenario with multi-criteria fitness definitions. The assessments provided insights into the value of adaptive and self-learning agents in resilience adaptation and improved design and configuration search.

#### **4.1. Research Design**

This work suggests a research design that will allow the development of candidate self-learning agentic systems capable of being evaluated within the synthetic environment of dynamic retail supply networks. The architecture and training by imitation design of individual agent learners are described, as are the approaches being taken to evaluate the resultant systems. The central aim of the research is to design new agent-based systems for controlling operations in the context of dynamic supply networks in the retail sector. The results from the two investigations so far conducted, which use simple rule-based systems to adapt to conditions in synthetic historical data provided by agents designed to emulate human decision making, demonstrate the feasibility of this aim.

As a result of examining the literature on agent-based systems for supply chain management, it is clear that there is significant room for improvement of the state of the art. Past efforts have not delivered the robust, responsive, agentic systems that are required for these systems to perform as intended with the current provision of data. A previous investigation comprising a model selection challenge on a supply chain scenario with a dynamic demand was developed for exploration of deep reinforcement learning. This scenario is IO-driven, requiring high throughput with minimal holding of products, but the GA effort was largely ineffective in optimizing for this objective.

As a result, this research is undertaking the investigation of supply chain agent design from a contrasting basis, with the aim of exploring a different set of available architectures and learning approaches, and of casting these in ways that seek to address the obvious shortcomings of earlier efforts. The focus here is on the design of individual agents, but this is occurring alongside an investigation of an appropriate synthetic environment for the evaluation of new agent systems, and each aspect of the research design will be detailed.

#### **4.2. Data Collection Techniques**

This section presents an overview of the data collection and preparation techniques for the case study. The focus of research is to investigate using a data-driven approach to enhance inventory efficiency. Hence, RFID data has been chosen for exploration as such data is collected frequently and comprehensively in supply networks.

The supply network in the case study is an IKEA-inspired retail supply network called RnJD. The design of RnJD shows similarities to the IKEA retail supply network. The first to third nodes represent suppliers and distribution centres (DCs), while the fourth node is the retailer. Ground delay and clearance delay are typical delay patterns. vehicular delay between facilities is the scenario designer for the case study. RnJD consists of four nodes: suppliers (AgenA), DCs (AgenB), and a retailer (AgenC). There are three wholesaler suppliers and five DCs in the RnJD network. Each node has two significant products with a safety stock level of 0.2 based on the total demand for each product in previous weeks. The connection among nodes is either all-to-all or random. In both designs, the vehicles are constantly at one of the facilities, the duration determined by the standard operation procedure, while the travel time is generated with a Normal distribution.

Profiting from the data-driven approach, dynamic holding and transportation costs can be considered in addition to the original static ones. With the increase of nodes and the simultaneous launch of the data-collection software of the nodes, the data volume can increase by several hundred GB within an hour.

### **4.3. Analysis Methods**

The key focus of the present research is on the examination of supply networks in the retail industry. The proposed conceptual study aims to contribute to the literature on supply networks and business ecology through a framework of self-learning agentic design. This theoretical framework incorporates ontological, epistemological, and design dimensions of agency and agentic technologies. Through focus articles, the framework is championed in the context of the retail industry and in a narrower context of retail supply networks. The focus on and contextualisation of the retail supply net-willing increases the theoretical contribution in specific areas, and the innovation and managerial relevance of the research. The present study adopts a constructivist approach through translation of a multi-paradigmatic theoretical framework into the retail supply network context. Using causation, the self-learning agentic system of designing and reshaping retail supply networks is theorised to allow for a step change technology. It is believed that the newly theorised technologies improve the efficiency of flow networks such as retail supply networks by a layer of design above real-time super-fast execution. The dynamicity of the supply networks in the retail industry facilitates reflections on design tuning and redesign of future agentic networks and culled agentic technologies and agentic time. The framework allows for interactive pluralism of design sciences extending existing off-the-shelf modelling technologies and aids explorative theory building on supply network agency, agentic technologies, and shifting topologies. Proposed methodological options are of cross-cutting nature amenable to various modelling languages and analyzed using scenarios or semi-analytically.

Peace is described as a dynamic and emergent property that can be constructed and co created by agentic designs, activities and processes. A unifying distinction between several streams of research addressing distributed design semiotics, eco-semiotics and sense-making exposes the multi-analysis of agentic networks. It is argued that sense-making and designing are interrelated because social constructs and artefacts continually coevolve. Implications of the co-evolution between social constructs and artefacts in purposive and non-purposive systems are discussed in the context of studying art, model evaluation, and artificial life.

### **5. Dynamic Retail Supply Networks**

Dynamic Retail Supply Networks (DRSN) provide particular opportunities and challenges to traditional Supply Network Design (SND) actors due to their decentralised complex nature. Retail Supply Network (RSN) nodes incur demands that are uncertain and time-varying during their evolution towards equilibrium states. In addition, transportation and handling service providers are multi-organisational, requiring dynamic routing and scheduling solutions across this emergent ecosystem. The existing agent-based DRSN modellers provide frameworks and concepts to simulate DRSN dynamics, but little effort has been done to test them empirically and to design the agentic DRSN agents and their operating, coordinating, and learning decision systems.

Challenges facing existing retailers in urban areas and the opportunities afforded by new agentic technologies are presented first. These present a compelling case for creating new agent-based modelling solutions, but further work is required to ensure that proposed systems have sufficient computational intelligence capabilities and functionality. Existing agent-based modelling frameworks and literature on supply networks, agentic architecture design, negotiations, semantic modelling, and machine self-learning, are surveyed as a potential basis from which to build new agentic systems. These systems will be needed to explore and test new agentic architectures and agents in more complex settings of urban retail supply chains than has been done thoroughly in existing agent-based modelling. Developments in this area will help ensure that existing and new DRSNs are designed, planned, optimised, and operated in ways that advance

the efficiency, equity, and sustainability of urban areas, while sustaining current and future economic functionality of retail.

Retail supply chains experience omnichannel disruptions due to various factors. Ride-hailing and delivery technologies afford opportunities for new types of SND. Agent-based methods provide empirical modelling options that can represent and study the evolution of DRSNs and how agents negotiate. Multi-level agentic architecture design is challenging in current approaches. What agent artefacts and intelligence best fit with the rest of the simulated and studied networks? What truck, drone, bike, or pedestrian can discuss how to make offerings more efficiently for home delivery, without a trigger mentored by a camp?

**Equation 2: Multi-Agent Learning Objective (with Inventory-Reward Feedback)**

$$\max_{\pi_i} \mathbb{E} \left[ \sum_{t=0}^T (R_{i,t} - c_{i,t}(I_{i,t})) \right]$$

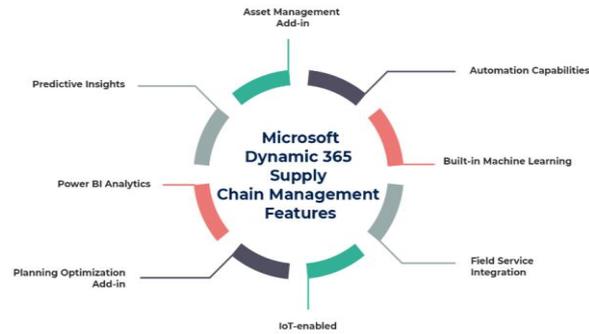
Where:

- $\pi_i$ : Policy of agent  $i$  (e.g., supplier, warehouse retailer)
- $R_{i,t}$ : Reward for agent  $i$  at time  $t$  (e.g., profit, customer satisfaction)
- $c_{i,t}(I_{i,t})$ : Inventory cost function for agent  $i$
- $(I_{i,t})$ : Inventory level of agent  $i$

### 5.1. Overview of Supply Chain Dynamics

Due to customers' ever-increasing demand for high responsiveness and low inventory management costs, competitive pressures force retailers to reduce their supply chain's integrated order fulfillment lead time while maintaining or improving operational performance. Given the increasing uncertainties and complexity in the market-driven work environment, agent-based self-learning approaches are generally favorable to traditional methods for order lead time reduction. Thus, recently, a multi-agent modular modeling approach is designed to investigate how a dense retail supply network's lead time can be reduced with self-learning agentic systems and design their agentic systems' architectures to achieve the goal. The development describes how the environment unsupervised generates supply chain scenarios and how it exerts pressure on supply chain self-learning agentic systems during scenario generation. It also discusses various agent architectures design, competitive agents, cooperative agents, competitors-cooperating agents, and agent bridges.

In an effort to enable stakeholders to better understand supply chain dynamics, this research investigates agent-based supply chain models to help visualize some theorems about the impacts of competition and cooperation on supply chain dynamics. It is a core function of many companies to procure raw materials and discard waste or semi-finished products. In the traditional supply chain view, both suppliers and consumers are external to the core of production. Because of the reverse flows from the consumers back to the suppliers, the whole supply chain including suppliers is wider than the production system. Agents in the suppliers are modelled as simple first-order low pass filters with the same parameters for each. Rather than performing optimization on agents' parameters, a methodology is developed to stabilize the supply chain network.



**Figure 3.** Dynamics 365 Supply Chain Management Features

The model is implemented in AnyLogic. Several graphical facilities are employed to visualize the transport flows, the supply chain nodes, and the consumers supply chain performance indicators. Depending on the goals, the individual consumption profiles of consumers, manufacturing rates and pump rates at the suppliers can be modified to see how the supply chain performance is affected. The two competitive suppliers are implemented with secrecy constraints, thus their algorithms are not known to one another. However, as agents simultaneously optimize their performance criteria, in a way they are both trying to achieve stability and a significant amount of information can be inferred about the other agent behavior that can be used to gain a competitive advantage [1].

### 5.2. Challenges in Retail Supply Networks

Retail supply chains have wide-ranging characteristics. The scale and complexity differ from one value chain to another within and across industries. Retail supply chains also differ between regions of the world from the characteristics and complexity of the supply network itself to the idiosyncrasies of business practices. The bulk of products sold through retail outlets are acquired through a retailer's supply network of upstream nodes. It is these types of retail supply chains along with their corresponding models that are the main focus.

Dynamic activity is often generated by consumers buying stock from retailers. Daily demand in the form of product sales is forecasted to give the retailer a target order quantity. Once seen, this is sent down to the supply network in a signal order. Other nodes forward demand by periodically placing an order with a node further upstream. The node passes this order upstream but places it alongside a forecasted order quantity. Each of the tasks must take care when generating signals to ensure that the due times always increase along the supply network paths. This is a long-established discrete event, continuous time, distributed process. The task structure and enforceability of the synchronization properties are permanent.

The reward structure dwells on prescriptive inventory swings measured as the difference between actual and target stock levels. These graphs provide an understanding of each node's individual control actions and inter-relationships across the supply network. Noise comes from, and is added to, the demand signal and the ordering mechanisms of supply nodes. The demand can change shape to simulate campaign activity, a new product introduction, or serious stock-out or competitor activity at a retail node [2].

## 6. Self-Learning Algorithms

"Self-learning agentic systems" refer to systems that take actions by themselves to improve their action selection logic to achieve objectives better. They rely on self-learning algorithms, which select actions based on existing knowledge and update knowledge based on the received information. Such systems that can continuously learn and adapt

behaviors to achieve goals have been an interest of researchers in artificial intelligence and robotics for decades. Well-established self-learning agentic agents include self-learning bots for video games and website crawlers. As fundamental subsections of the agentic systems, self-learning algorithms that can take the objective, action space, and observed state space as inputs and work with simulators to learn the knowledge autonomously are researched by the machine learning community, also with significant attention. Beyond the board or physical worlds, significant intention and efforts have been made in applying self-learning algorithms in various simulated environments with specific objectives.

The first challenging environment is multi-agent supply chain environments. This research considers a decentralized, multi-agent, cooperative supply chain game where the participants aim to minimize their total costs while managing the inventories and being imperfectly informed about demand. The core problem is modeled as a Markov decision process, and a self-learning algorithm based on a deep reinforcement learning technique is proposed. It is demonstrated that the proposed self-learning algorithm obtains near-optimal solutions when optimized alongside agents who follow a base-stock policy and performs much more satisfactorily and stable than a base-stock policy when the other agents utilize a more realistic model of ordering behavior that includes seasonal effects, trends, etc. A noteworthy feature of the proposed agent is that it does not require knowledge of the demand probability distribution and applies only historical data with recently observed values.

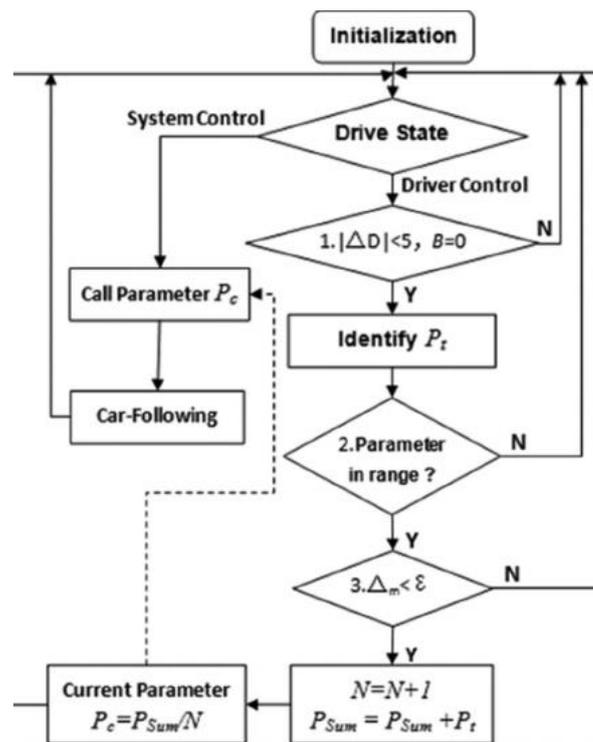


Figure 4. Flowchart of the self-learning algorithm.

Another more peculiar environment is large inventory networks. The objective here is to learn an inventory management policy of placing orders with an arbitrary non-negative quantity for each of the two types of product through four monotonic actions as well as a mechanical learning process. Rule-based approaches have succeeded in operation research-based classical inventory management. A multi-agent policy gradient approach has been extensively studied to derive a joint order-up-to policy using a centralized agent to improve stability in numerical results. On the other hand, the learning-based approach for inventory management has been studied since the year 2000.

In this type of approach, the demand data is treated as a time series, and the order quantity at each time step is determined using reinforcement learning. A self-learning agentic system is constructed to derive an inventory management policy for a large inventory network composed of an arbitrary number of retailers and products.

### **6.1. Machine Learning Techniques**

Supply chain (SC) disruption has become a common phenomenon since the onset of the COVID-19 pandemic in early 2020. In such crises, stakeholders look for credible information sources that provide data-driven insights to alleviate the panic and uncertainty. This advent of exponentially growing data from various heterogeneous sources has led to the resurgence of AI and ML across the domains. Business enterprises are now engaged in data democratization by increasingly adopting tools or platforms that make data accessible to multiple levels of users, whether trained professionals or laypeople. The ability of such self-learning agentic systems to not only learn from the prior data but also agentively evolve as novel and data-rich scenarios unfold is more significant than ever before. This new research approach advocates the unfolding of data streams across distinct levels of user creativity, from creativity with data straightforwardly harnessed by data mining/machine learning (ML) processes to creativity with cognitive data that must be abstracted by data science processes. A proprietary knowledge-sharing platform for supply chains in the form of a plethora of data pipelines can be created to train such self-learning agentic systems.

The following sections describe different AI-based designs including, Reinforcement Learning Agent for Real-Time Slot Allocation to CCA, Surrogate Agent for Improved Class Scheduling with MC, ChatGPT Agent for Proactive RFI Handling, and Topic-Creating Agent with Yellow Brick Visuals & Word Clouds. Moreover, the lessons learned and future research proposals are extensively investigated and proposed, respectively.

### **6.2. Reinforcement Learning Applications**

Reinforcement learning is increasingly being explored as an alternative for solving complex dynamic inventory problems in supply chains. Conventional approaches can become prohibitively costly and computationally expensive for complex problems, and rigidly designed control policies may fail during unforeseen scenarios. As a result, data-driven methodologies, such as machine learning and reinforcement learning, have been increasingly applied in designing and controlling supply chains. Exploring approaches, challenges, and opportunities for applying reinforcement learning in dynamic supply chain management is key. Reinforcement learning has been successfully applied in gaming, robotics, recommendation systems, and autonomous vehicle driving. There is currently an increased interest and research in applying reinforcement learning in other domains such as healthcare, finance, biology, material science, supply chains, climate science, etc. Reinforcement learning studies how agents interact with their environments to maximize some notion of cumulative reward or return. In reinforcement learning, there is only a scalar value as feedback to examine whether the learning is in the right direction or not. It is crucial to define the reward structure according to the problem statement.

Given a situation/state, reinforcement learning can be defined as an agent that learns in real time from its environment which actions to take. Based on its current policy, the agent takes an action and then receives a scalar reward. The agent maintains a value function to predict the expected long-term reward given a state. The most widely used reinforcement learning methods can be classified into model-free and model-based. In model-free reinforcement learning, the environment is assumed to be a black box. There are two components in this framework: a blind policy that selects an action based on the observation and a critic that learns how good the current policy is. The policy is updated by a policy gradient algorithm based on the critic's feedback. The critic can also be improved in a value iteration fashion. The main challenges of multi-agent reinforcement

learning include the curse of dimensionality, partial observability, and communication costs.

## 7. Agent Architecture

Wondering about the composition of the architecture, this study devised different agents with specific roles in the supply networks. Based on the target area of the agent, their job was predefined such that they either control the node, the links in between the nodes or perform the self-learning & decision-making tasks. Firstly, the node control agents are put in charge of checking the node parameters. The node states are needed by the supply nodes before deciding on consulting the upper-point distribution agent(s), because if a source has enough supply, logically it cannot be sent to consultation. These agents observe the change of allocated parameters in regular intervals, keeping the values in a local parameter memory. If the monitored values exceed a threshold, the upper-point distribution agents are consulted to reshuffle the agent resources, and service agents check if the threshold value is not exceeded, pf sum is reset to return to the default state. These agents pass over the thresholds, excessive POSsum (low supply) or less than expected POSsum (high supply) to the other-side distribution upper-point agent in a query decision. The agent which receives the consultation decides based on the previously recorded and fed data. The distribution agent then affects the parameter memory of the holder agent and influences its memory during communication.

Secondly, control agents are in charge of the links between the nodes, processing the states of the edges. As with the node agents, traffic estate agents observe a time period of it and store the travelled distance in memory. If the monitored limit value is met, or the travelled distance was defined beforehand, flow control agents are consulted. Depending on the edge states, these agents can delay or speed-up the flow of the travelling packages by adjusting the sensor's sensitivity, as well as making them unblock or unblock. After consulting the control network states are shared. Thirdly, two self-learning control strategies are implemented for the agent trained on the package distributions. One agent is a Q-learning-based, active exploration agent, an exploration greedy approach. In regular intervals, the SCM agents consult the current states and knowledge base of the tracking agents (learning node agents), and share the agent profiles equally, log the updates to the knowledge base, and compute info penalties. The goal of the self-learning agent is to predict the future package distributions based on past history. The incoming & outgoing transport queries in the last 10-second long time step signal both SCM and learning nodes are stored, these aggregates are used to train the action-making self-learning agents.

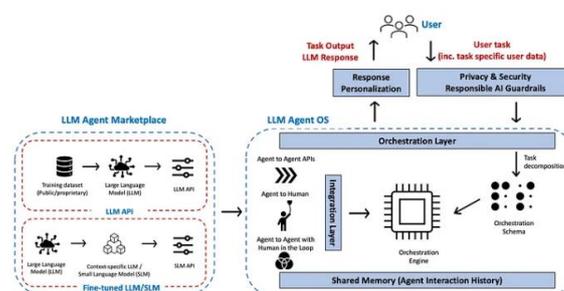


Figure 5. AI Agents Reference Architecture

### 7.1. Design Principles

A set of design principles for a self-learning agentic architecture is developed, relevant to the problem domain of retail supply chain management. This is presented with the elements of the architecture described in the previous sections, to demonstrate how it

enables a dynamic modeling approach to supply chain network design. The design principles and agentic architectures suggested here are first presented and outlined, followed by the knowledge domains and representation techniques used, ending on a discussion of the experiments run on the resulting agentic systems.

The proposed design principles for self-learning agentic architectures are as follows:

1. Formulate behavior in terms of high-level tasks, plans, and actions (TTPAs).
2. Model networked semi-autonomous agents, each with their own knowledge base and learning agenda.
3. Categorize and classify a diverse set of knowledge types forming a knowledge map.
4. Model behavior in a hybrid symbolic-connectionist manner.
5. Use temporal logic as a basis for defining plan constructs.
6. Historically label plans and develop associative memory to enable their transfer across agents.
7. Support the interpretation of TTPAs in a new environment to propagate knowledge.
8. Adapt plans through methods such as instantiation, modification, or creation of new course-of-action approaches.
9. Transfer knowledge either by communicating TTPAs in human-readable format or by stimulating the underlying model to allow for inductive transfer of knowledge.

These design principles are derived from the author's previous research experiences with agent-based modeling. As the case study presented in the next sections is developed, each design principle is elaborated on and explained with reformulation advice provided. To explore a new way of modeling retail supply chain behavior that is adaptive, self-learning, changing in structure, and realizable if given sufficient resources, a here-and-now experiment is described. An agent-based modeling environment captures how networked agents trade, delivering inquiry into retail supply chain networks. The investigation addresses how supply chain behavior emerges in agent-based modeling, how the self-learning capabilities of agents can give them in-depth understanding, and how the knowledge gained from past experiences is reused for future behavior.

## ***7.2. Communication Protocols***

As introduced in the previous sections, this framework describes a self-learning agentic system in the domain of retail supply networks, which undergoes frequent changes in its environment. In a retail scenario, supply chains are considered dynamic, necessitating protocol extensions and methodology-related developments in addition to the base architecture of the agentic system. Preemptive communication protocols allow for self-learning in the structured environments of players where all previous states are accessible [3].

A bounded challenge is subjective in collaborative scenarios of non-contributing agents, such as unintended player drop-outs. It has been argued that merging both the beliefs and the meanings of terminology in a multinational organization is an NP-complete problem. This gap may even be infinite in nested belief operators. Investigating how mechanisms can be devised to make iterative guessing manageable in chat-room-like scenarios will be left to future work.

It appears that unless prolonged unresponsiveness is detected, agents maintain a same-level epistemic model even when the graph of active agents is no longer connected. Some upper bounds on the agent capabilities and responsiveness are required, along with attention distributions, to narrow their perceptions of the world closer to those of humans in traditional organization form.

***Equation 3: Adaptive Replenishment Function with Agent-Based Noise***

$$Q_{i,t+1} = Q_{i,t} + \alpha (\hat{D}_{i,t} - Q_{i,t}) + \epsilon_{a,i,t}$$

Where:

- $Q_{i,t}$ : Current replenishment quantity
- $\alpha$ : Learning rate
- $\hat{D}_{i,t}$ : Forecasted demand
- $\epsilon_{a,i,t}$ : Exploration noise from agent  $i$ 's policy

## 8. Implementation Strategies

A multi-agent system consists of a set of intelligent entities (agents) that can interact with their peers and work together to solve some complex or difficult problem. Each agent relies on its own knowledge and capabilities to carry out its share of the processing but also communicates with others to coordinate, cooperate, or negotiate. Commonly, multi-agent systems are focused on systems formed from a number of agents having their own beliefs, minds, goals, and intentions and how those agents might interpret, negotiate and resolve conflicts among each other. Such systems may be statically defined, or agents may be dynamically created or destroyed. Moreover, new agents may enter or leave the system. In the taxonomy of distributed systems, agents have features of concurrent, cooperative, and loosely coupled systems. Multi-agent systems could have spatial or organizational structure, and the multi-agent system technical or application domain may impose further requirements.

Multi-Agent Systems (MAS), comprising multiple intelligent agents cooperating toward a common goal, have been widely used for the design of distributed systems, allowing systems to be modular, easier to build and reuse. MAS is characterized by the properties of non-centralized control, limited observation of environment, cooperative problem solving, balancing between distributed and centralized approach. MAS are natural for coordination problems requiring concurrent activities, parallel processing, distribution of resources, and decentralized control. A subscriber to a data event can publish it for posting. The system maintains a data structure for the content-based querying to become multilingual, as the system needs to be capable of dealing with queries written in different languages.

Coordination in MAS is difficult due to agents being autonomous and possibly distrustful, and not sharing a common world view. Agents may have conflicting views of the environment, conflicting goals and goals that conflict with the actions of other agents. Distributed event detection and planning problems arise in MAS domains involving communication between agents. Approaches based on the communication of information related by a middle agent who does not remember it were presented based on graph data structures. MAS implementations were developed, as well as naively simplified plans. Formulating a planning problem mathematically is a complex task. Most approaches developed have fully automated planning systems.

### 8.1. Prototyping Agentic Systems

This section deals with the ongoing studies on the coordination of agents in dynamic retail networks. Worldwide stock availability or ability in dynamic supply networks is a common and skewed consensus. Retail stock issues are often due to incorrect (too late) resupply date detection or supply reason discovery.

Design principles for the coordination of  $K$  contained learning agents in a dynamic flow network of tithals is presented. Instances of the architecture that follows the design principles are described. The first prototype of a learning agent equipped with an operator is presented.

Flow networks connecting batches lead to stock levels of stockpiles. At these networks, stockable flow processes, producing, distributing, or consuming/using the flow are involved with learning agents. These agents have control and monitor the meaning. They make possible a certain degree of autonomy in social behavior, local decisions on flow processes. Modern retail networks consist of several dynamic and distributed supply units and are environments of jointly used goods.

Here,  $K$  learning agents on retail networks detected by a top-down spring model and imitating agent are examined in detail. Three aspects of how a stocking agent functions as a learning agent on its own network are described. (a) An input data processing operator managing diversity, data characteristics, and adapted estimation. (b) A flow forecast and resupply time detection operator generating predictions at the pool of flow predictors by a training producer net. (c) A resupply initiation operator generating consistent resupply initiation knowing data with 15 minutes time-advance.

The first prototype agent equipped with an operator has a pilot run in which  $K$  agents react to stock observations on stable standard topologies. The results show that the learning agents are detectable, follow the dynamics of stockouts, recognize the resupply reasons, learn the abilities of their suppliers, detect observational uncertainty, and are able to perform smoother re-supply activity. These findings are confirmed here separately.

### **8.2. Integration with Existing Systems**

There are different applications and hardware systems for stock keeping unit (SKU) forecast, shop sales forecast, distribution center (DC) forecast, sales out smoothening, as well as any combined implementation of these applications and hardware systems. It is recommended to not put needless efforts in re-implementing the established applications and hardware systems, as they are well specified in terms of data and configuration requirements. Manufacturing execution systems (MES) systems manage the execution of real-time production processes. They include modules to schedule production orders, require data from sensors on the shop floor, and provide notifications on order releases, statuses, and confirmations. MES monitors machine utilization, schedules maintenance actions, requires data on machine sensors and job set-up, and may include modules to estimate production costs. Forwarding planning systems plan routes for truck tours driven by dedicated drivers, require data on the services to fulfill, and provide estimated times for arrival at services [4].

SAP systems are used to plan and schedule weekly deliveries from suppliers. They require demand forecasts from customers, unless the demand of raw materials is constant. As standard practices for maintaining the partnerships, the weekly deliveries scheduled by SAP are mostly not modified. Data and configuration requirements to transfer forecast delivery schedules from SAP to a supply chain provided are Son-R, Dead-Line, ETA, and Preferred-Delivery-Time. Existing data and specifications on the physical layout of a plant and its production processes will be extended with caps on forecast production submitting that consider the resource requirements to produce these forecasts and performance considerations on validity length and update frequency. Data and specifications on horizontal batch mechanisms and the sequence of triggering these mechanisms that transfer forecast order releases from the agentic systems to the application and hardware system provided are order ID, tip time, and forecast order release data for truck distribution. Just before the execution of the order trigger, a system monitoring tool continuously corrects the execution and sends out any notifications on missed forecasts. If needed, some built-in buffers within a warehouse may be modified to safeguard the correctness of execution.

## **9. Case Studies**

The shopping experience simulated in VERSA is derived from a case study in a large UK department store: "Store A". Store A is part of a high street departmental store chain

that operates in the retail industry for clothing and accessories. This chain considers its employees a valuable asset and reaffirms its commitment to human resource management (HRM) practices. The department store chain is committed to providing high quality service, efficiently meeting customer expectations and requirements. The historical records show that top and middle management executives have been involved in designing a separate aisle which can be allocated to those types of products which require more care in handling, ensure better inventory control and customer service.

PERKS & OFFERS is one of the departments within Store A, with emphasis on parallel checkouts and Kiosks. It primarily deals with the four main product categories. In order to simulate this feature, a hierarchical structure is defined in the specification and implementation of agents. Individual Kiosks are at the lowest level of this hierarchy, which takes direction from the second level - the Parallel checkouts. In turn the Parallel checkouts take direction from the Department Manager, which resides at the top level of this hierarchy.

VERA, Entity-Based Retail Simulation Architecture, is the core of the architecture of VERSA, which implements objects, methods and rules that dictate the interaction of the components. Objects in VERA are implemented using object-oriented technology based on C++. Virtual Retail Store, the main object, initiates the scenario and defines parameters at run-time. The Aggregate Manager store object within Virtual Retail Store manages the composition of many store objects viewing each object as a black box. To ensure simulation of a department average level of activity within the store, any department agent receives input reflecting a percentage share of business from the Store. Resulting from this information, the individual agents generate requests for the stock that will replenish their department's stock. The activity at the checkouts is then simulated as a total with all checkouts being compliant and processed synchronously. Sub-agents of Virtual Retail Store, Random Item Generator and Order Generator, create simulated sales transactions after a checkout object asks for item(s) input a random item, with 20% being invalid.

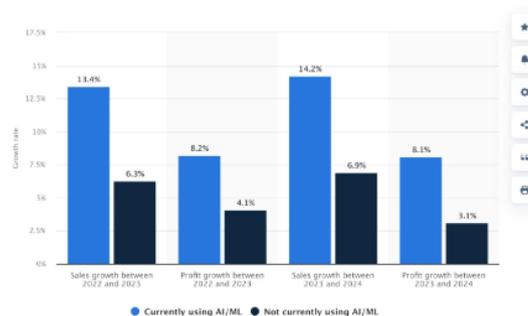


Figure 6. Machine Learning in Retail

### 9.1. Successful Implementations

Supply network design is traditionally not treated as dynamic especially in the context of supply planning and inventory control. Hence, holistic supply network design and planning procedures incorporating structural as well as operational aspects are possible. However, there is an urgent need to include both innovations in modeling as well as agent technology for developing agentic implementation approaches to develop, simulate and test solutions in a self-learning agentic network context. To address selection and implementation issues of sharing innovations in heterogeneous agentic networks, a detailed requirement engineering approach for agent development is required. Subsequently, the created agentic models can serve as high-level goal descriptions for the technical engineers providing the communication and negotiation protocols for

implementing the respective implementations in a specific software environment. An expansion with self-learning capabilities might also make sense immediately after testing the agentic solution. A stepwise process is necessary for each agentic network. First implementations are manual but can be executed and tested in parallel with the development of the respective implementations. This allows for checking if a given agentic design can be easily implemented as well as communicating a working prototype of desired agent designs for the detailed implementation of all software aspects. However, here the convincing insights are mostly theoretical in nature, and a process using accepted agent architectures combined with powerful simulation and optimization software is necessary.

Completely different tools can and are allowed to be used for complementary projects of other organizations so that agent design compatibility is impossible at the start. Starting with easy toy examples of compression application networks with standard models and dynamic programming can be used as a process starting point. Based on these results, more complex case studies can be developed including self-learning and self-organizing capabilities. This holistic approach leads to stepwise implementation and evolution paths of expanding capabilities, allowing for ongoing and coherent improvement of the design, implementation, and usability of local agents in a global framework.

## 9.2. Lessons Learned from Failures

One of the primary motivations for automating retail choices is for a system to alleviate the burden of an overwhelming number of options on consumers, directing customers toward serendipitous discoveries. This ambition has been unrealized, as the suggested items have been stale, and the systems fail to respond to changing trends. Several failed systems will be analyzed, along with exogenous causes of failure, such as changes in management, priority, or culture.

Between 2015 and 2017, a radio feature similar to Pandora was explored alongside existing playlist recommendation systems. For each song or artist, it constructed a queue of songs that adhered to many characteristics of the seed. The best-performing system generated rigid recommendations, curating safe, highly-replayed songs. The viral systems suggested low-rated, controversial songs in a bursty fashion, with sudden churn in both predictions and performance. In a priority conflict between the two main systems, the prior was discontinued. The chunked radio approach was successful in the context of a new product and was subsequently abandoned [5].

In the early days of a social media platform, its search engine suggested tweets. This was powered by a heuristic generating segment trees, comparing scores of candidate tweets according to their recentness, number of followers, and retweets. However, once the platform grew too large, its return time rose above two seconds, compromising user experience. The search team disappeared along with the service, as it was considered too costly to maintain. An ex-communicated engineer stated, "Flight 101 exhibited a fatal combination of bad luck and mismanagement." Twitter's disaster was compounded by a lack of a testing culture or clear ownership of timelines within the company. Search was also difficult to quantify for showing return on investment. Fearing a management shakeup, knowledge transfer to interns or contractors, and hence a worse outcome. The engineering team retreated to their offices, underground bunker style, and the feature never saw productization outside of return-at-zero bespoke systems.

Ultimately, priority misalignment will usually be fatal, and productization of new ideas may not be feasible if other systems are entrenched. In these cases, agentic systems thrive, suggesting discoveries across quarantine-partitioned paths, each independently modified and replaced as needed, as legacy software is often too brittle or bloated for modification.

## 10. Evaluation Metrics

The proposed framework uses a supply chain scenario that aims to meet customer demand and minimise costs while it is continuously varying. A batched demand is also added to represent a linear supply chain environment, which is why it is more interesting. The formulation of the environment thus results in a wider range of potential environments that the agent might see and thus a wider range of tasks for the continuous learning agents in that regard. The customer demand is placed at the retailer of this chain. As the customer demand is the only demand of the supply chain, the agent at the retailer is responsible for stock movement. This means that it has to decide whether or not to place an order. When there is not enough stock to fulfil demand, the retailer places an order. The retailer orders only from the warehouse, and the warehouse orders from the factory.

It is noted that batching effects are introduced here. When a batch order is placed by the retailer, it delivers them throughout the length of the duration of the batch. As such, the batched demand also effectively changes the amount of product currently required by the retailer. This requires the agent to react quickly because not only does it have to tell the warehouse how much product it needs, it will also have to take parts of the stock movement that it already has with the demand into account. Other models are simpler because providing a non-batch demand means that stock movement is essentially a non-event until much later.

Each stage in the environment results in the warehouse and factory having varying levels of complexity. Some actions are effectively non-events, and scheduling isn't incorporated directly in the task. This is different for the more complex expectations, because not all orders are delivered all at once, which means that stock movement is required. When an order is placed to the environment, orders to the warehouse, on the day of the event, must be clipped in order to conform to the capacity constraint of the environment. As such, the optimal cost for models cannot be assumed to be 0, as different models may need to take different paths in order to achieve an optimal cost. This also means that any model that is tested via a continuous learning manner must be robust, and this robustness is tested here.

### 10.1. Performance Indicators

Performance indicators are important to all other aspects of the self-learning agentic systems. Therefore, careful thought is required in identifying key performance indicators that reflect the purpose for which the agentic system was designed and for which it will be operated. It is difficult to change or adapt key performance indicators once they are set up so that they reflect the designing purpose for setting up the system. Key performance indicators govern the agent learning process in a network of autonomous agentic systems among which the workload is or need to be competitively apportioned and their agentic actions need to be appraised. The agentic mechanism embedded or programmed in the agents will run according to the key performance indicators. It is, hence, important that the key performance indicators are fair and impartial so that agents will not pursue their goal to the detriment of other agents.

A performance index requires a performance indicator or a function that outputs a value from input events or conditions and a performance metric to assess the output of the performance function. The performance index essentially quantifies some property of the networking agentic systems by means of its performance metrics at a certain level. When the system design is uncomplicated, the performance indicator, function, and metric may have a well-defined and unambiguous meaning, but in a large networking system, their meanings may not be unambiguous or may not be well-defined. Different or additional performance metrics are needed to appreciate its performance. The performance of a noisy entertainment video streaming system may be judged by its quality, latency and hence whether for the enjoyment of the audience, lots of things are considered. The quality may be assessed by means of the video-frame loss rate, video-

frame dropping rate, etc. When it is noisy, lots of redundant video frames are transmitted to mitigate the degradation of the video quality or the drop-outs. It is anticipated that while the number of transmitted frames is high, the video quality is good. A trade-off needs to be formed.

### **10.2. User Satisfaction Metrics**

Agent satisfaction and consumer satisfaction metrics are paramount in evaluating a service or product. Modeling agent satisfaction is pragmatic, as organizations generally acquire the system and thus have more information about the agent architecture. Organizations usually have a clear picture of the goals of agent actions; effector properties characterize the agent well; and observed performance metrics alone might lead to an unreliable view of the agent behavior.

Within organizations, there are roles concerned exclusively with supervising the behavior of the agents and observing possible applicant behavior that might either violate goals or have unwanted side-effects, or needing to focus on a diminishing of the arrival of bad/productive events from a specified source. In these roles, alarms might represent a desired service, thus, be a possible satisfaction topic. The focus on one of the subtopics follows from analyzing what is important for the gain of knowledge on the behavior of a specific agent-driven model, usually having information about the properties of that kind of agent. It almost always requires interacting with the agent by observing initially its parameters if it is in a situation that is especially interesting for the observer.

Consumer satisfaction metrics mainly fill the gap in the other process. The system behaves as desired when inputs and outputs are monitored, and the agent specifications are directly checked in the observable behavior of the network if it has been implemented correctly. The effects of a user call in terms of network changes regarding fidelity are checked. Alternatively, good answers' replies regarding time/fidelity are checked. Issues include the required knowledge of what scenarios were triggered and how real-time should be a feedback regarding error in behavior. An internal storage of the whole interaction with the network and result of actions taken during learning could be a primary means for performing a lot of unheard consumer satisfaction metrics. The possibility of buyer vigilance can be exploited as well.

## **11. Conclusion**

This paper identifies, understands, designs and engineers self-learning agentic systems with individualized agents that can be part of a wider system and network within the domain of supply chains in retail. It gained insights into the design possibilities and trade-offs for developing self-learning agentic systems in the retail supply network by exploring tele-neutral and tele-augment agentic systems where agents can model themselves and their environment, interact with others and learn from it, and act upon their world. The design objectives and the agentic system were evaluated in a virtual agent society in an agent-based model simulation of a retail supply network. The insights gained increase the understanding of the design options for self-learning and self-organizing agentic systems within and across domains. Such systems have algorithmic autonomy, can work on their own with little or no human interference, and are effective in a wide range of settings. This paper further contributes to the literature by combining and linking the existing frameworks into analysis-on-agent-design dimensions and thereby generating insight into agent and agentic designs.

This paper explored the theoretical side and did not cover the practical challenges of building and deploying a self-learning agentic system such as the computational complexity and required expenditure or in getting agentic systems accepted by the practitioners who need them. Further, building on this work and exploring the algorithm design challenges, rider systems, happiness limits of agents, system learning, agentic

politics, and mimicking wider agentic ecosystems would generate avenues for future research.

### 11.1. Future Trends

The Supply Network Management Systems (SNMS) ensures the quality of the products delivered and the quality of the service provided to the customer. However, the ever changing demand and supply conditions in a supply network make it difficult even for a human supply manager who is usually experienced, trained and familiar with the normal range of conditions. The quality of a human managed system thus deteriorates under disturbance conditions beyond the normal ranges which is called the robustness of the system. The strengthening of such robustness is thus a research issue. A system architecture was developed to address that issue. Under its paradigm and framework, agent systems are designed in the collaboration of human and software agents. The software agents exhibit agent characteristics of proactiveness, autonomy, social ability, and situatedness.

The human agents are represented as software agents in the form of a human agent interface. HAIs are designed as a collaborative, interactive, visual computer system, built on visualization-enhanced agents, to assortedly support different aspects and issues of the supply network management. In the presentation of supply chain management, focus is put on the Demand/supply plan and Scheduling, the 'GAP' between the Demand and Supply, and the reinforcement of the decision making capability of the human agent. Time, money and the lost customer may be lost due to the occurrence of agent-task, agent-agent, and agent-conditions conflicts. Aids in computational constraint violations, potential profit losses, potential disturbance causes, and encouragements, and the tele-operation of agents could be applied to complement the foresight ability of a human manager in coping with the conflicts. The supply network disturbances covered in the presentation are the demand disturbance, supply disturbance, and agent speed up disturbance.

## References

- [1] Karthik Chava, "Machine Learning in Modern Healthcare: Leveraging Big Data for Early Disease Detection and Patient Monitoring", *International Journal of Science and Research (IJSR)*, Volume 9 Issue 12, December 2020, pp. 1899-1910, <https://www.ijsr.net/getabstract.php?paperid=SR201212164722>, DOI: <https://www.doi.org/10.21275/SR201212164722>
- [2] Data Engineering Architectures for Real-Time Quality Monitoring in Paint Production Lines. (2020). *International Journal of Engineering and Computer Science*, 9(12), 25289-25303. <https://doi.org/10.18535/ijecs.v9i12.4587>
- [3] Vamsee Pamisetty. (2020). Optimizing Tax Compliance and Fraud Prevention through Intelligent Systems: The Role of Technology in Public Finance Innovation. *International Journal on Recent and Innovation Trends in Computing and Communication*, 8(12), 111-127. Retrieved from <https://ijritcc.org/index.php/ijritcc/article/view/11582>
- [4] Xie, Z., Li, H., Xu, X., Hu, J., & Chen, Y. (2020). Fast IR drop estimation with machine learning. *Proceedings of the 39th International Conference on Computer-Aided Design*, 1-8. <https://doi.org/10.1145/3400302.3415763>
- [5] Ghahramani, M., Qiao, Y., Zhou, M., O'Hagan, A., & Sweeney, J. (2020). AI-based modeling and data-driven evaluation for smart manufacturing processes. *IEEE/CAA Journal of Automatica Sinica*, 7(4), 1026-1037. <https://doi.org/10.1109/JAS.2020.1003114>