IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 18, 2022, accepted March 28, 2022, date of publication April 7, 2022, date of current version April 15, 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3165551*

# Toward an Automated Learning Control Architecture for Cyber-Physical Manufacturing Systems

**ILYA KOVALENKO[1], (Member, IEEE), JAMES MOYNE[2], (Member, IEEE), MINGJIE BI[3], (Graduate Student Member, IEEE), EFE C. BALTA[4], (Member, IEEE), WENYUAN MA[5], YASSINE QAMSANE[6], (Member, IEEE), XIAO ZHU[5], Z. MORLEY MAO[5], (Fellow, IEEE), DAWN M. TILBURY[2,3], (Fellow, IEEE), AND KIRA BARTON[2,3], (Senior Member, IEEE)**

[1]Department of Mechanical Engineering and Industrial & Manufacturing Engineering, Pennsylvania State University, University Park, PA 16802, USA
[2]Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109, USA
[3]Robotics Institute, University of Michigan, Ann Arbor, MI 48109, USA
[4]Automatic Control Laboratory, ETH Zürich, 8092 Zurich, Switzerland
[5]Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA
[6]Siemens Technology, Production Automation Engineering, Charlotte, NC 28273, USA

Corresponding author: Ilya Kovalenko (iqk5135@psu.edu)

This work was supported in part by the National Science Foundation (NSF) under Project NSF-1544678, and in part by the National Institute of Standards and Technology (NIST) Award 70NANB19H090.

**ABSTRACT** Manufacturers are constantly looking to enhance the performance of their manufacturing systems by improving their ability to address disruptions and disturbances, while reducing cost and maximizing quantity and quality. Even though innovative mechanisms for adaptability and flexibility continuously contribute to the smart manufacturing evolutionary process, they generally stop short of providing a capability for coordinated on-line learning. This is especially true when that learning requires exploration outside of established operational boundaries or uses artificial intelligence (as opposed to purely human intelligence) as part of the dynamic implementation of learning. In this work, we provide a vision for the development of an automated learning control architecture to extend the adaptability and flexibility capabilities of manufacturing systems. As part of this vision, we describe a set of requirements and objectives that, if addressed, provide an environment to allow distributed and automated learning across the manufacturing ecosystem. We then provide an example communication and control architecture that meets these requirements and objectives by gathering information, building a dynamic knowledge base, distributing intelligence, making decisions, and adapting the control commands sent to the equipment and people across the manufacturing ecosystem. The example architecture leverages both centralized and distributed control strategies and the ability to switch between the strategies to gather and learn from information in the system. Example case studies are provided illustrating how this architecture can be used to improve manufacturing system performance.

**INDEX TERMS** Smart manufacturing, system-level control, industrial automation, cyber-physical systems, adaptation and learning, multi-agent systems.

## I. INTRODUCTION

The need to design more adaptive and flexible manufacturing systems that perform effectively in uncertain and dynamic environments has been one of the main lessons of the recent COVID-19 pandemic crisis. During the pandemic, manufacturers struggled with fulfilling a number of new product orders and meeting customer requirements caused by a significant shift in demand [6], [7]. As shown in Figure 1, a number of research trends over the past several decades have focused on this objective. Various approaches have been developed to integrate and improve dynamic decision making and learning

The associate editor coordinating the review of this manuscript and approving it for publication was Mauro Gaggero.

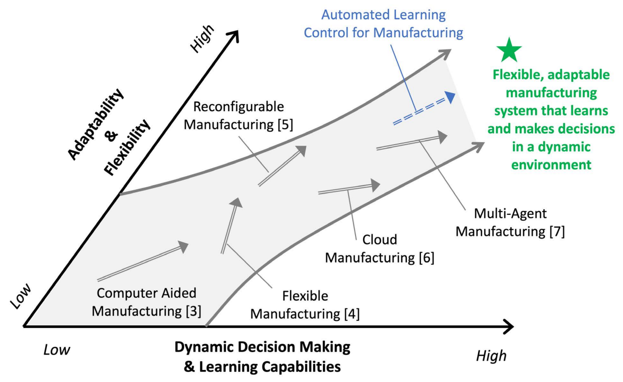

Automated Learning
Control for Manufacturing
High
Adaptability & Flexibility
Reconfigurable Manufacturing [5]
Flexible, adaptable manufacturing system that learns and makes decisions in a dynamic environment
Multi-Agent Manufacturing [7]
Cloud Manufacturing [6]
Low
Computer Aided Manufacturing [3]
Flexible Manufacturing [4]
Low
Dynamic Decision Making & Learning Capabilities
High

**FIGURE 1. The effect of the development of dynamic decision making and learning capabilities for manufacturing system adaptability and flexibility. The arrows indicate how recent trends are pushing manufacturing toward increased adaptability and flexibility through dynamic decision making and learning. The dashed, blue arrow shows how the work proposed in this paper contributes to this trend.**

within the manufacturing system control strategy to achieve more system adaptability and flexibility. However, there is still a significant amount of work required to create fully flexible and adaptable manufacturing systems that learn and make decisions in a dynamic environment. A vision for a control architecture that can move past its predefined capabilities, learn from new information, and improve its dynamic decision making capabilities based on newly obtained information needs to be realized.

Centralized and distributed control approaches are two of the most common methodologies that are used when developing control architectures [8]. In the centralized control architecture, functions are delegated to the system components (e.g., robots, workstations, etc.) to perform individual operations, while the centralized decision maker has the full authority for their initiation, coordination, and monitoring. A centralized control architecture is effective in ensuring that a manufacturing system meets customer and manufacturer requirements and objectives. However, the centralized control architecture often struggles with going past predefined capabilities and learning from new information due to the inherent complexity of using a global view of the manufacturing system (e.g., large time and computational complexity for tasks such as updating models, analyzing information). This type of architecture has been implemented in the majority of existing manufacturing systems [9].

In the distributed control architecture, a complex global control problem is divided into smaller local problems, each linked to an intelligent control unit, i.e., an agent. Due to the presence of multiple decision makers and knowledge bases, a distributed control architecture will often implement algorithms that allow the agents to move past predefined capabilities and learn from information in the manufacturing system. However, these capabilities are not often well defined and the local decisions made by the distributed agents are frequently difficult to verify as being aligned with global manufacturing objectives and constraints. Therefore, there is a need to develop a manufacturing control architecture that can explore beyond its predefined capabilities and learn from gathered information, while still making sure that the overall manufacturing system objectives and constraints are met.

Note that creating the right balance between centralized and distributed control is a challenge that is not unique in manufacturing. For example, network management, recently transformed by the adoption of software defined networking [10], is benefiting from increased programmability of the network controllers. However, how to realize the effect of logically centralized but physically distributed or hierarchical controllers [11], [12] to ensure scalability and responsiveness is still being actively explored.

In this paper, we describe a vision for automated learning in manufacturing systems that would allow the system to go beyond its predefined capabilities and automatically learn from new information. As part of this vision, we define the requirements and objectives for manufacturing systems to enable automated learning. We also propose an example control architecture that can meet the outlined requirements and objectives by enabling dynamic decision making in the manufacturing system. This example automated learning architecture implements a framework in which a centralized controller leverages intelligent, autonomous agents to gain new information from the system and push the limits of the system's predefined capabilities. The proposed automated learning architecture can enable more robust behavior and improve the flexibility of the manufacturing system, while maintaining a desirable system performance.

The rest of the paper is organized as follows. Section II presents a taxonomy to describe dynamic decision making and learning in manufacturing systems. The requirements for a general manufacturing system architecture are stated in Section III. Section IV overviews how some of the relevant existing literature (i.e., various architectures and frameworks) meets the manufacturing system architecture requirements. An example system-level architecture that meets the identified requirements is described in Section V. Some example case studies that will benefit from the proposed control architecture are provided in Section VI. Section VII lays out concluding remarks and future directions.

## II. TAXONOMY

This section provides a summary of the main concepts and terminology that are used to develop the requirements and the structure of the proposed system-level control architecture with enhanced dynamic decision making and automated learning capabilities.

### A. GENERAL TERMINOLOGY

**Manufacturing system** – a collection of resources, materials, people, and information arranged and operated to produce value-added physical products [13], [14].

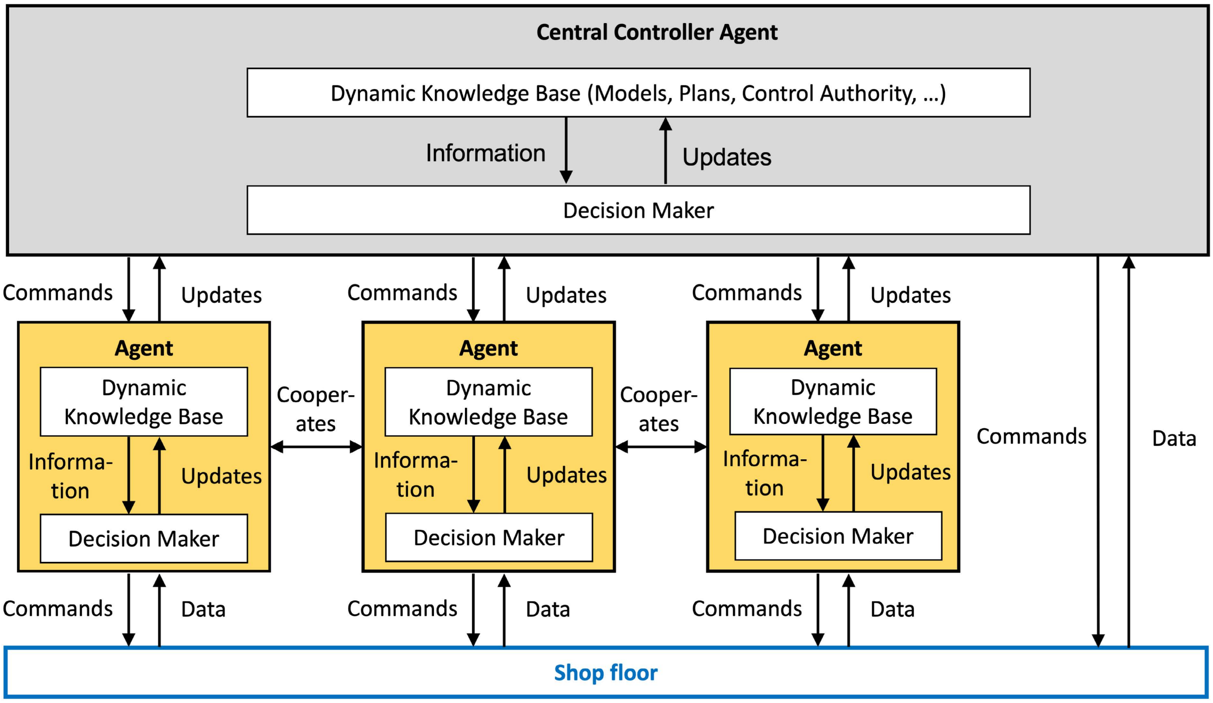


**FIGURE 2.** A simple, two-level heirarchical example of a manufacturing control architecture that consists of multiple agents, including a central controller agent.

**Manufacturing objectives** – parameters that have to be satisfied by the manufacturing system [15]. System objectives can vary widely depending on different factors through the manufacturing ecosystem (e.g., supply chain or shop floor requirements). The objectives are usually functions of manufacturing quantity, quality, and cost [16].

**Manufacturing objective metrics** – indicators that are used to measure whether the manufacturing system is meeting the manufacturing objectives. Some objective metrics are expressed as key performance indicators (KPIs).

### B. MANUFACTURING SYSTEM CAPABILITIES

The following terms are used to describe components in a system-level control architecture that enable a manufacturing system to meet its manufacturing objective.

**Agent** – a cyber element that obtains information about the environment (i.e., sense/measure), makes a decision based on its own internal objective (i.e., think/reason/decide), and affects the environment through its actions (i.e., act) [17]. For manufacturing systems, agents can be used to make decisions for individual distributed resources, parts, processes, cells, etc. Note that an agent can obtain information and affect the environment by communicating with the physical system, other agents, or other cyber elements. This communication can be accomplished using various communication protocols, such as [18], [19].

**Knowledge base** – a repository of information, data, models, etc. that is used by an agent to store and organize information. For manufacturing systems, this information can include the capabilities of the system, rules for system behavior, and models of the shop floor equipment [20]–[22]. A knowledge base can belong to an individual agent or it can be used by multiple agents to store shared information.

**Decision maker** – a component of an agent that is responsible for reasoning and determining how the agent should act [20]–[22]. The decision making component of an agent leverages the knowledge base of the agent to accomplish this task. Each agent must have an individual decision maker. Note that conflicts, such as conflicting requests or learned information from other agents, in the manufacturing system can be resolved by the decision maker through cooperation [23], [24] or by leveraging higher authority agents (e.g., a central controller) [25].

**Agent capability** – an action that the agent can perform. For example, the capabilities of a milling machine agent might include sending programming commands, scheduling various operations, and analyzing milling machine data. Capabilities can be set initially and/or could be learned over time. The collection of agent capabilities associated with a particular agent is referred to as an agent capability set.

**Capability boundary conditions** – the conditions under which an agent has the ability to perform the specified capability. Capability boundary conditions are often expressed with respect to how, where, when, and to what extent the capability exists. If an agent's decisions lead to an action that does not satisfy the capability boundary conditions, it cannot or should not perform any actions associated with that capability. Capability boundary conditions could be associated with manufacturing objective metrics, but could also be associated with other metrics not related to the manufacturing objectives, such as safety, security, and regulations (e.g., maximum carbon footprint). An example of a capability boundary condition would be a minimum temperature for achieving a minimum yield of a product for a given process, or a maximum temperature limit for all processes based on safety regulations.

A capability boundary condition could be directly tied to a particular parameter (e.g., temperature) or associated with a number of parameters (e.g., a mathematical objective function or a Boolean combination of parameters and functions). The agent's observations from the environment could also be used to define boundary conditions (e.g., can only adjust pressure when temperature is in a certain range). Capability boundary conditions could be established by the agent to define its full set of capabilities in the current environment. These boundary conditions could also be provided by another agent (e.g., a supervising or controlling agent) and may change based on the state of the manufacturing environment.

**Capability authority** – a parameter of an agent's capability that defines whether the agent has been given permission to perform that capability. If an agent does not have the authority over a capability, it is not permitted to perform the actions associated with that capability. The authority of a capability is usually granted along with a capability boundary condition tuple representing limitations to the granted authority. An agent may gain or lose capability authority over time. The authority can be managed by the agent itself or some governing entity (e.g., a supervising agent).

**Capability boundary conditions tuple** – the capability and its associated boundary conditions.

**Capability boundary conditions set** – a set of capability boundary condition tuples.

### C. SYSTEM-LEVEL MANUFACTURING CONTROL TERMINOLOGY

**Cyber-Physical Manufacturing System (CPMS)/ Cyber-Physical Production System (CPPS)** – a manufacturing system that contains a collection of physical, cyber, and cyber-physical agents associated with a defined set of objectives and a defined capability boundary conditions set. The system objectives can vary widely depending on the system and ecosystem environment. Note that the *capability boundary condition set* of the system is the superset of the capability boundary conditions set of the agents in that system.

**Manufacturing control architecture** – consists of a collection of agents and their communication infrastructure that acts on the physical environment to achieve the manufacturing objectives. The agents (or agent) control the physical components of the manufacturing system to meet a defined set of objectives. An example architecture for a manufacturing system that uses the terms defined in this section is shown in Figure 2.

**Centralized Smart Manufacturing System (CSMS)** – a Manufacturing System in which one controlling agent, termed "Central Controller" (CC) in this paper, determines the capability authority of the system [26]–[28]. This means that the CC either directly or indirectly dictates or establishes the capability boundary conditions of all the agents in the CSMS (agents may choose to further restrict boundary conditions). Oftentimes, the CC sits atop a multi-layer hierarchy of authority, e.g., an ISA-95 structured system [9].

### D. ADAPTATION AND LEARNING TERMINOLOGY

**Adaptation capability** – the ability of the agent to execute actions within the capability boundary conditions set of the agent.

**Adaptation capability authority** – is the parameter associated with the adaptation capability of an agent that defines whether the agent has been given permission to perform adaptation. This capability can be allowed/restricted by an entity such as a supervising agent or SME that has the power and capability to control that authority.

**Manufacturing agent adaptation** – refers to the action of an agent that results in changes of an agent's behavior within defined bounds of the agent capability boundary conditions set. The purpose of this action is to achieve an objective. The adaptation is usually accomplished through some combination of data collection and analysis such as machine learning, and actuation such as updating control parameters. It could also be accomplished as a result of *bounded implemented learning* (see definition below). The objective of the adapting agent can be anything and does not necessarily align with the higher-level objectives of the manufacturing system.

**Learning capability** – the ability to acquire knowledge or skills through experience, study, or by being taught. Learning is usually accomplished through some combination of data collection and analysis and some input and assimilation of intelligence, which would necessarily include some elements of Artificial Intelligence or Natural Intelligence, i.e., subject matter expert (SME) input. Note that individual agents in a CPMS could learn from information provided by other agents (e.g., a central controller updates its knowledge or decisions based on information provided by other agents) or could have the capability of self-learning (e.g., agent updates its own models and decisions based on new information from the physical system). Agent learning may result in knowledge updates for the agent, lead to new decisions based on these updates, or affect the behavior of the agent and the system.

**Learning capability authority** – is the parameter associated with the learning capability of an agent that defines whether the agent has been given permission to perform learning. This capability can be allowed/restricted by an entity such as a supervising agent or SME that has the power and capability to control that authority.

**Manufacturing agent implemented learning** – the execution of learning that results in changes in the agent's knowledge base. This execution may be accomplished purely by adaptation (e.g., an agent implementing a new method for setting a parameter within established bounds based on learning) or by an agent moving outside established capability boundary conditions (e.g., actuating a set-point outside a bound).

**Manufacturing agent bounded implemented learning** – implemented learning in which the implementation results in the agent staying inside established bounds dictated by its adaptation capability. An example of bounded implemented learning is improved granularity/fidelity of models of an agent that are used during adaptation.

**Manufacturing agent exploratory implemented learning** – implemented learning in which the implementation results in 1) the agent moving outside established bounds dictated by its adaptation capability (i.e., boundary violation) or 2) the agent having to adjust established bounds that results in a boundary set that is not fully contained in the previous boundary set (e.g., to "wider" or "shifted" bounds). This is often termed "relaxing boundary conditions". An agent often achieves this learning by both (1) relaxing the limits in the boundary condition set and (2) having incentives that allow the agent to explore beyond its existing limits. Relaxing boundary conditions happens in coordination with a supervising agent or SME.

### *E. ILLUSTRATION VIA A MANUFACTURING SYSTEM EXAMPLE*

We will use the proposed taxonomy to describe an example manufacturing system: the System-level Manufacturing and Automation Research Testbed (SMART) at the University of Michigan [29]. An overview of the testbed is shown in Figure 3. SMART has four computer numerical control (CNC) milling machines, two conveyors, and three industrial robots with an integrated industrial control system. Workblocks with RFID (radio-frequency identification) tags are placed in the system at Cell 3. The workblocks are moved to specific CNCs, where a manufacturing process creates a product feature on this workblock. Once all of the required features are created, a workblock travels back to Cell 3 to exit the system. RFID transceivers are located on the conveyors to locate and identify workblocks. The conveyor lines, robots, and CNC machines are able to handle a variety of parts using pallets, interchangeable grippers, and pneumatic clamps. SMART is a CPMS that has been used to develop and test various manufacturing control architectures. For example, in [28], SMART is converted into a CSMS with a single central controller agent to meet system throughput objectives.

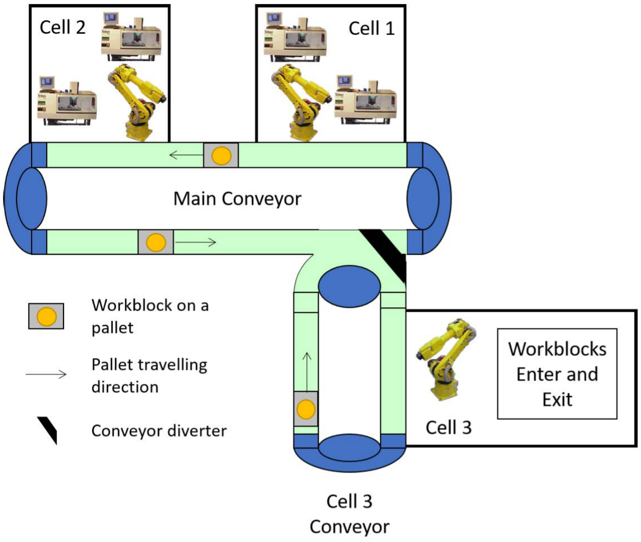


**FIGURE 3.** Overview of the System-level Manufacturing and Automation Research Testbed at the University of Michigan.

SMART has also been used to test a control architecture similar to the architecture shown in Figure 2 [30]. For the developed agent-based control architecture, a central controller agent initializes a set of resource and product agent, where a resource agent controls each manufacturing resource (e.g., CNC machine, conveyor, robot) and a product agent controls each workblock. For example, the CNC machine agent is one of the agents that is initialized by the central controller. This agent would be represented by one of the lower-level agents in Figure 2. The CNC machine agent stores information about the associated CNC machine in its knowledge base and uses the decision maker to send high-level commands to the lower-level machine controller (e.g., high level controller selects a milling program that the lower-level controller will run). The CNC machine agent has a number of capabilities, from sending commands to the lower-level controller to scheduling milling tasks to analyzing energy expenditure of the machine. Each of these capabilities is associated with boundary conditions, e.g., the machine agent cannot start a machining task if there is an operator present in the cell (safety), the tasks have a threshold limit for the amount of energy use (regulation), and the program's tool path is within some pre-defined bounds. The lower-level agents update the central controller with information about the shop floor. Then, the central controller can send commands regarding the capability authority of each agent, e.g., for a machine agent, the central controller can allow the mill to move beyond existing energy limits to ensure that the system throughput meets the system objective. The different machine agent capabilities and boundary conditions are stored as tuples in the capability boundary conditions set in the agent's dynamic knowledge base.

The machine agent has the adaptation capabilities and authority to change the manufacturing process by changing the tool path of the machine. The new tool path must remain within the pre-defined tool path bounds and must meet the safety and regulation requirements for manufacturing agent adaptation. Bounded implemented learning occurs when the machine agent finds a new tool path that can be used to complete the required product feature within the boundary conditions. If a part requires a completely new feature, the machine agent might have to move beyond previously defined capability bounds and start exploratory implemented learning. In this scenario, a new tool path that is outside the pre-defined tool path bounds and energy requirements might be required to accomplish the desired product feature. If a new tool path is found, the machine agent communicates with a higher authority (e.g., a central controller) to request to move the capability boundary conditions to complete the necessary manufacturing process.

Note that adaptation and learning are not limited to the machine agents and can be done by other resource agents and product agents in the system. However, to enable effective system adaptation and learning, a framework must be developed that allows a central controller and the agents to communicate and reason about the agents' capability boundary condition sets and capability authorities.

## III. REQUIREMENTS DEVELOPMENT

Using the taxonomy developed in Section II, we develop a vision for automated learning in CPMSs, provide the dominant characteristics of CMPSs today, perform a gap analysis to identify gaps that need to be met to achieve the proposed vision, and state the CPMS requirements and objectives to achieve the proposed vision.

### A. VISION

As noted in Section I, there is a need to develop a manufacturing control architecture that can explore beyond its defined capability boundaries and learn, while ensuring that the manufacturing system objectives and constraints continue to be met. Restated using the taxonomy developed in Section II, this need can be thought of as a *vision* for CPMSs to support automated exploratory learning in the manufacturing control architecture. This learning would be accomplished by using the capabilities of the agents to enable *manufacturing agent implemented learning*. This implemented learning would seek to further optimize or enhance manufacturing objective metrics, while maintaining a capability authority infrastructure across the system to guarantee that *all* manufacturing requirements are also met. The implementation of this vision requires that CPMSs have a manufacturing communication and control architecture that supports exploratory implemented learning. This architecture would thus have to (1) provide mechanisms to support adjustments to capability authorities and associated capability boundary condition sets across the agent base while (2) maintaining a capability authority of the entire system that ensures manufacturing requirements (including objectives and constraints) are met.

Realizing this vision places specific *requirements* on the design and implementation of the CPMS. The vision is also associated with *objectives* that the CPMS must strive to achieve if the capabilities defined in the vision are to be used effectively, e.g., as part of a continuous improvement process. In this section, these requirements and objectives are created by exploring characteristics of typical CPMSs in operation today and conducting gap analysis to identify deficiencies and challenges that prevent the achievement of the vision.

### B. CHARACTERISTICS OF A CPMS TODAY

The manufacturing control architectures of today's CPMSs vary widely, depending on a variety of factors including the objectives and requirements of the manufacturing sector and manufacturing instance, and age and location of the facility. Most CPMSs/CPPSs today are CSMSs supporting multi-layer hierarchical control and execution hierarchies with some level of alignment, at least informally, to the ISA-95 model [9]. Additionally, these systems usually have the following properties, resulting in large part from manufacturing objectives and requirements:

- Have a central controller (CC) agent that determines the capability authority of the entire system, per the CSMS definition [26]. The manufacturing control architecture might contain distributed agent-based sub-systems dedicated to specific tasks (e.g., maintenance management). However these sub-systems are not allowed to violate the capability boundary conditions established by the CC. The CC could be a fully automated cyber entity or could also be a hybrid of cyber and non-cyber components such as best practices, spreadsheets, and manual executions.
- Support various levels of manufacturing agent adaptation as part of the CSMS delegation process across the manufacturing control architecture. The adaptation could consist of fully automated cyber adaptation capabilities (e.g., model-based process control) or be a hybrid of cyber and non-cyber adaptation capabilities (e.g., manual product quality sorting based on machine vision).
- Maintain a capability boundary condition set in the CC that can only be updated through human (e.g., SME) involvement. That is, from the perspective of the CC, exploratory implemented learning can only be achieved with human involvement. The level of human involvement can vary widely depending on the application

environment and the type/potential impact of learning. For example, the SME might be the source of the learning to be implemented, or the SME might simply be in the verification and validation (V&V) loop of AI sourced knowledge updates [31], [32]. Regardless, any learning that results in a change to the system capability boundary conditions set requires some level of human approval. This process evolved naturally in manufacturing where innovation continues to be a largely human process due to tradition, manufacturing sector requirements (e.g., pharmaceutical industry [33]), general limitations of AI such as lack of Artificial General Intelligence AGI) capabilities [34], [35], and lack of trust or ability to automatically verify or quantify AI learning prior to implementation. Note that in some systems, agents below the CC might be given the capability authority to conduct exploratory implemented learning as long as the outer capability bounds of the CSMS are not violated. An example here would be permission given to a CNC milling agent to go into a learning mode at the agent's discretion to learn a new path for a part.
- Are increasingly becoming an integral part of a manufacturing ecosystem from raw materials to the customer experience [36], [37], but continue to maintain CSMS capability boundary conditions set within the defined CSMS. Note that the defined CSMS could be the ''four-walls'' of the manufacturing facility, but could also be an entire integrated manufacturing ecosystem including the entire supply chain and the customer.

### C. GAP ANALYSIS

From the perspective of achieving the vision of supporting automated exploratory learning, these are gaps in capabilities of current CPMS systems, as well as issues with AI techniques and human practices.

*The gaps in current CPMS systems* arise largely from a lack of emphasis on quantification and automation of the learning processes. These gaps include:

- The procedures for on-line implementation of learning can vary widely and are largely ad hoc. There have been significant strides as part of the smart manufacturing evolution to provide a lifecycle approach for learning development, V&V, deployment and assessment, e.g., as part of the digital twin (DT) implementation process [31]. However, issues such as diversity in learning topics and SMEs often make it difficult to realize consistent detailed procedures.
- The connection or linking of boundary conditions to manufacturing requirements is oftentimes not well-documented or well-understood quantitatively or even qualitatively. For example, many boundary conditions result from supplier documentation (e.g., maintenance practices for a machine), regulations, or simply tradition. Also, boundary conditions are often considered in isolation (e.g., maximum temperature and maximum pressure) and not part of a more complex multivariate set of condition relationships that are more representative of the application environment.
- There is little formal categorization or quantification of learning from a risk vs. reward perspective. Exploratory learning processes can be associated with widely different risks and rewards (e.g., expanding a control boundary condition versus a safety boundary condition). While these risks and rewards are often quantified and generally categorized for re-use (e.g., cost-benefit analysis procedures for adjusting a maintenance schedule), the mechanisms vary widely across CPMSs.
- There is no formalized process, structure or language for capturing and transferring learning between agents in a CPMS, especially between agents and the CC in a CSMS. As noted earlier, bringing the data together in a CPMS as part of the smart manufacturing evolution provides huge opportunities for learning. However the process for a CC to learn as a result of agent actions and communication of learning is largely non-existent or ad hoc and requires a human in the loop.
- Intelligent agents in a CPMS are oftentimes not given the freedom that would allow the system to take advantage of this intelligence. As an example, smart manufacturing systems today oftentimes have highly intelligent edge devices as part of an IIoT infrastructure; this intelligence can result from highly directed AI associated with the edge device application area or access to more detailed information closer to the application environment. CPMSs frequently do not provide the necessary flexibility to allow these ''smart'' devices to conduct exploratory learning or, as noted above, there generally is not a process for the CC to learn from these agents so as to support a more fruitful CSMS-wide exploratory learning environment.
- The available methods to ensure system security, especially data and intellectual property (IP) security are insufficient to maintain security levels in the envisioned CPMS. Data and IP security has been highlighted as the most important security issue in some manufacturing domains [38]. The distribution of information required to support many of the tenets of smart manufacturing is often gated by concerns over IP. This problem will be exacerbated by the need to communicate and share learnings as part of the CPMS vision.
- There is a lack of sufficient integration in the ecosystem and associated traceability (e.g., digital thread [39], [40] and blockchain [41], [42]) to support the CPMS vision in areas where supply chain information is a component of implemented learning. Examples include (1) (upstream) learning supplier patterns and implementing associated learning to improve throughput, and (2) (downstream) learning customer buying patterns and optimizing production to customer delivery requirements.
- Existing CPMS hardware and software may not be able to easily and effectively integrate new learning

approaches. The hardware and software found in existing CPMS were mainly developed to ensure system reliability and may need to be updated to incorporate the computation and reasoning required for enhanced dynamic decision making and learning.
- There is no standard communication mechanism to support interoperability, interchangeability, extensibility, reusability, and other ''ilities'' associated with implemented learning sharing and distribution. This communication solution could be an extension of an existing factory wide data communication solution such an ISA-95 solution provided by the Asset Administration Shell (AAS) [43] or built on existing protocols such as OPC-UA [44]. However enhancements would still be needed to support the requirements of implemented learning communication.

Gaps in AI techniques as they apply to achieving the CPMS vision result largely from the relative infancy of AI application in manufacturing. These include:

- AI learning is usually not accompanied by an understanding of the quality of or confidence in the learnings. As with any learning process, there is a possibility of error. More formalized techniques are needed to convey AI learning and associated quality or confidence such as those associated with DT predictions in a DT framework [31]. Quantification could include believability, range of impact, and thresholds on impact such as maximum negative impact on a metric.
- AI techniques in manufacturing are primarily narrow AI applications. Thus the impact of exploratory implemented learning in areas outside of the domain of the AI knowledge base are unknown to the AI system. Narrow AI techniques will need to be formally combined with techniques towards AGI or human intelligence before implemented learning can be trusted [35].
- The use of narrow AI technique also creates a learning gap in that learnings are confined to a limited understanding of aspects of the application environment. This prevents the creative type of learning associated with AGI or the SME often termed ''thinking outside of the box'' [32].
- There is no formalized process for AI-SME interaction in exploratory implemented learning. Ideally the learning process is a continuum that evolves over time from primarily SME learning to AI assisted learning to AGI and SME learning. Realizing this continuum requires developing a formalized methodology to guide AI-SME interaction throughout the learning process [32], [37].
- AI techniques have been successfully used in specific applications, such as process monitoring, optimization, etc. However, their utilization is limited at the system level of decision making due to the difficulties of data synthesis. The heterogeneous manufacturing data may contain a high degree of irrelevant, redundant, and missing information, which influences the performance and suitability of AI algorithms relative to expected results [45].

Gaps in human practices relate largely to culture, including:

- From the manufacturer SME perspective there is oftentimes a general resistance to support AI techniques when they are considered as a replacement for SMEs rather than a tool for enhancing SME capabilities. Recent efforts in this space directed at educating, training, and connecting the manufacturing workforce are meant to address this challenge [46].
- From the AI developer perspective there is oftentimes a belief that data driven solutions will solve problems and SME expertise is not necessary. Recent efforts in this space have begun to address this challenge by illustrating the importance of combining SME and AI knowledge with an objective of ''no knowledge left behind'' in smart manufacturing [32], [37].

### D. REQUIREMENTS AND OBJECTIVES

With an understanding of the CPMS vision, current practices and gaps, CPMS requirements and objectives can be formulated that are necessary to achieve the vision. Leveraging existing works that have defined requirements and objectives for general industrial AI applications [47]–[49], security [50], and distributed CPMS [51], [52], we define the primary requirements and sub-requirements for future CPMS:

R1 *Exploratory learning must be quantifiable* in terms of learning parameters that necessarily include assessment of benefit, risk, quality of (or confidence in) learning, and impact on capabilities, capability boundary conditions, and capability authorities. This necessarily places similar requirements on the intelligence mechanism used in developing the learning. *The relationship between learning, learning parameters, and capability boundary conditions must be quantifiable.* This includes any interrelationships between boundary conditions related to the learning.

R2 *A framework must be in place to support the distributed implementation of learning* throughout the system.

a) The framework must *include a mechanism to support the communication of learning among agents* within a particular level and between levels in a system.

b) The framework must *include a mechanism to support the implementation of learning among agents including any adjustments to learning parameters.* This mechanism necessarily includes methods for agents to specify, communicate and change capabilities, capability boundary conditions, and capability authority in the CPMS.

c) The framework must *extend across the entire ecosystem* as defined by the CPMS implementation which would likely include elements of the upstream and downstream supply chain.

d) The framework must *provide a level of re-usability, interoperability (e.g., an abstraction layer or a communication protocol to support different agents, agent types, or agents classes [53], [54]), interchangeability, maintainability, and extensibility to ensure overall benefit.*

R3 *An evolutionary process must be in place to support implementing learning* resulting from a continuum of human and AI knowledge.

a) *The process must support existing learning mechanisms.* This could include isolation of boundaries within a boundary set for adjustment, serial assessment of impact on critical aspects such as safety, learning from the SME (see below), and visualization and explanation of learning for SME empowering.

b) *The process must support the evolution of the implemented learning mechanisms (as abilities of and confidence in AI systems evolve).* This includes the incorporation of AGI with AI, and SME empowering and evolving role definition.

c) *A re-usable learning development and implementation lifecycle must be developed as part of the process.* The lifecycle must include verification and validation of learning prior to deployment, assessment of learning implementation, and feedback to the learning. Specifically for bounded implemented learning, it is important to develop verification and validation algorithms to ensure that the system behaves in a desired manner.

d) *The process must account for traceability of the learning* in terms of contributors, contributions and knowledge base evolution.

e) *The process must consider and address conflict resolution in the system. Agent-to-agent (e.g., [23], [24]) and non-agent-to-agent (e.g., human-to-agent [55], [56]) conflict resolution must be developed as part of the process.*

The CPMS objectives related to the vision are:

O1 *Maximize use of existing processes to enable automated exploratory implemented learning*

a) *Enable extension of the adaptation paradigm* to support adjustments based on learning for specific bounds to which confidence in learning can be applied and/or possibility of mistakes can be accepted.

b) *Use off-line and manual processes in existence for implemented learning as input to developing automated on-line learning implementation processes.* This could include existing life-cycle processes.

O2 *Provide incentives for CPMSs and their agents to learn* in a manner that is most beneficial to the CPMS.

a) *Provide quantifiable incentives to improve manufacturing objective metrics* and expand/relax capability boundary conditions performance objectives, while establishing appropriate penalties for mistakes as well as missed opportunities for learning.

b) *Provide incentives to explore outside of the ecosystem boundaries*, e.g., to achieve exploratory implemented learning that incorporates supply chain information or learning communication.

c) *Provide incentives to enhance the re-usability, interoperability, interchangeability, maintainability, and extensibility of the distributed implemented learning framework* as part of a continuous improvement process.

O3 *Provide mechanisms that support human and AI interaction* in the learning process with an objective of "no knowledge left behind."

a) *Provide methods for structured interaction between AI systems and SME networks.* Clearly define AI and SME complementary roles (e.g., innovating and approving) and boundaries. Define role evolution paths and conditions for evolution.

b) *Support and incentivize the empowering and enabling of the SME as a collaborator in CPMS* and its evolution (growth), rather than the elimination of the SME. This includes the SME learning from the AI.

c) *Support the empowering and enabling of the AI system* and its evolution. This includes incentivizing the AI system to learn from the SME and the SME to aid in the teaching of the AI system.

O4 *Ensure that appropriate levels of security, especially data and IP security, are maintained during implemented learning.* Achieving this goal will likely require a combination of applying existing and emerging capabilities such as homomorphic encryption and federated learning [57], [58].

## IV. MAPPING EXISTING WORKS TO REQUIREMENTS AND OBJECTIVES

As described in Section I, there is a need to develop a control architecture that improves system-level dynamic decision making and learning. To achieve this objective, the control architecture must meet the requirements and objectives identified in Section III. A number of existing works have defined and addressed some of these requirements and objectives. Table 1 provides a summary of different works in the manufacturing domain, including centralized control approaches, distributed control approaches, hybrid (centralized and distributed) control approaches, and reference architectures. The rest of this section provides more details about the works identified in Table 1.

### A. CENTRALIZED CONTROL APPROACHES

A common approach for control of manufacturing systems is centralized control. For this approach, a central controller has

**TABLE 1. Summary of requirements (R) and objectives (O) addressed in literature.**

| Domain | Category | Reference | R1<br>*Exploratory learning must be quantifiable* | R2<br>*Framework must support the distributed implementation of learnings* | R3<br>*Evolutionary process for implemented learning must be in place* | O1<br>*Maximize use of existing processes* | O2<br>*Provide incentives for learning* | O3<br>*Support human/ AI interaction* | O4<br>*Ensure security during learning* |
|---|---|---|---|---|---|---|---|---|---|
| Centralized Control | Application | [26], [59] | A | - | - | - | - | - | - |
| Distributed Control | Architecture | [53], [60]–[63] | - | - | - | - | - | - | - |
| | | [64] | - | D, A | D, A | - | A | - | - |
| | Requirement | [5] | - | D | D | - | - | - | - |
| | | [65] | - | D | - | D | - | D | D |
| | | [66], [67] | D | D | - | - | - | D | - |
| | | [68] | D | D | - | D | - | D | - |
| | | [69] | - | D | D | D | - | - | - |
| | Agent design | [70] | - | - | - | - | - | - | - |
| | | [71] | A | - | - | - | - | - | - |
| | | [21], [22], [24], [72] | A | A | - | - | - | - | - |
| | Application | [73], [74] | A | - | - | - | - | - | - |
| | | [75] | A | A | - | - | - | - | - |
| | | [76]–[78] | A | D, A | - | A | A | - | - |
| Hybrid Control | Application | [79], [80] | A | A | A | - | - | - | - |
| Reference Architectures | Roadmap | [81]–[84] | D | - | - | D | D | D | D |

The values in the table are "-", "D" or "A" , where "-" means the referenced do not mention the Requirement/Objective (R/O);
"D" means the references define or state the R/O; "A" means the references propose a method that addresses the R/O.
Note that none of the references *fully* define or address any of the individual requirements or objectives provided in Section III.

the necessary models, data streams, and control logic to make decisions about the production system. The controller obtains run-time data through the sensors on the plant floor [26], [59]. The data is processed by the controller to run optimization procedures (e.g., production planning, maintenance scheduling), make rule-based decisions (e.g., system reconfiguration, fault detection), or to monitor process KPIs (e.g., system throughput, yield, quality). Due to its centralized structure and access to all components on the plant floor, central controllers can optimize global objectives and implement changes across the entire system. As a result, centralized architectures are often capable of fulfilling R1 since they have a clear understanding of the global view of the CPMS, which they may utilize to define capability boundary conditions. It is important to note that such system-level exploratory learning applications are not extensively studied in the literature due to the issues in computational complexity and scalability.

As manufacturing systems get more complex and autonomous, the learning problem becomes increasingly challenging. Hierarchical architectures to enable scalability and modularity are utilized for this purpose (e.g., ISA-95). The main drawback of centralized architectures is the lack of distributed learning capabilities (i.e., requirements related to R3), which limits the applicability of exploratory learning applications in practice. A digital-twin based exploratory scenario for a manufacturing cell is given in [85]. A centralized scheduler is trained using DTs of the CPMS (satisfying R1). While R2 is not explicitly addressed, the offline training of multiple agents is proposed as a solution for distributed learning. Centralized approaches also result in heavy burden on network bandwidths due to the large amounts of data collected frequently that has to be transmitted to the central controller as well as latency and poor response time.

### B. DISTRIBUTED CONTROL APPROACHES

Another approach for the control of manufacturing systems is distributed control. For most distributed control approaches for manufacturing systems, a number of software agents make high-level decisions for various manufacturing system components (e.g., machines, physical parts, product orders, etc.) [5], [8], [65]–[67]. These decisions determine the performance of the entire manufacturing system [64]. The agents make decisions to take actions based on their own internal objectives and information about the environment [21], [24].

The focus of most existing works is to construct the overall multi-agent architectures to achieve desired manufacturing objectives. PROSA [60] and PABADIS [61] provide a general description of agent knowledge, communication, and decision-making behaviors in their multi-agent architectures. ADMARMS develops the requirements and qualitative design principles for the knowledge, functionality, and communication capabilities of its agents [53]. ADACOR provides a formal specification to describe the behavior of its proposed agents using Petri nets [62], [63]. In [21], [70], the internal architectures that enable agent communication and decision-making of two types of manufacturing agents - product agent and resource agent - are developed. However, these architectures do not provide well-defined agent capabilities, and thus cannot support quantifiable implemented learning or the communication of these learnings (R1 and R2).

A few existing works have provided definitions for agent capability, but these definitions are specific to the capabilities of manufacturing resources. In [71], [73], agent capabilities of manufacturing resources are broken down into atomic (simple) and complex (combined) capabilities. These capabilities are described in a semantic manner with several parameters acting as boundary conditions. In [74], agent capabilities are defined by a set of specifications in the form of preconditions and postconditions using propositional logic. A capability can only be executed when its preconditions are true, and the postconditions will hold if the capability ends with success. However, the capability descriptions, combination rules, and condition-based logic of these works are predefined and do not change dynamically. In [22], [24], [72], a finite state machine (FSM) is used to represent the capabilities of each manufacturing resource, which enables agents to update their capabilities dynamically as they receive information from resources in their local manufacturing environment. While introducing quantifiable and dynamic characteristics to agent capabilities, this method only passes information from the physical layer to the agent layer without offering learning mechanisms (R2).

For the learning capability in multi-agent systems, [68] and [69] mentioned explorative knowledge augmentation on a conceptual level. Most existing works apply reinforcement learning to a specific problem (e.g., production scheduling) with specified learning parameters [76]–[78]. However, these works do not contain formal agent capabilities and associated boundary conditions and do not support an evolutionary learning process with dynamic decision-making (R2 and R4). Therefore, in the current multi-agent architectures, the agent capabilities and quantifiable learning parameters are not connected, which limits the ability to support implementing learning behaviors.

### C. HYBRID (CENTRALIZED AND DISTRIBUTED) CONTROL APPROACHES

For industrial systems, [79] classifies hybrid (combined centralized and distributed) control architectures into three classes: class I contains typical centralized, hierarchical control, class III contains distributed, heterarchical control, and class II introduces a concept of a hybrid control architecture to bring the advantages of class I and class III together. ORCA-FMS [80] work further classifies class II into four sub-classes based on two criteria: whether the structure can evolve over time (static or dynamic) and whether the control is applied to all entities or not (homogeneous or heterogeneous). Based on this classification, ORCA is a dynamic and heterogeneous architecture in class II. It has both a global controller and many local controllers. The global controller has the state information of the whole system while the local controller only has the local view. It also has two operating modes: normal mode and disruptive mode. In normal mode, the global controller optimizes the global performance and gives orders to the local controller, which will execute these orders. When the local controller detects a perturbation, it switches to disruptive mode and handles the perturbation locally. Our work generalizes such classification by using the definitions in section II (e.g. agent, agent capability, agent adaptation) to describe a class II manufacturing system.

### D. REFERENCE ARCHITECTURES

In addition to the various control approaches, a number of reference architectures have also been developed that define some of the requirements and objectives. ISA-95 is a well-known standard that defines a data model for the integration of enterprise and control systems in manufacturing [9]. Given the established framework of ISA-95, efforts have been focused on infrastructure requirements for enhancing the ISA-95 with the Industrial Internet of Things (IIoT). In Germany, RAMI 4.0 [81] has been proposed as an architecture for the communication between the different system components across the value chain and introduced the idea of asset administration shell acting as a single point of information retrieval. In the US, the Industrial Internet Reference Architecture (IIRA) [82] has been introduced to create a common set of architecture requirements, characteristics, and patterns within and across industries for higher interoperability and enhanced development of Industrial Internet systems. Outside of Germany and the US, other initiatives have been introduced in Japan, China, Korea, and the UK. The Japanese Industrial Value Chain Initiative introduced the Industrial Value Chain Reference Architecture (IVRA) [83]. China has defined its own Intelligent Manufacturing System Architecture (IMSA) composed of three essential elements, namely, lifecycle, system hierarchy, and intelligent functions [84]. The UK has been focusing more on smart services instead of technology within Industry 4.0 [86].

Although the above mentioned reference architectures provide concepts and methods for developing concrete architectures, they are not specific architectures for concrete systems. The concepts and methods have to be applied to realize concrete system architectures and extensions of these reference architectures need to be addressed to realize practical solutions. However, systems developed using these reference architectures will still meet some of the requirement and objectives identified in Section III. Specifically, [81]–[84] state the need for quantifiable learnings and learning parameters (R1). These architectures also provide some methods to support human–AI interaction (O3) and to ensure the security of manufacturing systems (O4).

## V. EXAMPLE ARCHITECTURE TO SUPPORT AUTOMATED LEARNING

None of the existing system-level control architectures for manufacturing systems addresses all of the requirements or meets the objectives identified in Section III. Therefore, there is a need to develop an effective automated learning control architecture for manufacturing systems. In this section, we provide a high-level overview of a control architecture that could be extended to meet the requirements and objectives from Section III. A visualization of this example architecture is shown in Figure 2.

### A. ARCHITECTURE COMPONENTS

The proposed control architecture utilizes a number of agents to gather information, make decisions, and send commands to the equipment on the shop floor. To achieve this objective, each agent stores its capability boundary conditions set that consists of capability boundary conditions tuples in its knowledge base. Formally, the capability boundary conditions set, *Capabilities* can be represented as:

$$\begin{aligned} \mathit{Capabilities} = \{&(\mathit{capability}_1, \mathit{conditions}_1), \\ &(\mathit{capability}_2, \mathit{conditions}_2), \ldots\} \end{aligned} \tag{1}$$

where $\mathit{capability}_i$ is a capability of the agent and $\mathit{conditions}_i$ is the associated boundary conditions. Note that the capability boundary conditions set includes the *adaptation* and *learning capabilities* of the agent.

Each agent also contains a mapping that represents whether the permission to perform that capability has been provided:

$$\mathit{Authority} : \mathit{Capabilities} \rightarrow \{\mathit{true}, \mathit{false}\} \tag{2}$$

The capability boundary conditions set and the capability authority are dynamic properties of the agent and will often be altered by other agents in the system, as described in Section V-B.

In addition to the boundary conditions set, an agent stores other information in its knowledge base (e.g., production schedule, dynamic model, etc.). Various other works have explored the information that each agent needs to store to make effective decisions [20], [30], [72]. All of this information is used by agents to decide what commands need to be sent to various resources on the shop floor.

### B. STRUCTURE AND COMMUNICATION

To meet the requirements and objectives, we will leverage a CSMS-based approach to the development of the architecture. The proposed architecture contains a central controller (CC) agent that receives updates from the agents in the system, collects data from the shop floor, and receives updates from the manufacturing agents. However, in contrast to existing CSMS, the CC can set parameters for the adaptation capability and the learning capability to enable adaptation, bounded implemented learning, and exploratory implemented learning within the central controller as well as among the agents under its control.

To enable adaptation and learning, the CC gives agents the authority to perform adaptation and learning. In addition, the CC can set the boundary conditions for these capabilities. For example, the CC can enable adaptation, bounded implemented learning, and exploratory implemented learning only for specific agent capabilities. If exploratory implemented learning authority is enabled for a capability, the agent is allowed to relax existing boundary conditions and send commands to the shop floor that violate these conditions (e.g., [23]). In addition, the agent frequently updates the CC with any information that it obtains from the data gathered from the floor. The CC keeps track of this information and monitors the system to ensure that any global manufacturing objectives are satisfied.

### C. MEETING REQUIREMENTS AND OBJECTIVES

The proposed architecture can be extended to meet the requirements and objectives identified in Section III. To meet requirement R1, individual agents need to ensure that exploratory learnings are quantifiable. The CC should also ensure that it provides exploratory learning authority to capabilities that can be quantifiable. While the proposed framework provides the means to ensure an the integration of distributed learning (requirement R2), it is still necessary to develop the algorithms required to ensure that learnings are effectively communicated in the system and across the supply chain. To ensure that requirement R3 is met, the implemented learning algorithms used by the individual agents must be able to support existing learning methods and ensure the reusability of these methods. Further there must be a system in place for human and AI interaction to maintain the capabilities and authority and promote their evolution.

The proposed control architecture can be used to ensure that the CPMS objectives are satisfied. For objective O1, the proposed architecture can be leveraged to ensure existing adaptation paradigms can be extended to create exploratory learning. The CC can provide the agents with initial authority for adaptation and then start to slowly extend the boundary condition for agent learning. Also capabilities can be partitioned into on-line critical-path versus off-line operation so that exploratory learning could be accomplished at lower risk and cost. As part of the communication between the CC and the agents, the CC can provide incentives to the agents to allow for enhanced exploratory implemented learning to satisfy Objective O2. Similar, for Objective O3, there is a need to implement and develop SME-agent interfaces and interactions to ensure that there is ‘‘no knowledge left behind’’ (e.g., [55]). Finally, security protocols need to be added to the proposed architecture to ensure cybersecurity and meet Objective O4. One possible approach to enhance cybersecurity is through the extension of existing communication standards and the development of a security practices that ensure the communication between the various agents meets existing requirements. This would be complemented with a framework to support and encourage secure information and knowledge sharing.

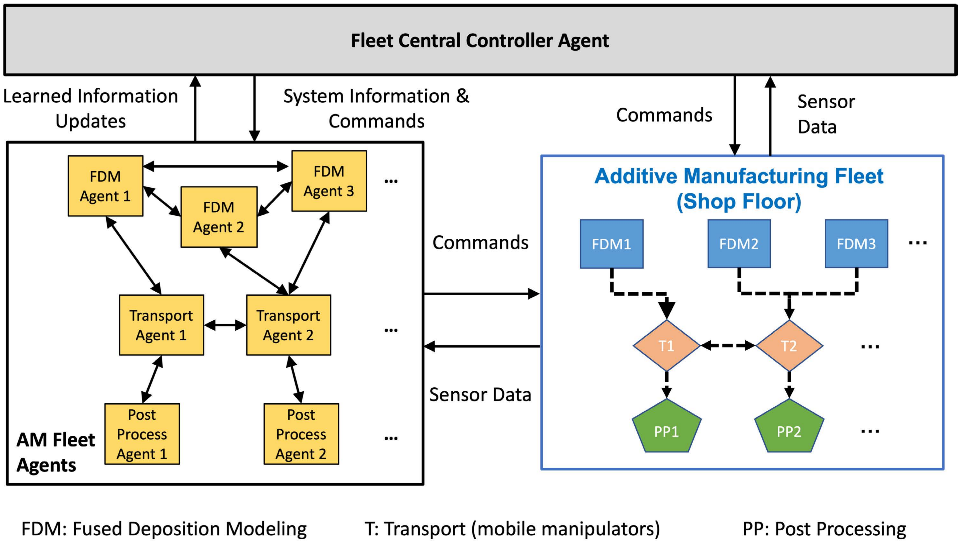


**FIGURE 4.** The setting of the AM Fleet scenario. The arrows between the agents show the communication links while the arrows on the AM Fleet show the physical connections for material flow in the system.

## VI. DYNAMIC RECONFIGURATION EXAMPLE

As part of the vision toward automated learning manufacturing systems, we provide a simple example illustrating the automated learning architecture for the additive manufacturing (AM) fleet shown in Figure 4. An AM fleet utilizes multiple AM resources in parallel for high throughput and flexibility [87]. The example AM fleet consists of machines that (1) produce the products, (2) perform post-processing (e.g., support structure removal, part cleaning, material treatment processes, finishing processes, or quality control measurements), and (3) transport the printed parts between other machines. While there may be a number of other machines in industrial AM fleets (e.g., raw material feed to the AM resources, assembly and packaging resources), we limit the conceptual AM fleet to the resources given in Figure 4 for the purpose of presentation.

We have three Fused Deposition Modeling (FDM) machines in the fleet, each producing dedicated products *P1, P2, P3*, respectively. In the nominal case, we assume that each machine is capable of producing all types of parts, but does not have the authority for reconfiguration. This nominal case describes the situation when each machine produces the corresponding product based on an average demand forecast. Therefore, for this case, each machine has a preset production schedule. The physical layout of the conceptual AM fleet is shown on the right hand side of Figure 4. The fleet has two mobile manipulators (T1, T2) responsible for material transportation between the FDM machines and the post processing stations, with the physical interconnections shown by the dashed arrow in Figure 4. The cyber-physical interconnections and data exchange are also shown in Figure 4. A fleet controller (e.g., the controller described in [26]) is in charge of production management in the system. The production schedules are provided to individual agents, which control the associated process in each machine. Agents use sensor data to control the physical resources and learn new information. Learned information is shared with the fleet controller for further analysis and learning. The fleet controller shares system information for production boundary conditions, authorities, etc., with the agents, and the agents use their belief models and controllers to send actuator commands on the physical system.

In this conceptual case study, we consider three cases with increasing dynamic decision making, flexibility, and learning capabilities. The three cases of interest presented in Figure 5, are as follows.

- *CASE-1: No Learning.* In this scenario, the agents in the system have no learning capability or implemented learning authority. Thus, the agents rely on their given models, capabilities, and authorities for production.
- *CASE-2: Bounded Implemented Learning.* The agents in this scenario have the capability to learn from the previous iterations and past process data. The learning is confined to the capability boundaries set by the fleet controller. Therefore an agent can only learn to improve within the predefined boundaries, satisfying the given constraints defined by the fleet controller. This type of learning is common in modern manufacturing systems with learning-based controllers.
- *CASE-3: Exploratory Implemented Learning.* This corresponds to the proposed learning-based adaptation and dynamic reconfiguration scenario. By utilizing the proposed exploratory implemented learning methods from this work, the agents are able to learn beyond their capability boundary conditions. The learned actions by the agents are shared with the fleet controller for further analysis and verification. The fleet controller can choose to allow the agents to implement new control

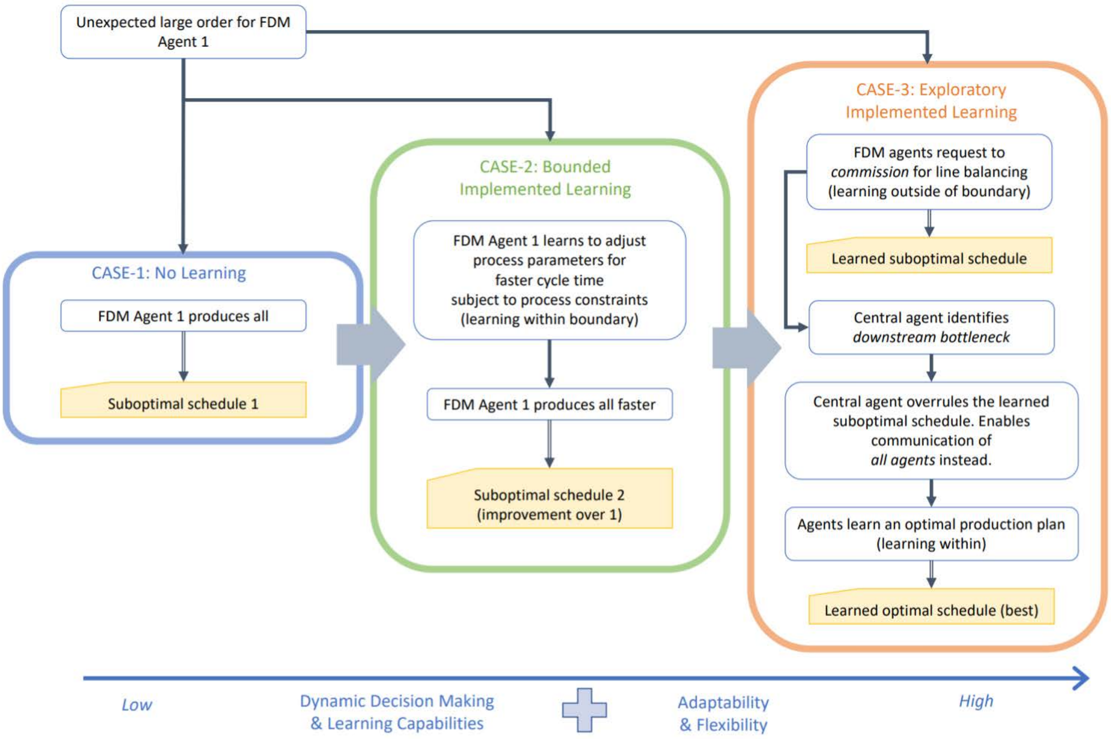

Unexpected large order for FDM Agent 1
CASE-1: No Learning
FDM Agent 1 produces all
Suboptimal schedule 1
CASE-2: Bounded Implemented Learning
FDM Agent 1 learns to adjust process parameters for faster cycle time subject to process constraints (learning within boundary)
FDM Agent 1 produces all faster
Suboptimal schedule 2 (improvement over 1)
CASE-3: Exploratory Implemented Learning
FDM agents request to *commission* for line balancing (learning outside of boundary)
Learned suboptimal schedule
Central agent identifies *downstream bottleneck*
Central agent overrules the learned suboptimal schedule. Enables communication of *all agents* instead.
Agents learn an optimal production plan (learning within)
Learned optimal schedule (best)
Low
Dynamic Decision Making & Learning Capabilities
Adaptability & Flexibility
High

**FIGURE 5.** The three cases discussed in the case study.

actions accordingly, or overrule the proposed actions of the agents and instead change their capability boundary conditions to ensure desired system performance and constraint satisfaction. We assume that the FDM agents have the capability to communicate with each other in this scenario.

In all cases, we consider an unexpected large order of *P1*, which is the product produced only by FDM1 in the nominal case. This represents an unexpected disturbance in the production system, and the responses of the AM fleet in the three cases are studied to compare their capability adapting to this unexpected disturbance. The decision making steps taken in each case are illustrated by the vertical flow charts in each corresponding case in Figure 5. In CASE-1, the agents in the system have no learning capability. Therefore, the FDM1 Agent commands the FDM1 machine to produce all the incoming orders. Since all orders are produced by a single machine, this leads to a suboptimal schedule where the order of the incoming products take a long time (compared to an optimized schedule using multiple machines), and FDM1 becomes a bottleneck.

CASE-2 represents resources with limited learning capability. Agents in the system are able to utilize data from the physical system to learn better control actions and policies within the capability boundaries defined by the fleet controller. In this case, the agent of FDM1 learns to improve the process performance by reducing the cycle time. The agent utilizes a data-driven optimization or online learning method to improve process performance while adhering to the constraints set by the fleet controller in terms of e.g., maximum process speed, allowable printing temperature, maximum layer height, and expected part performance as a function of the printing parameters. Under these constraints, the FDM1 Agent employs learning, which now results in another suboptimal schedule (compared to an optimized schedule using multiple machines), but one that is an improvement over the schedule of CASE-1.

CASE-3 describes the proposed learning-based control scenario decision flow chart. Given the same unexpectedly large order, the FDM agents now communicate with each other to commission the incoming production. This results in a balanced learned suboptimal schedule that is optimized at the level of the FDM machines. The schedule is suboptimal because it does not consider the implications of this new schedule on the downstream resources (transportation and post processing). Note that this learned schedule is outside of the initial capability boundary of the FDM agents and it is a product of learning between the agents. The FDM agents send a request to the fleet controller to implement the learned suboptimal schedule. Since the fleet controller has a centralized view of the fleet, it identifies that the learned suboptimal schedule leads to bottlenecks on the downstream since the product *P1* can only be post-processed by the PP1 station. Therefore, the fleet controller does not allow the agents to implement the given schedule, but instead uses the learned information and an understanding of the entire system to determine how to best use the resources to improve production. In this case it modifies the capability authority of the FDM agents to give them the authority to communicate with the downstream transport and post process agents to learn a new schedule. After this authorization, all agents communicate to learn an optimal schedule that optimizes the system-wide cycle time and productivity, which is the best learned optimal schedule in this example.

Note that the criteria for whether an agent should or should not use learning, bounded implemented learning, or exploratory implemented learning will be system and situationally dependent. For example, a manufacturer might have to specify that if the agent is doing exploratory implemented learning, it has to be done offline or be verified by another agent or person. However, in general, if exploratory implemented learning is supported, then bounded implemented learning should be supported. Similarly, if bounded implemented learning is not supported, then no learning should be allowed. Future work will look at setting explicit criteria for choosing the different types of learning approaches and extending this approach to automated learning for other applications and case studies.

## VII. CONCLUSION AND FUTURE WORK

A key to success in advanced manufacturing is the ability of accommodating new capabilities and pushing the boundaries of adaptability and flexibility. The explosion of AI capabilities over the past decade has created an environment where existing manufacturing infrastructure paradigms, often in place for decades, are not sufficient to the task of accommodating these new capabilities. Specifically, while these existing manufacturing paradigms can often be enhanced to support increased levels of adaptability and flexibility, they rarely can accommodate capabilities where the system actually learns and moves beyond established boundary conditions without direct human intervention.

In this paper a system-level architecture has been proposed from a requirements perspective that allows factories to extend adaptability into the domain of implemented learning and especially exploratory implemented learning. A taxonomy for implemented learning in CPMSs is presented that allows us to delineate and quantify the capabilities and limitations of agents in a coordinated system; these capabilities are defined with respect to pre-defined or initial capabilities, and capabilities that arise from the ability to adapt or learn. Using this taxonomy a vision for CPMSs is defined to support automated exploratory learning. An audit of existing CMPMs and AI systems reveals many technology gaps that must be addressed to realize this vision including procedures for implementing learning, lack of a quantification of learning and a formal process for transferring learnings between agents, and lack of sufficient guarantees that learning will provide a net benefit without negatively impacting system security and safety. A set of requirements are then developed to achieve the CPMS vision by addressing the defined technology gaps. Key requirements include the quantification of learning, the ability to communicate and make use of learning between agents, and the ability to support the transition along the continuum from pure human implemented learning to AI assisted and AI implemented learning. The vision is also associated with a number of operational objectives including leveraging existing processes, incentivizing and enabling learning through human-AI interaction and goals such as ''no knowledge left behind,'' and maintaining a secure environment throughout the learning process.

Using this requirement set, a hybrid control architecture is proposed that enables dynamic decision making and implemented learning in a CPMS. The architecture addresses the derived requirements as it includes a mechanism for communicating quantified capabilities and capability authorities. An additive manufacturing fleet example illustrates the use of the architecture in scenarios where there is no learning, bounded implemented learning, and exploratory implemented learning. Future work will focus on using a real-world manufacturing testbed or system to validate the automated learning concept and control architecture. Moving forward, even if the vision of a framework for automated exploratory learning in CPMSs is fully achieved, the learnings will only be implemented if there is trust in these learnings. This includes convincing the human element invested in the process that the learning will provide a net improvement towards manufacturing objectives and will not negatively impact metrics such as safety and security.

Quantification in the learning process (benefit, confidence, et.) as outlined in this paper is an important initial step in providing this trust, however quantification of uncertainty in AI system learning remains as a challenge [88]. One way in which the quantification and trust can be improved is to strive to align the AI learning with the corresponding human learning process. As an example, AI learning is often broken down into identification/perception and reasoning tasks with different techniques (such as neural net and symbolic AI respectively) applied to these categories [89]. The quantification of learning in each of these domains, the interaction between the two domains, and the proactive interaction with human learning as part of the business process will require further exploration [32]. As another example the design and standardization of the human-AI interface to support an interactive learning process will enhance the interoperability and interchangeability of human and AI elements in the learning process. This interface must be proactive in its engagement of these resources seeking intelligence input rather than just accommodating it. Trust and the human-AI connection represents just one of the key technical challenges in moving towards automated learning in CPMSs. The narrowness of AI and a lack of understanding of its boundaries of narrowness lead to additional issues of applicability and trust; understanding and better specifying AI solution boundaries, widening of AI, and moving towards general AI will all work towards addressing this challenge [34]. V&V of AI evolving systems and AI sourced knowledge updates (as discussed in Section III) will be particularly challenging as these

V&V systems in many cases must also ‘‘learn.’’ The maintenance of the growing and evolving knowledge base will present a number of challenges. Security and information partitioning in learning environments will present challenges that are already being observed today in supply chain connectivity in the manufacturing ecosystem and other information and knowledge sharing environments such as the cloud. Additionally, while some learnings are absolute, others may be temporal or contextually dependent. The dynamics and context richness typical in CPMSs thus represent additional challenges in knowledge management.

As a final point, a key challenge in moving towards automated learnings in CPMSs might be termed the human exploitation challenge. There must be constant attention to respect the human element in systems. This respect can take many forms including ensuring fairness in learning and learnings sharing, ensuring lack of bias to factors that do not relate to CPMS objectives, and avoiding exploitation of the human element in any way. Regardless of these human-AI interaction challenges, the importance of ‘‘no human left behind’’ can not be understated. Learning at the highest level will always be human guided, with delegation of learning capabilities at lower levels increasingly delegated to AI or AI guided systems. This process of moving through the human to AI continuum towards factories that learn automatically requires strong cooperation from both sides and a corresponding commitment to the value that each side will provide to the ultimate solution.

## ACKNOWLEDGMENT

The authors would like to thank Allison White, John Faris, the Integrated Manufacturing Process Automation and Control Technologies (IMPACT) Group, and the Barton-Tilbury Groups with the University of Michigan for their input to the discussions.

## REFERENCES

[1] T.-C. Chang, R. A. Wysk, and H.-P. Wang, *Computer-Aided Manufacturing*. Upper Saddle River, NJ, USA: Prentice-Hall, 1991.

[2] J. Browne, D. Dubois, K. Rathmill, and S. P. Sethi, ‘‘Classification of flexible manufacturing systems,’’ *FMS Mag.*, vol. 2, no. 2, pp. 114–117, 1984.

[3] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel, ‘‘Reconfigurable manufacturing systems,’’ *CIRP Ann.*, vol. 48, no. 2, pp. 527–540, 1999.

[4] X. Xu, ‘‘From cloud computing to cloud manufacturing,’’ *Robot. Comput.-Integr. Manuf.*, vol. 28, no. 1, pp. 75–86, Feb. 2012.

[5] P. Leitão, ‘‘Agent-based distributed manufacturing control: A state-of-the-art survey,’’ *Eng. Appl. Artif. Intell.*, vol. 22, no. 7, pp. 979–991, 2009.

[6] A. Kumar, S. Luthra, S. K. Mangla, and Y. Kazançoğlu, ‘‘COVID-19 impact on sustainable production and operations management,’’ *Sustain. Oper. Comput.*, vol. 1, pp. 1–7, Jan. 2020.

[7] X. Li, B. Wang, C. Liu, T. Freiheit, and B. I. Epureanu, ‘‘Intelligent manufacturing systems in COVID-19 pandemic and beyond: Framework and impact assessment,’’ *Chin. J. Mech. Eng.*, vol. 33, no. 1, pp. 1–5, Dec. 2020.

[8] P. Leitao, V. Marik, and P. Vrba, ‘‘Past, present, and future of industrial agent applications,’’ *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 2360–2372, Nov. 2013.

[9] B. Scholten, *The Road to Integration: A Guide to Applying the ISA-95 Standard in Manufacturing*. Paris, France, ISA, 2007.

[10] H. Kim and N. Feamster, ‘‘Improving network management with software defined networking,’’ *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 114–119, Feb. 2013.

[11] E. Amiri, E. Alizadeh, and K. Raeisi, ‘‘An efficient hierarchical distributed SDN controller model,’’ in *Proc. 5th Conf. Knowl. Based Eng. Innov. (KBEI)*, Feb. 2019, pp. 553–557.

[12] Y. E. Oktian, S. Lee, H. Lee, and J. Lam, ‘‘Distributed SDN controller system: A survey on design choice,’’ *Comput. Netw.*, vol. 121, pp. 100–111, Jul. 2017.

[13] A. Caggiano, *Manufacturing System*. Berlin, Germany: Springer, 2014, pp. 830–836.

[14] Y. Qamsane, A. Tajer, and A. Philippot, ‘‘Towards an approach of synthesis, validation and implementation of distributed control for AMS by using events ordering relations,’’ *Int. J. Prod. Res.*, vol. 55, no. 21, pp. 6235–6253, Nov. 2017.

[15] J. J. Michalek, O. Ceryan, P. Y. Papalambros, and Y. Koren, ‘‘Balancing marketing and manufacturing objectives in product line design,’’ *J. Mech. Design*, vol. 128, no. 6, pp. 1196–1204, Nov. 2006.

[16] D. S. Cochran, J. F. Arinez, J. W. Duda, and J. Linck, ‘‘A decomposition approach for manufacturing system design,’’ *J. Manuf. Syst.*, vol. 20, no. 6, pp. 371–389, Jan. 2001.

[17] S. Franklin and A. Graesser, ‘‘Is It an agent, or just a program?: A taxonomy for autonomous agents,’’ in *Intelligent Agents III Agent Theories, Architectures, and Languages (ATAL)1996*, Lecture Notes in Computer Science (Lecture Notes in Artificial Intelligence), vol. 1193, J. P. Müller, M. J. Wooldridge, N .R. Jennings Eds. Berlin, Germany: Springer, 1997, doi: 10.1007/BFb0013570.

[18] F. Bellifemine, A. Poggi, and G. Rimassa, ‘‘Developing multi-agent systems with JaDE,’’ in *Intelligent Agents VII Agent Theories Architectures and Languages(ATAL)*, (Lecture Notes in Computer Science) vol. 1986, C. Castelfranchi, Y. Lespérance Eds. Berlin, Germany: Springer, 2001, doi: 10.1007/3-540-44631-1_7.

[19] C. Liu, H. Vengayil, Y. Lu, and X. Xu, ‘‘A cyber-physical machine tools platform using OPC UA and MTConnect,’’ *J. Manuf. Syst.*, vol. 51, pp. 61–74, Apr. 2019.

[20] W. Lepuschitz, A. Zoitl, M. Vallée, and M. Merdan, ‘‘Toward self-reconfiguration of manufacturing systems using automation agents,’’ *IEEE Trans. Syst., Man, Cybern., C (Appl. Rev.)*, vol. 41, no. 1, pp. 52–69, Jan. 2011.

[21] I. Kovalenko, D. Tilbury, and K. Barton, ‘‘The model-based product agent: A control oriented architecture for intelligent products in multi-agent manufacturing systems,’’ *Control Eng. Pract.*, vol. 86, pp. 105–117, May 2019.

[22] S. Rehberger, L. Spreiter, and B. Vogel-Heuser, ‘‘An agent-based approach for dependable planning of production sequences in automated production systems,’’ *At-Automatisierungstechnik*, vol. 65, no. 11, pp. 766–778, Nov. 2017.

[23] I. Kovalenko, E. C. Balta, D. M. Tilbury, and K. Barton, ‘‘Cooperative product agents to improve manufacturing system flexibility: A model-based decision framework,’’ *IEEE Trans. Autom. Sci. Eng.*, early access, Mar. 17, 2022, doi: 10.1109/TASE.2022.3156384.

[24] I. Kovalenko, D. Ryashentseva, B. Vogel-Heuser, D. Tilbury, and K. Barton, ‘‘Dynamic resource task negotiation to enable product agent exploration in multi-agent manufacturing systems,’’ *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2854–2861, Jul. 2019.

[25] T. N. Wong, C. W. Leung, K. L. Mak, and R. Y. K. Fung, ‘‘Dynamic shopfloor scheduling in multi-agent manufacturing systems,’’ *Expert Syst. Appl.*, vol. 31, no. 3, pp. 486–494, Oct. 2006.

[26] F. Lopez, Y. Shao, Z. M. Mao, J. Moyne, K. Barton, and D. Tilbury, ‘‘A software-defined framework for the integrated management of smart manufacturing systems,’’ *Manuf. Lett.*, vol. 15, pp. 18–21, Jan. 2018.

[27] Y. Qamsane, C.-Y. Chen, E. C. Balta, B.-C. Kao, S. Mohan, J. Moyne, D. Tilbury, and K. Barton, ‘‘A unified digital twin framework for real-time monitoring and evaluation of smart manufacturing systems,’’ in *Proc. IEEE 15th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2019, pp. 1394–1401.

[28] M. Potok, C.-Y. Chen, S. Mitra, and S. Mohan, ‘‘SDCWorks: A formal framework for software defined control of smart manufacturing systems,’’ in *Proc. ACM/IEEE 9th Int. Conf. Cyber-Phys. Syst. (ICCPS)*, Apr. 2018, pp. 88–97.

[29] I. Kovalenko, M. Saez, K. Barton, and D. Tilbury, ‘‘SMART: A system-level manufacturing and automation research testbed,’’ *Smart Sustain. Manuf. Syst.*, vol. 1, no. 1, pp. 232–261, 2017.

[30] I. Kovalenko, ‘‘Intelligent product agents for multi-agent control of manufacturing systems,’’ Ph.D. dissertation, Dept. Mech. Eng., Univ. Michigan, Ann Arbor, MI, USA, 2020.

[31] J. Moyne, Y. Qamsane, E. C. Balta, I. Kovalenko, J. Faris, K. Barton, and D. M. Tilbury, "A requirements driven digital twin framework: Specification and opportunities," *IEEE Access*, vol. 8, pp. 107781–107801, 2020.

[32] J. Moyne, "Bringing subject matter expertise and analytics together in solutions for microelectronics smart manufacturing: Consensus domains and approaches," in *Proc. Adv. Process Control Conf. XXXI* San Antonio, TX, USA, Oct. 2019. [Online]. Available: http://apcconference.com

[33] S. Romero-Torres, J. Moyne, and M. Kidambi, "Towards pharma 4.0; Leveraging lessons and innovation from Silicon valley," *Amer. Pharmaceutical Rev.*, vol. 5, pp. 132–141, May 2017.

[34] J. Kahn, "The quest for human-level AI," *Fortune*, vol. 181, no. 2, pp. 62–70, Feb. 2020.

[35] B. Goertzel and E. Pennachin, *Artificial General Intelligence. Cognitive Technologies*. Berlin, Germany: Springer, 2007.

[36] J. Moyne, E. Balta, I. Kovalenko, Y. Qamsane, and K. Barton, "The digital twin in the manufacturing ecosystem of the future," in *Proc. 1st Int. Workshop Smart Manuf. Modeling Anal.*, Apr. 2019. [Online]. Available: http://sdc-mfg.engin.umich.edu/roleofdtingreenfield/

[37] "Factory integration," in *Int. Roadmap Devices Syst.*, 2021. [Online]. Available: https://irds.ieee.org/editions/2021

[38] J. Moyne, S. Mashiro, and D. Gross, "Determining a security roadmap for the microelectronics industry," in *Proc. 29th Annu. SEMI Adv. Semiconductor Manuf. Conf. (ASMC)*, Apr. 2018, pp. 291–294.

[39] P. Ramdasi and P. Ramdasi, "Industry 4.0: Opportunities for analytics," in *Proc. IEEE Punecon*, Nov. 2018, pp. 1–5.

[40] A. M. Crowell, S. A. Cramer, J. P. Lopez, E. K. Rashka, R. C. Reuss, V. K. Wordsworth, and G. C. Lewin, "Development of a quality assurance tool for additive manufacturing," in *Proc. Syst. Inf. Eng. Design Symp. (SIEDS)*, Apr. 2018, pp. 76–81.

[41] M. S. Rahman, I. Khalil, N. Moustafa, A. P. Kalapaaking, and A. Bouras, "A blockchain-enabled privacy-preserving verifiable query framework for securing cloud-assisted industrial Internet of Things systems," *IEEE Trans. Ind. Informat.*, early access, Aug. 18, 2021, doi: 10.1109/TII.2021.3105527.

[42] J. Francis, *Enterprise Blockchain for Supply Chain*. Piscataway, NJ, USA: IEEE Press, 2019.

[43] X. Ye and S. H. Hong, "Toward industry 4.0 components: Insights into and implementation of asset administration shells," *IEEE Ind. Electron. Mag.*, vol. 13, no. 1, pp. 13–25, Mar. 2019.

[44] *Unified Architecture*. Accessed: Apr. 7, 2022. [Online]. Available: https://opcfoundation.org/about/opc-technologies/opc-ua/

[45] J. F. Arinez, Q. Chang, R. X. Gao, C. Xu, and J. Zhang, "Artificial intelligence in advanced manufacturing: Current status and future outlook," *J. Manuf. Sci. Eng.*, vol. 142, no. 11, Nov. 2020, Art. no. 110804.

[46] National Science and Technology Council's Subcommittee on Advanced Manufacturing, *Strategy for American Leadership in Advanced Manufacturing*, U.S. National Science & Technology Council, Washington, DC, USA, Oct. 2018.

[47] M. W. Hoffmann, R. Drath, and C. Ganz, "Proposal for requirements on industrial AI solutions," in *Machine Learning for Cyber Physical Systems*. Berlin, Germany: Springer Vieweg, 2021, pp. 63–72.

[48] O. S. Pianykh, G. Langs, M. Dewey, D. R. Enzmann, C. J. Herold, S. O. Schoenberg, and J. A. Brink, "Continuous learning AI in radiology: Implementation principles and early applications," *Radiology*, vol. 297, no. 1, pp. 6–14, Oct. 2020.

[49] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol. (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[50] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–35, 2018.

[51] M. Van Steen, "Distributed systems principles and paradigms," *Network*, vol. 2, p. 28, Feb. 2002.

[52] M. A. Faruque, F. Regazzoni, and M. Pajic, "Design methodologies for securing cyber-physical systems," in *Proc. Int. Conf. Hardw./Softw. Codesign Syst. Synth. (CODES+ISSS)*, Oct. 2015, pp. 30–36.

[53] A. M. Farid and L. Ribeiro, "An axiomatic design of a multiagent reconfigurable mechatronic system architecture," *IEEE Trans. Ind. Informat.*, vol. 11, no. 5, pp. 1142–1155, Oct. 2015.

[54] L. A. Cruz Salazar, D. Ryashentseva, A. Lüder, and B. Vogel-Heuser, "Cyber-physical production systems architecture based on multi-agent's design pattern—Comparison of selected approaches mapping four agent patterns," *Int. J. Adv. Manuf. Technol.*, vol. 105, no. 9, pp. 4005–4034, 2019.

[55] G. Zheng, I. Kovalenko, K. Barton, and D. Tilbury, "Integrating human operators into agent-based manufacturing systems: A table-top demonstration," *Proc. Manuf.*, vol. 17, pp. 326–333, Jan. 2018.

[56] N. Schurr, J. Marecki, J. P. Lewis, M. Tambe, and P. Scerri, "The defacto system: Coordinating human-agent teams for the future of disaster response," in *Multiagent Systems, Artificial Societies, and Simulated Organizations (International Book Series)*, vol. 15, R. H. Bordini, M. Dastani, J. Dix, A. El Fallah Seghrouchni Eds. Boston, MA, USA: Springer, 2005, doi: 10.1007/0-387-26350-08.

[57] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[58] R. Kontar, N. Shi, X. Yue, S. Chung, E. Byon, M. Chowdhury, J. Jin, W. Kontar, N. Masoud, M. Nouiehed, C. E. Okwudire, G. Raskutti, R. Saigal, K. Singh, and Z.-S. Ye, "The internet of federated things (IoFT)," *IEEE Access*, vol. 9, pp. 156071–156113, 2021.

[59] Y. Qamsane, E. C. Balta, J. Moyne, D. Tilbury, and K. Barton, "Dynamic rerouting of cyber-physical production systems in response to disruptions based on SDC framework," in *Proc. Amer. Control Conf. (ACC)*, Jul. 2019, pp. 3650–3657.

[60] H. Van Brussel, J. Wyns, P. Valckenaers, L. Bongaerts, and P. Peeters, "Reference architecture for holonic manufacturing systems: PROSA," *Comput. Ind.*, vol. 37, no. 3, pp. 255–274, Nov. 1998.

[61] A. Lüder, J. Peschke, T. Sauter, S. Deter, and D. Diep, "Distributed intelligence for plant automation based on multi-agent systems: The PABADIS approach," *Prod. Planning Control*, vol. 15, no. 2, pp. 201–212, Mar. 2004.

[62] P. Leitão and F. Restivo, "ADACOR: A holonic architecture for agile and adaptive manufacturing control," *Comput. Ind.*, vol. 57, no. 2, pp. 121–130, Feb. 2006.

[63] P. Leitão, A. W. Colombo, and F. Restivo, "A formal specification approach for holonic control systems: The ADACOR case," *Int. J. Manuf. Technol. Manage.*, vol. 8, nos. 1–3, pp. 37–57, 2006.

[64] J. Barbosa, P. Leitão, E. Adam, and D. Trentesaux, "Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution," *Comput. Ind.*, vol. 66, pp. 99–111, Jan. 2015.

[65] W. Shen, Q. Hao, H. J. Yoon, and D. H. Norrie, "Applications of agent-based systems in intelligent manufacturing: An updated review," *Adv. Eng. Informat.*, vol. 20, no. 4, pp. 415–431, Oct. 2006.

[66] B. Vogel-Heuser, J. Lee, and P. Leitão, "Agents enabling cyber-physical production systems," *At-Automatisierungstechnik*, vol. 63, no. 10, pp. 777–789, 2015.

[67] P. Leitao, S. Karnouskos, L. Ribeiro, J. Lee, T. Strasser, and A. W. Colombo, "Smart agents in industrial cyber-physical systems," *Proc. IEEE*, vol. 104, no. 5, pp. 1086–1101, Mar. 2016.

[68] Z. Kemény, J. Váncza, L. Wang, and X. V. Wang, "Human–robot collaboration in manufacturing: A multi-agent view," in *Advanced Human-Robot Collaboration in Manufacturing*, L. Wang, X. V. Wang, J. Váncza, Z. Kemény Eds. Cham, Switzerland: Springer, 2021, doi: 10.1007/978-3-030-69178-3_1.

[69] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," *Auto. Agents Multi-Agent Syst.*, vol. 11, no. 3, pp. 387–434, Nov. 2005.

[70] S. Rehberger, L. Spreiter, and B. Vogel-Heuser, "An agent approach to flexible automated production systems based on discrete and continuous reasoning," in *Proc. IEEE Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2016, pp. 1249–1256.

[71] E. Järvenpää, N. Siltala, O. Hylli, and M. Lanz, "The development of an ontology for describing the capabilities of manufacturing resources," *J. Intell. Manuf.*, vol. 30, no. 2, pp. 959–978, Feb. 2019.

[72] M. Bi, I. Kovalenko, D. M. Tilbury, and K. Barton, "Dynamic resource allocation using multi-agent control for manufacturing systems," *IFAC-PapersOnLine*, vol. 54, no. 20, pp. 488–494, 2021.

[73] N. Antzoulatos, E. Castro, L. de Silva, A. D. Rocha, S. Ratchev, and J. Barata, "A multi-agent framework for capability-based reconfiguration of industrial assembly systems," *Int. J. Prod. Res.*, vol. 55, no. 10, pp. 2950–2960, May 2017.

[74] R. C. Cardoso, L. A. Dennis, and M. Fisher, "Plan library reconfigurability in BDI agents," in *Engineering Multi-Agent Systems. (EMAS2019)*, (Lecture Notes in Computer Science) vol. 12058, L. Dennis, R. Bordini, Y. Lespérance Eds. Cham, Switzerland: Springer, 2020, doi: 10.1007/978-3-030-51417-4_10.

[75] A. Salvador Palau, M. H. Dhada, and A. K. Parlikad, "Multi-agent system architectures for collaborative prognostics," *J. Intell. Manuf.*, vol. 30, no. 8, pp. 2999–3013, Dec. 2019.

[76] S. Qu, J. Wang, S. Govil, and J. O. Leckie, "Optimized adaptive scheduling of a manufacturing process system with multi-skill workforce and multiple machine types: An ontology-based, multi-agent reinforcement learning approach," *Proc. CIRP*, vol. 57, pp. 55–60, Jan. 2016.

[77] R. Lu, Y.-C. Li, Y. Li, J. Jiang, and Y. Ding, "Multi-agent deep reinforcement learning based demand response for discrete manufacturing systems energy management," *Appl. Energy*, vol. 276, Oct. 2020, Art. no. 115473.

[78] Y. G. Kim, S. Lee, J. Son, H. Bae, and B. D. Chung, "Multi-agent system and reinforcement learning approach for distributed intelligence in a flexible smart manufacturing system," *J. Manuf. Syst.*, vol. 57, pp. 440–450, Oct. 2020.

[79] D. Trentesaux, "Distributed control of production systems," *Eng. Appl. Artif. Intell.*, vol. 22, no. 7, pp. 971–978, Oct. 2009.

[80] C. Pach, T. Berger, T. Bonte, and D. Trentesaux, "ORCA-FMS: A dynamic architecture for the optimized and reactive control of flexible manufacturing scheduling," *Comput. Ind.*, vol. 65, no. 4, pp. 706–720, May 2014.

[81] M. Hankel and B. Rexroth, "The reference architectural model industrie 4.0 (RAMI 4.0)," in *Proc. ZVEI*, vol. 410, Apr. 2015, p. 2.

[82] S.-W. Lin, B. Miller, J. Durand, R. Joshi, P. Didier, A. Chigani, R. Torenbeek, D. Duggal, R. Martin, and G. Bleakley, "Industrial internet reference architecture," Industrial Internet Consortium (IIC), Needham, MS, USA, Tech. Rep. 304, 2015.

[83] Y. Nishioka, "The industrial value chain initiative: A Japanese contribution to smart manufacturing, Hannover Messe," Industrial Value Chain Initiative, Hannover, Germany, Tech. Rep. 28, 2016.

[84] S. Wei, J. Hu, Y. Cheng, Y. Ma, and Y. Yu, "The essential elements of intelligent manufacturing system architecture," in *Proc. 13th IEEE Conf. Autom. Sci. Eng. (CASE)*, Aug. 2017, pp. 1006–1011.

[85] K. Xia, C. Sacco, M. Kirkpatrick, C. Saidy, L. Nguyen, A. Kircaliali, and R. Harik, "A digital twin to train deep reinforcement learning agent for smart manufacturing plants: Environment, interfaces and intelligence," *J. Manuf. Syst.*, vol. 58, pp. 210–230, Jan. 2021.

[86] J. Huxtable and D. Schaefer, "On servitization of the manufacturing industry in the U.K.," *Proc. CIRP*, vol. 52, pp. 46–51, Jan. 2016.

[87] E. C. Balta, D. M. Tilbury, and K. Barton, "A centralized framework for system-level control and management of additive manufacturing fleets," in *Proc. IEEE 14th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2018, pp. 1071–1078.

[88] C. Q. Choi, "7 revealing ways AIs fail: Neural networks can be disastrously brittle, forgetful, and surprisingly bad at math," *IEEE Spectr.*, vol. 58, no. 10, pp. 42–47, Oct. 2021.

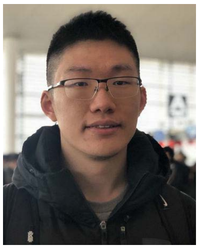
[89] E. Strickland, "The turbulent past and uncertain future of AI: Is there a way out of AI's boom-and-bust cycle?" *IEEE Spectr.*, vol. 58, no. 10, pp. 26–31, Oct. 2021.

**ILYA KOVALENKO** (Member, IEEE) received the B.S. degree in mechanical engineering from the Georgia Institute of Technology, Atlanta, GA, USA, in 2015, and the M.S. and Ph.D. degrees in mechanical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2018 and 2020, respectively. He held a postdoctoral research position with the University of Michigan, from Fall 2020 to Fall 2021. In January 2022, he joined the Department of Mechanical Engineering and the Department of Industrial & Manufacturing Engineering, Pennsylvania State University, where he currently holds the position of an Assistant Professor. His research interests include control theory and application, with a focus on multi-agent systems, advanced manufacturing, cyber-physical systems, and robotics.

**JAMES MOYNE** (Member, IEEE) received the B.S.E.E. and B.S.E. degrees in mathematics, in 1984, and the M.S.E.E. and Ph.D. degrees from the University of Michigan, Ann Arbor, in 1985 and 1990, respectively. He is currently an Associate Research Scientist with the Department of Mechanical Engineering, University of Michigan. He is also a Consultant for standards and technology at Applied Materials, focusing on smart manufacturing initiatives. He has experience in model-based process control, prediction technology (predictive maintenance, virtual metrology, and yield prediction), and smart manufacturing topics, such as big data, digital twin, and analytics. He is the author of a number of refereed publications and holds patents in each of these areas. He is the Co-Chair of the Factory Integration Thrust of the International Roadmap for Devices and Systems (IRDS), and the Information and Control Committee of SEMI, which is the semiconductor manufacturing standards organization. He is on the advisory board of Pharma (smart manufacturing in pharmaceuticals) and the CTO of the Integrated Measurement Association.

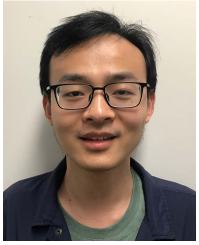
**MINGJIE BI** (Graduate Student Member, IEEE) received the B.S. degree in marine engineering from the Huazhong University of Science and Technology, China, in 2018, and the M.S. degree in mechanical engineering from the University of Michigan, Ann Arbor, in 2020, where he is currently pursuing the Ph.D. degree with the Robotics Institute. His research interests include system-level control, multi-agent systems, smart manufacturing, agile supply chain, and robotics.

**EFE C. BALTA** (Member, IEEE) received the B.S. degree in manufacturing engineering from the Mechanical Engineering Faculty, Istanbul Technical University, Turkey, in 2016, and the M.S. and Ph.D. degrees in mechanical engineering from the University of Michigan, Ann Arbor, in 2018 and 2021, respectively. Since June 2021, he has been a Postdoctoral Researcher with the Automatic Control Laboratory (IfA), ETH Zürich, Switzerland. His research interests include control theory, learning-based control methods, additive manufacturing, robotics, optimization, manufacturing science, and systems engineering.



**WENYUAN MA** received the B.S. degree in computer science from Cornell University, in 2020. He is currently pursuing the Ph.D. degree in computer science and engineering with the University of Michigan, Ann Arbor. His research interests include computer systems and networking.

**YASSINE QAMSANE** (Member, IEEE) received the M.Sc. and Ph.D. degrees in electrical engineering, telecommunications, and industrial technologies from Cadi Ayyad University, Marrakech, Morocco, in 2012 and 2018, respectively. He held a postdoctoral research position with the University of Michigan, Ann Arbor, from March 2018 to May 2021. He is currently a Research Scientist at Siemens Technology, USA, where he is currently working on evaluating and developing technologies for future automation systems with a focus on automation engineering and programming, requirements engineering, information modeling, and robotics.



**XIAO ZHU** received the B.S. degree in electronic engineering from Tsinghua University, China, in 2016. He is currently pursuing the Ph.D. degree in computer science and engineering with the University of Michigan, Ann Arbor. His research interests include the broad area of computer systems and networking.

**Z. MORLEY MAO** (Fellow, IEEE) received the B.S. degree (Hons.) in electrical engineering and computer science and the Ph.D. degree in computer science from the University of California at Berkeley, in 1994 and 2003, respectively. In 2004, she joined the CSE Division, EECS Department, University of Michigan, Ann Arbor, where she is currently a Professor. Her research interests include computer networks, network and system security, and mobile systems.

**DAWN M. TILBURY** (Fellow, IEEE) received the B.S. degree *(summa cum laude)* in electrical engineering from the University of Minnesota, in 1989, and the M.S. and Ph.D. degrees in electrical engineering and computer sciences from the University of California at Berkeley, Berkeley, in 1992 and 1994, respectively. In 1995, she joined the Mechanical Engineering Department, University of Michigan, Ann Arbor, where she is currently a Professor, with a joint appointment as a Professor of EECS. In June of 2017, she became the Assistant Director of Engineering with the National Science Foundation, while maintaining her position with the University of Michigan. She has published more than 150 articles in refereed journals and conference proceedings. Her research interests include control systems, including applications to robotics and manufacturing systems.

**KIRA BARTON** (Senior Member, IEEE) received the B.Sc. degree in mechanical engineering from the University of Colorado at Boulder, in 2001, and the M.Sc. and Ph.D. degrees in mechanical engineering from the University of Illinois at Urbana–Champaign, in 2006 and 2010, respectively. She held a postdoctoral research position with the University of Illinois, from Fall 2010 to Fall 2011, at which point she joined the Mechanical Engineering Department, University of Michigan, Ann Arbor. She is currently an Associate Professor with the Department of Mechanical Engineering and a Core Robotics Faculty Member of the University of Michigan. She conducts research in modeling, sensing, and control for applications in advanced manufacturing and robotics, with a specialization in cooperative learning and smart manufacturing. From 2018 to 2021, she was selected as a Miller Faculty Scholar. She was a recipient of an NSF CAREER Award, in 2014, the 2015 SME Outstanding Young Manufacturing Engineer Award, the 2015 University of Illinois, Department of Mechanical Science and Engineering Outstanding Young Alumni Award, the 2016 University of Michigan, Department of Mechanical Engineering Department Achievement Award, and the 2017 ASME Dynamic Systems and Control Young Investigator Award.