A Connective Framework to Support the Lifecycle of Cyber–Physical Production Systems

This article envisions a connective framework to support the engineering of cyber–physical production systems (CPPSs) in smart manufacturing through the use of a set of digital twins consistent with the real system throughout its lifecycle.

By Robert Harrison, Daniel A. Vera, and Bilal Ahmad , *Senior Member IEEE*

ABSTRACT | The potential benefits of the adoption of cyber–physical production systems (CPPSs) and their significant role in enabling smart manufacturing is well recognized today. However, it is less clear how such CPPS can be most effectively and consistently engineered and maintained throughout their lifecycle due to the existing divide in the information technology (IT) and operational technology (OT) landscape and *ad hoc* integration practices that result in inconsistent data and data models at various levels of manufacturing processes. The work presented in this article addresses this problem by envisioning a connective framework to support the engineering of CPPS through the use of a set of digital twins consistent with the real system throughout its lifecycle, not just used in the design and deployment phases. A review of the latest perspectives on using digital integration frameworks, methods, and solutions for lifecycle engineering of CPPS is provided in this article. This article demonstrates how a suitable framework, named SIMPLE, can be realized to effectively address the lack of consistent data models throughout the engineering lifecycle, including implementation details and example cases developed by the authors at

the Warwick Manufacturing Group (WMG) in selected industrial sectors. Consideration is given to supporting cyber-to-physical systems' connectivity and extendable engineering toolsets, forming the basis for multidisciplinary digital engineering environments. Key discussion points include the role and importance of effective integration of IT and OT, suitable frameworks for integration and collaboration.

KEYWORDS | Cyber–physical production system (CPPS); digital integration; digital twins; Industry 4.0; smart manufacturing.

I. INTRODUCTION

The potential role and benefits of implementing cyber–physical production system (CPPS) as part of smart manufacturing are now well recognized and include optimization, improved efficiency, predictive maintenance, improved collaboration, and information mobility. Assets can be optimized and used to their maximum potential. The linkage of a system-of-systems within a suitable ecosystem can bridge the gaps between data systems, control systems, and physical systems, reducing the time lags in the production processes. Predictive maintenance enables issues to be addressed before failures occur, cutting downtime and avoiding machine malfunctions. Improved collaboration is enabled through more dynamic workflows throughout the lifecycle and across engineering teams. Information mobility in this context is the ability to access data from any system in role-specific perspectives. This can improve productivity and creativity across job roles enabling better decision-making and the swifter provision of solutions. There is a clear consensus that production systems that implement smart manufacturing

and are interconnected are more efficient, productive, and intelligent than their unconnected equivalents [1]–[3].

CPPS consists of autonomous and cooperative elements and subsystems that are connected based on the context within and across all levels of production, from processes through machines up to production and logistics networks [4].

This article explores the need for, and application of, information and communication technology (ICT)-enabled integration frameworks to support the complete lifecycle of CPPS. At the relevant lifecycle engineering phases, different tools need to be coherently used in an integrated manner in support of relevant digital twins. Data need to be appropriately structured, shared, and accessed. Relevant middleware needs to be provided supporting appropriate messaging patterns and layered on appropriate communication to the sensing and actuating systems in the physical world. Such a framework needs to accommodate distributed applications (e.g., engineering, analytics, and decision-support tools), digital twins deployed to support the systems operational phases, and the physical system components. Often, these will be legacy components and tools with disparate interfaces and data formats.

II. BACKGROUND
A. Legacy and IT–OT Integration Challenges

While the promise and potential of smart manufacturing are significant, the realization of this potential is often limited by the inability to effectively integrate systems to obtain good quality data and to be able to adapt and manage such systems through their engineering lifecycle. This is further compounded by the disparate origins of information technology (IT) and operational technology (OT) systems, often referred to as the IT–OT divide. For example, if effective digital twins are to be constructed, information must be gleaned from multiple sources of data, e.g., production machines, real-time IoT sensors, historical sensor data, traditional manufacturing execution systems (MESs), enterprise resource planning (ERP), product/process lifecycle management (PLM) systems, and human input from domain and industrial experts. There is a failure to practically realize suitable frameworks to effectively bridge the gap between IT and OT systems, there is significant fragmentation of solutions, connectivity is poor, and the evolution of such systems is problematic [5].

The design of OT and IT systems has traditionally met specific requirements in order to serve distinctly different enterprise functions and user bases. These differences in technology, organizational culture, and function created a gulf between the OT and IT environments, creating barriers to capitalizing on the potential benefits of OT–IT convergence. However, for the paradigm of the smart factory to be fully realized, such systems need to be implicitly integrated [6].

Ciavotta *et al.* [7] report that, despite the success of IoT witnessed in the last decade, the adoption of IoT/CPPS deployments in manufacturing remains limited for a number of reasons, including a lack of suitable standards and recognized interoperability. Furthermore, security and real-time management issues also lead to current deployments being implemented in an *ad hoc* fashion and, often, limited to unidirectional data collection from the shop floor for monitoring [7]. There is typically poor lifecycle support, little reuse, fragmented connectivity, and a lack of engineering tool integration as vendor-specific partial solutions predominate.

Promising developments have seen a convergence of the technologies used to implement IT and OT systems and many of the software and engineering methods [8]. This article looks from the systems' integration perspective toward the realization of such converged systems.

B. CPPS and Emerging Digital Twins

Cyber–physical systems (CPS) are distributed, heterogeneous systems connected via networks and are usually associated with the concept of the IoT [9], whereas CPPS mechatronic components are coupled via networks to computational entities that enable production systems to adapt when changes occur throughout their lifecycles [10]. CPPS is formed by the integration of physical and digital systems across all levels of manufacturing enterprises that collaborate to form intelligent and responsive production systems [11].

The engineering of CPPS requires cross-disciplinary collaboration, which often results in inconsistencies among models and unintentional errors in data that can lead to failures at the deployment and operational phases [12]. As described by Colombo *et al.* [13], the new paradigms for implementing CPPS, such as service-oriented architecture (SOA), cloud computing, IoT, big data, and the industrial Internet, need to be deeply investigated, especially in real-world operations [13].

A closely associated and now well-established concept is that of digital twins; indeed, within the business community, the metaphor of a "digital twin" is gaining popularity as a way to explain the potential of IoT-based assets and smart environments [14]. Virtual representations, referred to as digital twins, are considered as a key enabler of engineering CPPS that can help in significantly reducing the complexity of engineering heterogeneous systems.

Digital twins are an emerging technology, which allows a systematic design, build, test, and operate approach that has significant potential to assist in system validation and prediction and making informed decisions. To develop high-fidelity digital twins multiple virtual models (models of products, processes, and resources), physical systems and manufacturing IT systems are required to be connected to form a network of data-sharing entities. Through such integration, the virtual models can be calibrated in synchronization with the related physical entity, while the physical entity can be dynamically optimized and adjusted based on the insights gained from the intelligence/analytics and simulation capabilities of virtual models. However, achieving integration of such data-intensive

Table 1 Example Digital Twin Features and Use Cases. Source: IIC Journal [16]

	Plan	Build	Operate	Maintain
Document management	PLM	PLM	Operation instructions	Service record
Model	Physical properties predict		Optimization	Diagnostics
Simulation	Design simulation	Virtual commissioning		
3D representation	Design drawings	Manufacturing instructions		Service instructions
Data model	Engineering data	Production data	Operational data	Service data
Visualization			Operational state display	health status display
Model synchronization			Real-time movement	Model inversion
Connected analytics			Operational KPIs	Asset health KPIs

networked objects is considered as one of the major challenges. The integration involves many layers of technology to enable data acquisition, communication, storage, and processing.

Digital twins provide digital representations of the physical system (in complex or selective forms) that updates and changes as the physical-twin system changes. Progressing through the phases of the manufacturing system lifecycle, the digital or physical systems may alternately be the sources, and sinks of data as the system are, for example, being defined driven by simulation or later when the digital model of the system is calibrated based on the execution of the physical system [15]. A complex interchange of data between the participants in this system of systems needs to be supported, which we will later allude to in Section III.

In 2019, the Industrial Internet Consortium (IIC) published an article on digital twin architecture and standards, proposing six sets of operations to characterize digital twin interactions within the Industrial IoT ecosystem; namely, they are discoverable, support underlying data repositories, and support event notification, the digital twin contents can be securely synchronized, and user authentication is supported. An integrated information model, separated from those representing each digital twin, forms the basis for all interactions, including design, orchestration, execution, and administration. This document provided a useful summary of digital twin features and use cases in an example lifecycle context [16] (see Table 1).

The ISO 23247 project initiated in 2018 has the objective of creating a digital twin manufacturing framework [17]. The framework is composed of a set of general principles, a reference architecture, digital representations of manufacturing elements, and the identification of relevant technologies for synchronization, exchange, and management of digitally represented manufacturing twins.

Various architectures that are suitable for the realization of digital twin use cases have been conceptualized.

Talkhestani *et al.* [18] suggest that a digital twin requires three main characteristics: synchronization with the real asset, active data acquisition from the real environment, and the ability of simulation. During the operation phase within the lifecycle of a CPPS, any occurring changes in the physical system should be fed into the digital twin so that it is always synchronized to the current state of the CPPS [18].

Maintaining such models throughout the system lifecycle and ensuring consistency of data between the various applications is a significant challenge. From this, stems the need for the automatic detection of change, the management of interdisciplinary dependencies, and consistency checking. Various approaches have been proposed to support this change management [18], [19]. A vision for an event mechanism to accommodate such change notification is presented within the integration framework discussed in this article, and the authors extend this concept throughout the lifecycle, for example, in support of performance prediction ahead of system deployment, for analytics-based optimization during the operational phase and for what-if analysis during reconfiguration to accommodate upscaling of production (see Section IV).

C. Integration Frameworks and Architectures

The increasing heterogeneity and complexity of manufacturing systems have highlighted the limitations of classical architectures. The smart manufacturing applications require a migration from the traditional ISA-95 layered architecture to a more flexible and interoperable architecture [20]. An approach is needed to support the evolution of the monolithic automation pyramid into flexible architectures maintaining support for existing legacy systems with an enhanced integration strategy. Fig. 1 depicts the evolution of data-driven/smart manufacturing organizations' architecture. It highlights the importance of designing solutions that contribute to integrate physical systems, virtual/digital systems through the use of robust data models, which is the purpose of the Smart InforMation PLatform and Ecosystem for manufacturing (SIMPLE) platform [21].

The concept of services enables the information exchange and interaction between any elements of the hierarchical levels of the industrial process. Several works developed and evaluated SOA in industrial applications [22], which included the SOCRADES, IMC-AESOP, GRACE, and ARUM European research projects. There is now growing interest in microservices, an SOA variant, and an architectural style in which the applications are decomposed into simple services by offering modularity, making applications easier to develop, test, deploy, and, most importantly, to change and maintain [23].

A number of authors have suggested frameworks that might be utilized to accommodate the management of industrial data within the context of smart manufacturing. Several standards bodies, industrial consortia, and
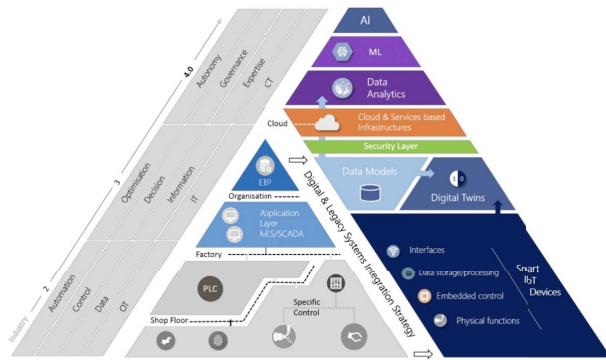
Fig. 1. *Evolution of the automation pyramid: authors' original work. Emphases on the importance of adopting a systematic approach to deploying data-driven manufacturing capabilities.*

research groups have been working in the field of architectures and frameworks for Industry 4.0 [19]. These provide comprehensive implementation-independent guidelines. Notable examples include RAMI 4.0 [24], IIRA [25], and 5C [9].

Trunzer *et al.* [19] consider how these approaches might be consolidated into a single system architecture looking at selected implemented use cases in a number of notable projects, including IMPROVE, PERFoRM, and BaSys4.0. They consider the perspectives of architecture, middleware, interoperability, and reconfigurability [19].

Ciavotta *et al.* [7] describe an architecture that is used to connect different simulation tools to describe a system model [7]. Each tool usually describes different facets of the system and their level of detail differs. Brandstetter and Wehrstedt [26] explain a related cosimulation framework to couple simulation models of different engineering domains and simulation tools to save modeling effort and analyze the system's behavior and the interaction of system components within the CPPS virtually [26].

As noted by Soldatos [27], the Industrial Internet-of-Things (IIoT) systems also provide the means for interconnecting legacy machines with IT systems and ultimately treating them as CPPS systems. This is mainly achieved through the augmentation of physical devices with middleware that implements popular IoT protocols, such as Message Queuing Telemetry Transport (MQTT), Open Platform Communications Unified Architecture (OPC UA), and WebSocket. Overall, CPPS and IIoT systems will be at the very core of all Industry 4.0 deployments in the years to come [27].

Saqlain *et al.* [28] report on experimental results from a smart factory case study that demonstrates that a framework can manage the regular data and urgent events generated from various factory devices in the distributed industrial environment through state-of-the-art communication protocols. The collected data are converted into useful information, which improves productivity and the prognosis of production lines [28]. Their proposed

framework contains five basic layers, physical, network, middleware, database, and application layers, to provide a service-oriented architecture for the end users.

The MAYA project introduced a distributed architecture to support different distributed virtual components. The main elements of this approach are as follows: a centralized support infrastructure, a simulation framework, and a communications layer. The project targeted the design, engineering, and management of CPPS systems during all the phases of the lifecycle [29].

Nguyen and Dugenske [30] proposed MQTT-based flexible architecture for manufacturing IoT using the publish and subscribe mechanism. The approach is aimed at connecting machines and applications requiring support for multiple protocols.

A clear goal within the realization of smart manufacturing is greater autonomy within more distributed architectures and frameworks. Through such autonomy, edge devices can operate independently of a central system making local decisions. Edge computing also simplifies the communication chain and reduces potential sources of error by connecting to physical assets directly and collecting, analyzing, and processing data directly. Edge devices can also directly execute operations, such as filtering and aggregating raw data, significantly reducing the need to transport a large amount of raw data to the cloud for further analysis [1], [2], [31], [32].

The research work and resulting platforms reviewed above mostly focus on the engineering of CPPS (architectures and deployment methods) and achieving connectivity with the resources and/or assets in the physical layer essential for collecting operational data during the systems' operation. There are also a number of articles published on unifying data formats and structures (e.g., AutomationML). However, no or minimum attention is given to the provision of a framework ensuring data consistency across various levels of manufacturing operations that can be shared with various engineering tools and components of CPPS. To address this gap, the key objective of the work presented in this article is the provision of a generic connective framework to achieve a tight coupling between the connectivity functions of the platform and a prescriptive processes, products, and resources (PPR) manufacturing data model. This ensures consistency between the operational data and the engineering data sets and digital models used to support the engineering phases, guaranteeing consistency between digital (or cyber) and physical systems, and also enables a seamless transition between the engineering and operational phases of CPPS lifecycle (see Fig. 2).

D. Future Vision

The aim of this article is to give an insight into the SIMPLE connectivity platform and its role in supporting configurable systems that can be progressively engineered throughout their lifecycle, drawing on appropriate standards and methods. Sections II-A–II-C of this article have
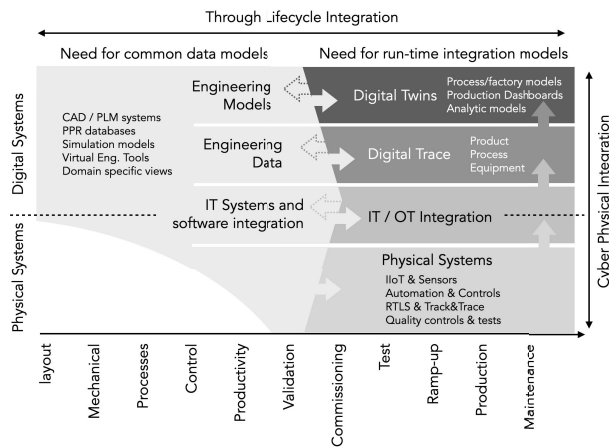
Fig. 2. *System integration requirements to ensure the transition from engineering to operational phases.*

described some of the challenges and proposed solutions related to the realization of effective frameworks for smart manufacturing systems integration. The need to support legacy integration and the continuing, if narrowing, divide between IT and OT systems has been highlighted. The emergence of digital twins in the context of CPPS has been reviewed, and the related standardization activities are briefly described together with relevant research. The practical realization and utilization of such smart manufacturing systems also require effective engineering methods and tools to support both their use and continuous evolution [33].

A series of projects at the University of Warwick and previous research by the Automation Systems Group (ASG), Loughborough University, have established lifecycle engineering, integration, and connectivity methods founded on the realization of a common data model shared between engineering applications throughout the lifecycle of manufacturing automation systems [34]. On-going research by the ASG has seen this approach evolve via the current SIMPLE research project into the concept of a shared dataspace, which can be populated and accessed throughout the engineering lifecycle to enable the integration of the physical system with its digital representation(s).

Section III describes the SIMPLE connective framework to functionally integrate the cyber and physical elements of manufacturing systems throughout their lifecycle and supporting the practical realization of digital twins via an open, configurable framework. In order to be successful, such an approach must be able to integrate disparate data sources effectively, understanding the context of their use at various lifecycle phases from the perspectives of both the physical system and related digital twins in order to gain insight into, and optimize, its operation.

The emphasis, and a key research contribution of SIMPLE, is the creation of an efficient, scalable, connective framework of minimum necessary complexity while fully contextualizing and cross-referencing data. The approach utilizes an efficient publish and subscribe integration layer for the real-time integration of digital twins with physical systems.

III. SIMPLE FRAMEWORK: MANAGING CPPS LIFECYCLE

The transition from engineering to operational phase marks the deployment of physical equipment on the shop floor and a significant shift in operational requirements across the manufacturing organizations. The RAMI 4.0 Architecture [35] and Industry 4.0 paradigm promote a holistic approach to the design and implementation of CPPS. In particular, the "Lifecycle and Value Stream" dimension of the IEC62980 RAMI 4.0 model highlights the need to consider the evolution of operational requirements throughout the complete manufacturing systems' lifecycle. Fig. 2 illustrates the approach used to identify key operational aspects of a digital twin framework throughout the CPPS lifecycle.

Existing engineering practices often result in a lack of continuity and consistency between the engineering and operational phases of a production systems' lifecycle. During the operational phase, a further dissociation exists between physical systems and the set of digital representations of those systems. The objectives of the SIMPLE framework are to facilitate: 1) the transition and maintain coupling between the engineering phase and the operational phase (through lifecycle integration) and 2) connectivity between digital systems and physical systems during their operations (CPPS integration).

A. Engineering and Operational Data Models

Data models (structure, content, and formats) that are used to collect, store, and manage the physical systems' digital trace (i.e., events and data generated by physical systems in operation) are not directly derived from or consistent with data models developed and used in engineering phases. This results in the building up of large operational data sets that cannot be related directly to the engineering data. This cleavage prevents operational data or digital trace from being efficiently used to enrich and refine engineering information throughout design iterations. It also indirectly contributes to the undocumented creep of engineering models as changes to the physical systems are made throughout the operational phases. Furthermore, the incompatibility between engineering and operational data models prevents the development of data-driven systems common to both the physical and digital systems (e.g., data views and information visualization, key performance indicators (KPIs) definition and representation, and analytics). The SIMPLE framework implements a skeleton PPR-centric data model that emphasizes the contextual information and metadata content required to provide connective capability between engineering and production data sets.

B. IT Systems Across Lifecycle Phases

IT systems deployed to support engineering and operational phases are different in nature. The storage and management of engineering data rely on complex relational data models and large centralized databases and data management systems (DBMSs) or engineering data warehouses (e.g., PLM solutions and PPR data hubs). The deployment of these systems mostly relies on direct software and database or client/server communication architectures.

Conversely, the operational phase is supported by OT systems (e.g., real-time industrial control networks, controls nodes, production, and orchestration control systems) and IT-level data systems centered around the management of streams of data generated from physical systems in real time, such as time-series/historian databases, and publish/subscribe, and broker-based communication platforms. It should also be noted that the data generated by physical systems are often collected and managed by either edge-level systems (e.g., supervisory control and data acquisition (SCADA) and local data servers deployed on the shop floor) or cloud-based data-lakes and/or industrial IoT platforms (e.g., MindSphere, Predix, and ThingWorx). In contrast to engineering data, operational data also tend to be stored in partially unstructured and often noncontextualized forms and/or with no reference to the engineering data set.

An effective digital twin framework requires interfaces to all the systems described above. The integration between the IT and OT domains is an essential element in the implementation of connected factories, digital twins, and CPPS, as it supports the communication of real-time signals, events and data from various production, IT, and cloud-level systems (e.g., SCADA, MES, ERP, and IoT platforms).

C. SIMPLE Framework

The objectives of the SIMPLE project being conducted by the ASG at WMG, University of Warwick, are to define a connective framework and implement the associated software components in order to provide practical and functional solutions to the challenges highlighted in Sections II and III: 1) facilitating the flow of data and information between engineering and operational phases of CPPS lifecycle and 2) facilitating the flows of data and information between digital models (cyber) and physical systems.

The SIMPLE platform development is partially funded by Innovate UK Manufacturing Made Smarter, Industrial Strategy Challenge Fund (ISCF) round 1 program, and its objective is to stimulate the development of manufacturing-specific but cross-sector and cross-industry digital capabilities. As such, the SIMPLE framework specification and the SIMPLE platform implementation focus on core capabilities that can be used as-is or adapted effectively to a large set of use case applications, organization sizes, and IT systems. The SIMPLE platform implementation targets a low complexity, low overhead (in terms of implementation and deployment time, skills, resources, and technologies), manageable, and scalable platform. It should be noted that the data transport and communication architecture of the SIMPLE platform is aimed at supporting logging and communication of state and status change information to enable soft-real-time synchronization between physical equipment (e.g., controllers and automation components) and their digital counterparts (e.g., digital models, production information, and management tools).

The term connective is used instead of integrative or integration (the act of combining into an integral whole) to place emphasis on the project's aim, which is to implement a framework that does not inherit the complexity of the systems that it integrates. Software systems integration often results in highly complex integration components or integrated engineering solutions, whose complexity increases exponentially with the number and/or complexity of systems. Alternative approaches focus on defining common data and/or functional models that capture all aspects of the systems between which integration is required [36]. While such approaches potentially allow a reduction in the software integration overhead as common data models are used for information exchange between systems, they often result in either excessively complex data structures and repositories, or narrow and domain-specific solutions.

The following design guidelines were defined to guide the SIMPLE framework functional definition, design, and implementation. The guidelines focus on reducing the overall complexity of the platform (1 and 6), enabling digital/virtual connectivity (2, 4, and 5), and accommodating IT implementation constraints (3).

1) The information required to cross-reference—engineering data, digital models, and physical systems—should be defined by a set of data models focusing on contextual and metadata information.
2) The synchronization at the real time of multiple digital models (i.e., composite digital twins), and of digital twins with the physical systems, can and should be supported by specific events and messaging models.
3) Client/server and web-service-based architectures and request/response communication models are not suitable for real-time integration of digital twins and physical systems. Publish/subscribe models should be favored in order to ensure the deployability and scalability of digital twin capabilities.
4) A digital twin platform should implement connectivity to both the OT and IT layers. As such, digital twin platforms should be implemented as part of the IT and OT layers and should promote the implementation of connectivity (protocol translation), communication/events, and information processing and transformation, as close to the edge as possible.
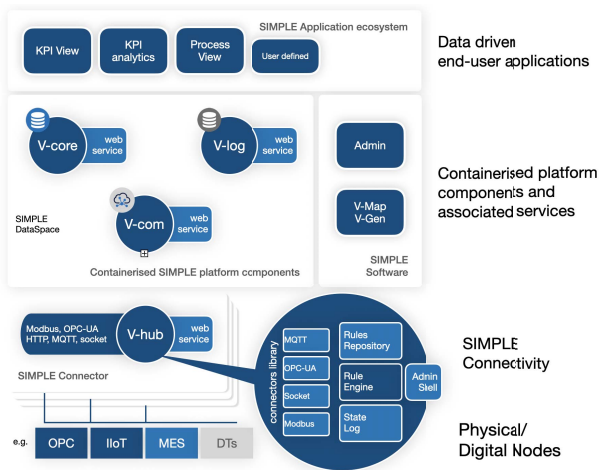
Fig. 3. *Overview of SIMPLE platform architecture and core software components' stack.*

5) A digital twin platform should promote the connectivity of physical systems with existing digital models and the connectivity between existing digital models in order to avoid multiplication of models and duplication of data.

6) In order to achieve connectivity between multiple software solutions and physical systems, low-level but complete, contextualized and cross-referenced data sets yield more value than detailed but fragmented and incomplete data sets.

Details of the SIMPLE software platform functional specification and implementation are provided in Section IV.

IV. SIMPLE PLATFORM IMPLEMENTATION

The SIMPLE platform implements several software components whose functionalities are related to: 1) simple connective PPR-centric relational data models as links between engineering and operational data (V-core); 2) simple event models and logging of real-time operational data (V-log); 3) connectivity component to OT and the organization IT or cloud layers supporting both protocol translation and edge data processing capabilities (V-hub); and 4) a publish/subscribe (MQTT-based) communication platform (V-com) (see Fig. 3 for details).

The core SIMPLE platform components are containerized (Docker implementation) to enable rapid and consistent deployment on a variety of IT platforms. Unlike typical IoT or IIoT platforms, some core components of the SIMPLE platform (i.e., V-hub) are designed and implemented to enable deployment on the edge, at the organization IT level, or in the cloud. The platform also implements software applications to support the configuration, deployment, and administration of the platform components (Admin). Future development phases will include the development of an application ecosystem that is not discussed or described further in this document.

Sections III-A–III-E provide information on each of the SIMPLE platform's components and applications.

A. Generic Process Model

Fig. 4 illustrates the approach underlying the design of SIMPLE data and event models. Key requirements of the SIMPLE platforms are to: 1) provide concise models to capture relationships between PPR data; 2) capture events from both physical and digital systems; and 3) enable synchronization of digital models describing manufacturing systems at various levels of hierarchy (e.g., factory, line, cell, and components levels).

In the generic example shown in Fig. 4, specific processes are mapped to elements of a resource hierarchy (see V-core implementation details in Section IV-B), and process-related events (e.g., resource status change) are logged by the V-log component (see details in Section IV-C). The concise model was used as a basis to develop the SIMPLE platform functionalities as it enables essential production KPI calculation and production performance analysis. It also allows information defined at various levels of details to be mapped; in Fig. 4, even if not explicitly defined (white color process bar), the assembly station process can be inferred from the component-level pick-&-place events and the process/resource relationships, which would allow a line-level discrete event simulation (DES) model to be synchronized with a kinematics level simulation for instance.

B. V-Core PPR Data Models

The V-core component implements a relational, PPR centric skeleton data model that focuses on capturing the structure of physical resources and products and their relationship to specific processes. The V-core data model aims at providing a generic model applicable to a wide range of applications. The model was used to implement PPR modeling for the construction industry (i.e., structural insulated panel assembly), battery, and seat manufacturing for the automotive industry and is generally applicable to most discrete manufacturing applications.
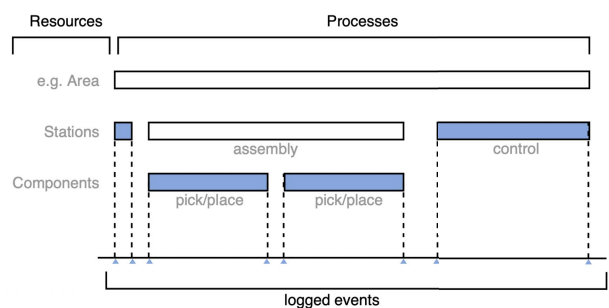

Fig. 4. *Design of SIMPLE process and resource-related data and event models.*
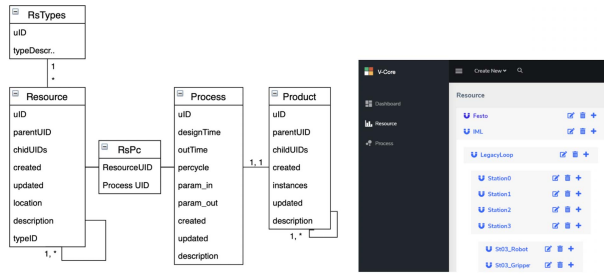
Fig. 5 provides details of the key tables describing the V-core relational data model. Tables related to the V-core component configurations and administration (e.g., users and right management, versioning, and change management) have been omitted. A screen grab of the V-core database UI that can be used to manually edit the PPR information is also included in Fig. 5.

V-core implements a REpresentational State Transfer (REST) over hypertext transfer protocol (HTTP) application programming interface (API) that allows external applications to retrieve, populate, or update the database content using JavaScript Object Notation (JSON) formatted payload. V-core also implements an MQTT interface and can both publish and subscribe to the V-com broker component (see section IV-D); for instance, if changes are made to the PPR data using the V-core REST API (e.g., addition/deletion of resources or remapping of PPR relationships by an external application), V-core will publish an event to inform other SIMPLE platform components or external subscriber applications. As a subscriber, V-core can also receive PPR-related change events and update the PPR information accordingly. Examples of practical use cases of the V-core data model, including its integration with the vueOne virtual engineering solution and integrated manufacturing and logistics (IML) battery module assembly line at WMG, are provided in Section V.

C. V-Log and SIMPLE Event Model

The V-log component is a containerized time-series database, currently implemented using Influx DB. V-core implements both: 1) an MQTT interface (to V-com; see Section IV-D) to publish and subscribe to specific events defined by the SIMPLE framework) and 2) a REST API that allows retrieval of logged events by SIMPLE components or third-party applications.

The SIMPLE event and messaging model define two types of events that are logged by the V-log component (see Fig. 6): 1) changes of content and/or PPR relationships in the V-core data, which allows complete traceability of system design and configuration changes made throughout a system lifecycle and 2) change of resources' status published by physical or digital systems at real time. The

status is defined by the source node itself. Examples of status types for a manufacturing asset are active/inactive, idle, blocked/starved, in fault, and so on.

All messages contain a reference to the publisher node (unique node ID). A node is defined as any physical (e.g., IoT devices, edge, and OPC UA server) or digital assets (e.g., simulation models/environments) that connect and publish to the SIMPLE platform's V-com broker via a V-hub instance. Events are expected to be time-stamped by the source node and, if not, are time-stamped by the V-com broker. Fig. 6 provides the message structure for data and status-related change events. Other events relate to the SIMPLE communication platform management (e.g., node connection/disconnection, and MQTT Last Will and Testament) and are not detailed in this document.

The messages related to resources' status change events include a reference to the resource itself and a reference to one of the processes associated with it in the V-core database. The payload field can be used to pass additional data and information in any format and structure (e.g., plain, JSON, or eXtensible Markup Language (XML) formatted string, numerical value, and binary object) that the subscriber node can interpret. For instance, the payload field can be used to return an image after an inspection process is complete, the results of a measurement, logs generated by controllers, and so on.

D. V-Com Publish-Subscribe Broker

The V-com component is an MQTT publish/subscribe message broker built on top of the Mosquito MQTT platform. V-com is configured by default to support MQTT level 2 Quality of Service (QoS) (a four-step handshake guarantees delivery of messages exactly once) as the additional communication overhead is acceptable given the purpose of the SIMPLE platform. V-com is configured by default to retain messages for all topics, which allows new subscriber nodes to obtain all past messages.

The V-com component implements two main Topics (a term used to describe MQTT message hierarchy): 1) for messages related to changes in the V-core databases and
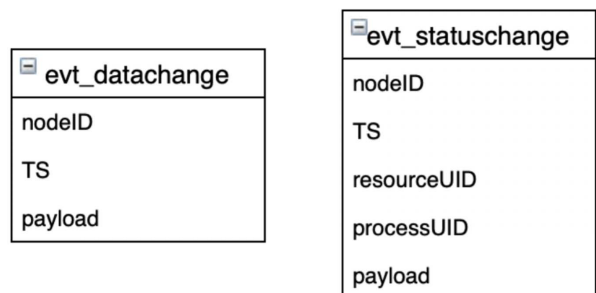

Fig. 6. *V-log related event defined by SIMPLE, allowing change notification at design time (data change) and real time (status change).*

2) messages related to real time communication with live physical or digital systems (i.e., data change and status change events). Subtopics are defined using the unique ID of resources as defined in the V-core database (see Section IV-B). The list of topics can be retrieved by new publisher/subscriber nodes using V-core REST API.

The V-com implements a message content and structure validation procedure that ensures that all messages are formatted according to the SIMPLE event and messaging model (see above section). Any messages whose content or structure is not compliant with the V-log event model are discarded in order to preserve the consistency of information within the SIMPLE defined namespace.

E. V-Hub Connectors and Related Software Components

V-hub is the component that can be configured to achieve real-time connectivity with systems and applications deployed in the IT and IT/OT levels of manufacturing organizations (e.g., MES, engineering databases and web servers, modeling and simulation environment, and OPC UA servers). V-hub instances support two key functions:

1) Protocol Translation: From communication protocols used by external systems to the MQTT-based SIMPLE communication space, the library of connectors that V-hub instances can currently implement are HTTP client, socket/web sockets, OPC UA client, Modbus, and MQTT, which will be extended based on emerging use case requirements.

2) Data Transformation: Data transformation is supported by the rule engine implemented as part of each deployed V-hub instance. Rule engines can be programed to apply specific data processing and transformation. The primary objective of data transformation is to generate an output message whose structure and information content complies with the SIMPLE V-log messaging format (see Section IV-C). In addition, the information contained in the input message can be processed (e.g., numerical calculations, string processing, and formats' translation).

A typical example of a rule implemented by V-hub instances aims at the mapping of OPC UA channels and tags to a resources status change as defined by the V-log StatusChange messaging model (e.g., if tagA is True and tagB is False, then output string array ["OPC_serverA," timestamp, "Gripper_station1," "," "status=InFault," and "fault_message"]). The rule engine implements an event log that allows asynchronous input messages to be processed within the same rule. The latest message from a given source is logged; then, the rule engine parses the rules library and executes the rules that contain references to the input node.

The design and implementation of the SIMPLE platform connectivity components differ from typical IoT platform implementations; IoT platforms (e.g., Siemens MindSphere, GE Predix, and Fujitsu RICE), typically implement both protocol translation and data transformation as a centralized functionality deployed on cloud/server systems (e.g., NodeRed-based platform such as Siemens MindSphere). However, such approaches result in high server load and bandwidth utilization across networks and can cause interruption of operations and loss of data if connectivity is lost. Industrial edge solutions are emerging (e.g., IgnitionEdge and Fujitsu IntelliEdge) that provide both hardware and software for edge-level connectivity, protocols translation, and data transformation, as well as local caching of messages and data. The SIMPLE platform promotes a similar approach (i.e., deployment or protocol translation and data transformation as close to IT/OT edge as possible) but differs in two fundamental aspects:

1) The SIMPLE implementation reinforces a specific information model (messages information content and structure) aligned to the V-core data model. This allows us to ensure that all information communicated within the SIMPLE namespace is contextualized and consistent with data stored in V-core.

2) Every instance of the V-hub connector is deployed as a stand-alone self-contained service on the targeted device, resulting in fully distributed protocol translation and data transformation capabilities. A single V-hub component can implement connectivity to one or more nodes and implement one or more rules, and one or more V-hub services can be deployed on a single device. It should also be noted that V-hub services can be deployed on server systems if required. The V-hub component is implemented using Python and V-hub connectors using available open-source python libraries. V-hub can, therefore, be deployed on any platform that can run a Python interpreter. Those combined capabilities provide a significant level of flexibility in structuring and deploying connectivity across a variety of IT and OT system architectures and configurations.

A software application that implements two functional modules (V-map and V-gen) has been developed to support the configuration and deployment of the V-hub component's instances. The V-Map software is used to: 1) select and configure the connectors (e.g., OPC UA client) required to achieve connectivity with IT and OT level nodes (e.g., OPC server, MES, and DES) and 2) implement rules to apply to incoming messages/events (i.e., data transformation). The current implementation of the rule editor is essentially a simple Python code development integrated development environment (IDE). However, future implementation will focus on the design and implementation of use cases or customer-specific rule editors with more refined capabilities (e.g., rules management library, rules template, and advanced UI/UX).

Once the required connectors have been selected and configured and the rules have been defined, the V-gen software environment is used to compile the connectors, rule
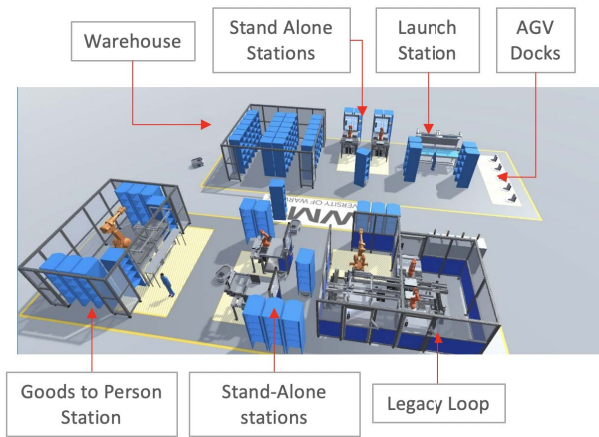
Fig. 7. *Layout of the IML demonstrator at WMG.*

engine, rule definition, and event logger code (into byte-Code), as well as a simple service shell that allows V-hub instance to be administered (e.g., retrieval of versioning information, retrieval of the event log, and live monitoring of input/output messages and events). The compiled code can then be deployed on the targeted devices and executed to support real-time operations. This approach mirrors the automatic PLC control code generation developed by ASG as a part of the vueOne virtual engineering solution [33].

V. USE CASES AND THEIR IMPLEMENTATION

A. Integrated Manufacturing and Logistic Demonstrator

A full-scale IML demonstrator installed in the Warwick Manufacturing Group (WMG) is used to demonstrate example use cases in this article (see Fig. 7). This system showcases Industry 4.0 methods and encompasses both new production systems and legacy equipment within a series of advanced manufacturing scenarios, which is being used for both research and training with a range of industrial partners. The IML is a dynamically adaptable modular and reconfigurable, and hence, the application can be progressively changed as new requirements emerge. Machine stations can be exchanged physically and also virtually, i.e., new virtual station models can be swapped in (and out) in place of physical stations. It is currently configured to carry out a battery submodule assembly demonstration as a part of an Innovate UK and HVM Catapult-funded project. The product assembly consists of 18 650 and 26 650 form-factor cylindrical cells to be assembled into submodules and modules incorporating bus bars and an integrated cooling system.

The IML features MES, three autonomous guided vehicles (AGVs), AGV fleet manager, control systems, and automation equipment from leading vendors, e.g., Siemens, Rockwell Automation, ABB, Mitsubishi, and Festo. It aims to provide a full-scale demonstrator for new manufacturing automation methods, tools, and technologies with the objective to support the entire lifecycle, e.g., enabling the digital validation, verification, and visualization, control code generation, and cloud-based engineering services. The demonstrator system has been implemented to support the combination of legacy and agile systems—stations connected through a traditional conveyor-based system, stand-alone stations, distributed warehousing, and AGV-based autonomous logistic system for pallet transportation and line-side component supply. The integration of intralogistics and assembly and the use of distributed warehousing offer the potential to minimize disruptions due to production abnormalities and reduce nonvalue-adding activities within adaptable processes and dynamic changes in product variety and volumes while maintaining efficiency.

This section provides use case examples to demonstrate how the SIMPLE platform is used in achieving integration between various software tools and physical components at the design, deployment, and production phases of the IML. The use cases provided illustrate how such connectivity allows: 1) improved virtual validation by integrating virtual engineering tools with MES and physical components, such as programmable logic controllers (PLCs); 2) improved accuracy of models by calibrating data models during production through integration of virtual models and physical systems; and 3) improved optimization of scheduling in (soft) real time by integrating DES tools with production system to resimulate and reschedule internal logistics in case of any abnormalities during production.

B. Digital-Digital Integration

Digital-digital integration is carried out using the PPR data model of V-Core (see Fig. 8). PPR data model-based integration not only allows reuse of information but also helps in enforcing version control and keeping the digital models up-to-date, thus eliminating discrepancies. At the design stage, two types of digital twins are developed (i.e., line level and station level) to design and simulate the IML.

The line-level model of the IML is developed in DES tool Witness that offers a detailed analysis of the overall line-level process and intralogistics. Data and real-time connectivity between the vueOne and Witness digital models are carried out using the SIMPLE platform.

Station-level models are developed in vueOne virtual process planning software. vueOne is a 3-D kinematic-level process planning tool that offers detailed level modeling and simulation capabilities for automatic, semiautomatic, and manual operations. Various types of sensors and actuators and manual operations can be realistically modeled and simulated. Details of vueOne capabilities are reported in [33]. Once models of stations are validated, the station-level information (sequence of operations, cycle time, details of machine components, and their physical interface mapping) is used to update the process definition in the V-core data repository, via V-core HTTP API.
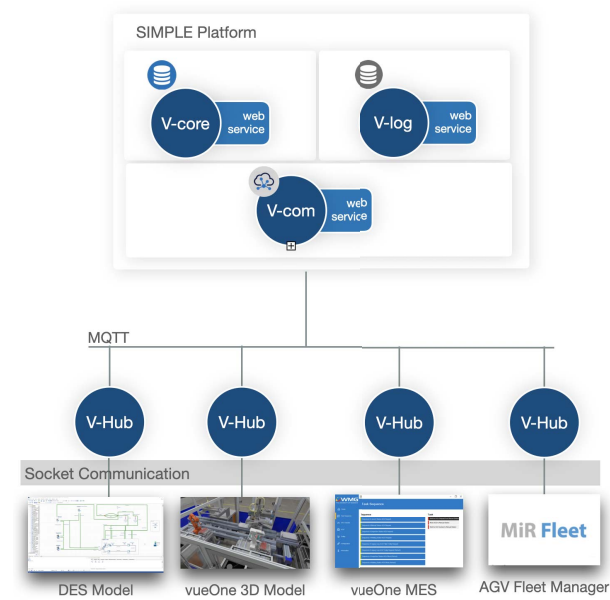
Fig. 8. *Overview of the digital-digital integration of the IML demonstrator digital models.*

V-core's data model holds a replica of the entire line-level information and its hierarchy (i.e., areas, zones, stations, systems, and components). Station-, system-, and component-level information of the IML is imported from vueOne, whereas area- and zone-level information is defined manually in the v-hub core.

Once area- and zone-level information is defined, the complete line-level information can then be retrieved by the DES model using V-core HTTP API interface. After carrying out the integration, both station- and line-level models can subscribe to changes in the data model and can be dynamically updated if an update is pushed from the V-core or any of the client sides.

The work is currently carried out to further extend V-core-based integration to enable the integration of MES and AGV Fleet Manager. This will significantly help in validating and optimizing the performance of the overall system before deployment.

C. Digital–Physical Integration

For CPPS engineering, it is vital to have digital–physical integration and have real-time data synchronization between virtual models, physical equipment, and manufacturing IT systems to closely align them over the operational phase. This results in new application areas for modeling and simulation technologies beyond the design phase. Example use cases are presented in the following.

1) Virtual Commissioning: During the design stage, the digital–physical integration is used to carry out virtual commissioning. Fig. 9 shows the virtual commissioning setup of a stand-alone welding station to validate control software in a virtual environment. The communication between the vueOne model and station PLC is

achieved either by using native communication drivers (e.g., S7comm) or through the V-hub component and OPC UA connector. The input and output signals of S71500 PLC are mapped to the respective sensor and actuator components of the virtual model. This integration could be further extended to include connectivity with MES to test control software in a more realistic environment, thereby avoiding making costly changes to the software afterward.

2) Data Model Calibration: During the production phase, V-com is used to collect the PPR data from the physical system in real time and stores it in V-log that is a time-series database (see Section IV-C). Data model calibration is performed once sufficient data are collected from the physical equipment (e.g., average cycle time). The calibrated data are then pushed to the vueOne and DES models to make the digital models in-line with the performance of the physical systems [e.g., process time data field in V-core's process table (see Section IV-B)].

3) Dynamic Optimization of AGV Fleet Management: In this use case, the SIMPLE platform is used to integrate MES, Fleet Manager and physical equipment with the DES model, and a service module Smart AGV Management System (SAMS) [37] to carry out research work in dynamically optimizing line side supply and pallet transportation in real time based on equipment status and production demand. SAMS carries out optimization with the help of mixed-integer nonlinear programming (MINLP) using genetic algorithm (GA) integrated based on both demand information, real-time resource status information, and DES simulation output. Details of SAMS are not in the scope of this article. Both historic and live data are provided to the SAMS module using V-log and V-com. The optimization study is performed using historic data to better understand the consequences of production
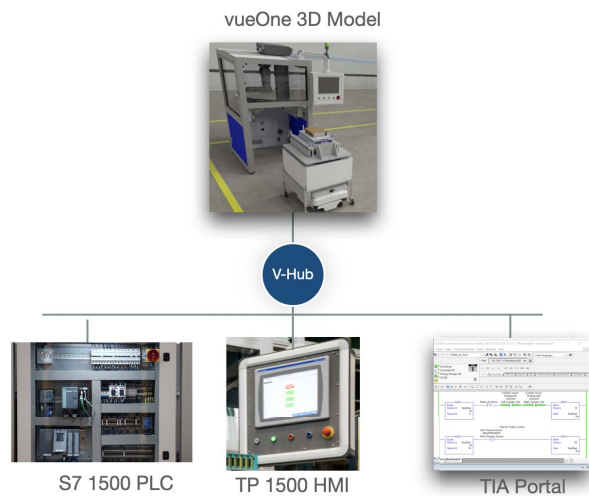

Fig. 9. *Setup for virtual commissioning of stand-alone welding station.*

abnormalities and optimizing the schedule to minimize the effects of abnormalities.

In the case of abnormality detection or change in production demand, simulation and analytics-based optimization is carried out, and rescheduling instructions are released to MES as a response action to optimize throughput. A consequent DES and AGV fleet management simulation are then performed to study the impact on the schedule. The impact is measured and reflected through KPIs, such as overall equipment effectiveness, and build to schedule. When implemented in real time, such an optimization approach will offer a step-change in the adaptability of manufacturing systems.

VI. CONCLUSION

The realization of a connective framework for CPPS has been presented, which aims to address legacy system and IT/OT integration challenges. SIMPLE is an efficient, scalable, connective framework for manufacturing systems, of minimum necessary complexity, while fully contextualizing and cross-referencing data. The approach proposed utilizes an efficient publish and subscribe connective platform for the real-time integration of digital twins with physical systems. Through the contents of this article, the authors have attempted to highlight the key and unique capabilities of the SIMPLE connectivity platform, which are the result of the tight and consistent integration between data/information collection and management capabilities (typically provided by IIoT platforms) and manufacturing centric PPR data models used throughout the data pipeline to ensure consistency and quality of the resulting data sets.

This article has highlighted the need to extend the ISA-95 manufacturing pyramid via an enhanced flexible integration strategy. The role of a services-oriented approach to integration is considered, with a growing trend toward the utilization of microservices in this context. The importance of digital twins, their characteristics, features, and example use cases is considered through a focused review of contributions and selected relevant research projects and initiatives.

The need to comprehensively support connectivity in both the engineering and operational phases and aspects of the smart manufacturing system is highlighted, including the event model and its support for change in the context of the evolution of manufacturing systems, both from a PPR perspective, and its real-time status traceability and synchronization.

A full-scale IML demonstrator installed at WMG is used to demonstrate example SIMPLE use cases related to digital-to-digital and digital-to-physical integrations. The SIMPLE connectivity platform aims at providing data-centric integration capabilities. As such, the evaluation of the benefits in terms of increased operational effectiveness (e.g., productivity) cannot be carried out through the measurement of direct production KPIs for instance. Instead, the improvement targeted by the presented research is to provide a systematic, prescriptive, and robust data collection platform to ensure that the data collected are consistent, complete, and contextualized. The improvements targeted are reduced data cleansing and curating time and effort, reducing to zero the collection of incomplete data sets and/or data sets with inconsistent data/information structure and formats. Complementary and further research phases will focus on comparatively assessing as-is and SIMPLE-based data workflow within an organization and evaluating differences in data processing steps and resource allocation (time-, manual-, and software-based data processing workflows) to obtain data sets of similar quality (content, structure, and format). Future development plans also include the integration of quality and completeness indicators as part of the SIMPLE platform administrative tools, which will inform users on the status and quality of their data collection relative to the benchmark defined by the prescriptive SIMPLE PPR data model.

Future development phases of the SIMPLE project will also focus on both refining existing functionalities and expanding the capabilities of the platform while remaining consistent with the vision of providing low complexity and highly maintainable and deployable solution. The development of out-of-the-box advanced analytics capabilities based on the V-com and V-log data content would add significant value to the SIMPLE platform as a production analysis platform. Similarly, dynamic advanced PPR data visualization and KPI dashboarding can be implemented using the information and events currently managed by the SIMPLE platform components.

The implementation and continuous development of the V-hub connector library are important aspects of platform development. The integration of new protocols and approach at both the OT and IT/OT layer (e.g., support for data distribution service (DDS), OPC PubSub, OPC time-sensitive networking (TSN), IEC 61499, and distributed control architectures) in the SIMPLE specification will provide new opportunities for real-time connectivity to OT-level systems. At the IT level, MQTT multibroker bridging will be investigated as such capability would directly impact the scalability of the platform, as well as its maintainability.

Full deployment and testing of the SIMPLE platform is envisaged at the WMG Energy Innovation Centre and the UK Battery Industrialisation Centre (UKBIC) battery manufacturing and testing facilities, with the objective of virtually validating production campaigns for a variety of product configurations (e.g., battery cells, packs, and modules). The UKBIC case study will provide an example use case in a state-of-the-art smart factory, where the deployment and use of digital twins will be critical to achieve demanding production and business objectives. ∎

REFERENCES

[1] D. Greenfield, "Fog computing vs. edge computing: What's the difference," *Automat. World*, 2016.

[2] J. Vater, L. Harscheidt, and A. Knoll, "A reference architecture based on edge and cloud computing for smart manufacturing," in *Proc. 28th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2019, pp. 1–7.

[3] F. Lopez *et al.*, "Categorization of anomalies in smart manufacturing systems to support the selection of detection mechanisms," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 1885–1892, Oct. 2017.

[4] L. Monostori *et al.*, "Cyber-physical systems in manufacturing," *CIRP Ann.*, vol. 65, no. 2, pp. 621–641, 2016.

[5] O. Givehchi, K. Landsdorf, P. Simoens, and A. W. Colombo, "Interoperability for industrial cyber-physical systems: An approach for legacy systems," *IEEE Trans. Ind. Informat.*, vol. 13, no. 6, pp. 3370–3378, Dec. 2017.

[6] F. Burzlaff and C. Bartelt, "A conceptual architecture for enabling future self-adaptive service systems," in *Proc. 52nd Hawaii Int. Conf. Syst. Sci.*, 2019, pp. 7472–7481.

[7] M. Ciavotta, M. Alge, S. Menato, D. Rovere, and P. Pedrazzoli, "A microservice-based middleware for the digital factory," *Procedia Manuf.*, vol. 11, pp. 931–938, Jan. 2017.

[8] Y. Lu, F. Riddick, and N. Ivezic, "The paradigm shift in smart manufacturing system architecture," in *Proc. IFIP Int. Conf. Adv. Prod. Manage. Syst.* Cham, Switzerland: Springer, 2016, pp. 767–776.

[9] E. A. Lee, "Cyber physical systems: Design challenges," in *Proc. 11th IEEE Int. Symp. Object Compon.-Oriented Real-Time Distrib. Comput. (ISORC)*, May 2008, pp. 363–369.

[10] L. Ribeiro, "Cyber-physical production systems' design challenges," in *Proc. IEEE 26th Int. Symp. Ind. Electron. (ISIE)*, Jun. 2017, pp. 1189–1194.

[11] M. Jbair, B. Ahmad, M. H. Ahmad, and R. Harrison, "Industrial cyber physical systems: A survey for control-engineering tools," in *Proc. IEEE Ind. Cyber-Phys. Syst. (ICPS)*, May 2018, pp. 270–276.

[12] B. Vogel-Heuser and M. Zou, "Leveraging inconsistency management in the multi-view collaborative modelling of cyber-physical production systems," *IET Collaborative Intell. Manuf.*, vol. 1, no. 4, pp. 126–129, Dec. 2019.

[13] A. W. Colombo, S. Karnouskos, O. Kaynak, Y. Shi, and S. Yin, "Industrial cyberphysical systems: A backbone of the fourth industrial revolution," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 6–16, Mar. 2017.

[14] B. Marr. *What is Digital Twin Technology-and Why is it So Important?* Accessed: Jul. 2, 2020. [Online]. Available: https://www.forbes.com/sites/bernardmarr/2017/03/06/what-is-digital-twintechnology-and-why-is-it-so-important/

[15] E. Curry, *Real-Time Linked Dataspaces: Enabling Data Ecosystems for Intelligent Systems*. Cham, Switzerland: Springer, 2020.

[16] K. Harper, S. Malakuti, and C. Ganz, "Digital twin architecture and standards," *IIC J. Innov.*, pp. 1–12, Nov. 2019. [Online]. Available: https://www.iiconsortium.org/news/joi-articles/2019-November-JoI-Digital-Twin-Architecture-and-Standards.pdf

[17] *Automation Systems and Integration—Digital Twin Framework for Manufacturing—Part 1: Overview and General Principles*. Geneva, Switzerland: ISO. Accessed: Jul. 2020. [Online]. Available: https://www.iso.org/standard/75066.html

[18] B. A. Talkhestani *et al.*, "An architecture of an intelligent digital twin in a cyber-physical production system," *Automatisierungstechnik*, vol. 67, no. 9, pp. 762–782, Sep. 2019.

[19] E. Trunzer *et al.*, "System architectures for industrie 4.0 applications," *Prod. Eng.*, vol. 13, nos. 3–4, pp. 247–257, 2019.

[20] O. Carlsson, J. Delsing, F. Arrigucci, A. W. Colombo, T. Bangemann, and P. Nappey, "Migration of industrial process control systems to service-oriented architectures," *Int. J. Comput. Integr. Manuf.*, vol. 31, no. 2, pp. 175–198, Feb. 2018.

[21] UKRI. *Manufacturing Made Smarter Winning Projects*. Accessed: Sep. 2020. [Online]. Available: https://www.ukri.org/news/147m-investment-in-mms/

[22] P. Leitão, A. W. Colombo, and S. Karnouskos, "Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges," *Comput. Ind.*, vol. 81, pp. 11–25, Sep. 2016.

[23] P. D. Francesco, I. Malavolta, and P. Lago, "Research on architecting microservices: Trends, focus, and potential for industrial adoption," in *Proc. IEEE Int. Conf. Softw. Archit. (ICSA)*, Apr. 2017, pp. 21–30.

[24] K. Schweichhart. *Reference Architectural Model Industrie 4.0 (RAMI 4.0), An Introduction*. Accessed: Sep. 15, 2020. [Online]. Available: https://ec.europa.eu/futurium/en/system/files/ged/a2-schweichhart-reference_architectural_model_industrie_4.0_rami_4.0.pdf

[25] Industrial Internet Consortium. *Industrial Internet Reference Architecture*. Accessed: Nov. 24, 2020. [Online]. Available: https://www.iiconsortium.org/IIRA.htm

[26] V. Brandstetter and J. C. Wehrstedt, "A framework for multidisciplinary simulation of cyber-physical production systems," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 809–814, 2018.

[27] J. Soldatos, "Introduction to industry 4.0 and the digital shopfloor vision," in *The Digital Shopfloor: Industrial Automation in the Industry* (Automation, Control and Robotics), J. Soldatos and O. Lazaro, Eds. Aalborg, Denmark: River Publishers, 2019.

[28] M. Saqlain, M. Piao, Y. Shim, and J. Y. Lee, "Framework of an IoT-based industrial data management for smart manufacturing," *J. Sensor Actuator Netw.*, vol. 8, no. 2, p. 25, Apr. 2019.

[29] S. Weyer and M. Keinan, "Classification of application scenarios for a virtual commissioning of CPS-based production plants into the reference architecture RAMI 4.0," presented at the VDI Automation-18: Seamless Converg. Automat. IT, 2018.

[30] V. Nguyen and A. Dugenske, "An Internet of Things for manufacturing (IoTfM) enterprise software architecture," vol. 2, no. 2, pp. 177–189, Oct. 2018.

[31] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proc. Workshop Mobile Big Data*, 2015, pp. 37–42.

[32] V. Gazis, A. Leonardi, K. Mathioudakis, K. Sasloglou, P. Kikiras, and R. Sudhaakar, "Components of fog computing in an industrial Internet of Things context," in *Proc. 12th Annu. IEEE Int. Conf. Sens., Commun., Netw. Workshops (SECON Workshops)*, Jun. 2015, pp. 1–6.

[33] R. Harrison, D. Vera, and B. Ahmad, "Engineering methods and tools for cyber-physical automation systems," *Proc. IEEE*, vol. 104, no. 5, pp. 973–985, Mar. 2016.

[34] R. Harrison, S. M. Lee, and A. West, "Component based distributed control systems for automotive manufacturing machinery developed under the foresight vehicle programme," *SAE Trans.*, vol. 111, no. 5, pp. 218–226, 2002.

[35] *Reference Architecture Model Industrie 4.0 (RAMI 4.0)*, Standard 91345:2016-04, DIN SPEC, Deutsches Institut für Normung (DIN), Berlin, Germany, 2016, doi: 10.31030/2436156.

[36] G. N. Schroeder, C. Steinmetz, C. E. Pereira, and D. B. Espindola, "Digital twin data modeling with AutomationML and a communication methodology for data exchange," *IFAC-PapersOnLine*, vol. 49, no. 30, pp. 12–17, 2016.

[37] F. Yao, B. Alkan, B. Ahmad, and R. Harrison, "Improving just-in-time delivery performance of IoT-enabled flexible manufacturing systems with AGV based material transportation," *Sensors*, vol. 20, no. 21, p. 6333, Nov. 2020.

ABOUT THE AUTHORS

Robert Harrison received the B.Tech. and Ph.D. degrees in mechanical and manufacturing engineering from Loughborough University, Loughborough, U.K., in 1981 and 1991, respectively.

He was a Research Fellow with the National University of Singapore, Singapore, from 1993 to 1995, where he was a Ford Research Fellow, from 1996 to 2002. Since 2005, his Automation Systems Group has been a partner on several large EU projects related to service-oriented distributed systems, including SOCRADES, RIMACS, Arrowhead, and IMC-AESOP. Formally the Director of Automation Systems at Loughborough University, he joined the University of Warwick, Coventry, U.K., in March 2013. From 2013 to 2018, he led the EPSRC Knowledge Driven Configurable Manufacturing Project. He is currently a Professor of Automation Systems and the Head of the Digital Technologies Directorate at WMG, University of Warwick. His current research interests include Industrial Strategy Challenge Fund (ISCF) Manufacturing Made Smarter SIMPLE, DIALOG, and DAMPO projects, EU DIGIMAN, and numerous contract research and HVM Catapult projects, with BAE Systems, Lear Corporation, Atlas Copco, and so on.

Dr. Harrison received the Royal Academy of Engineering Global Research Award in 2004 with visiting research roles at Schneider Electric and Ford.

Daniel A. Vera received the M.Sc. degree in mechanical and manufacturing engineering from ENIT, Tarbes, France, in 2000, and the Ph.D. degree in manufacturing engineering from Loughborough University, Loughborough, U.K., in 2004.

He is currently an Associate Professor with the Warwick Manufacturing Group (WMG), University of Warwick, Coventry, U.K. He has been working in the domain of manufacturing engineering for over 15 years. His research interests are focused on various aspects of manufacturing from the modeling, analysis, and optimization of engineering processes to the design and development of 3-D-based virtual engineering and collaboration tools for supporting the manufacturing system lifecycle. He has worked on numerous U.K. and European projects as a lead researcher. He is currently taking a leading role in the commercialization of newly developed engineering tools and services for digital manufacturing.

Bilal Ahmad (Senior Member, IEEE) received the M.Sc. degree in mechatronics and the Ph.D. degree in automation systems from Loughborough University, Loughborough, U.K., in 2007 and 2014, respectively.

He was a Research Fellow and a Senior Research Fellow with the Warwick Manufacturing Group (WMG), University of Warwick, Coventry, U.K., and a Research Associate at the Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University. He is currently an Associate Professor with the Warwick Manufacturing Group (WMG), University of Warwick. He has worked on a range of high-profile national and international research projects. He is named as PI and Co-I on a number of Innovate UK and HVM Catapult projects. He has published his research findings in over 50 peer-reviewed journal articles and conference papers. His research interests are in the areas of manufacturing digitalization and lifecycle engineering of cyber–physical production systems, especially focusing on the design and deployment of real-time control systems and their connectivity with IT systems.