Sustainable Production through Advanced Manufacturing, Intelligent Automation and Work Integrated Learning, J. Andersson et al. (Eds.)

doi:10.3233/ATDE240171

Exploring Production System Knowledge Graph Applications Using a Simulation Framework

Martin BIRTIC [a,1], Richard SENINGTON [a] and Anna SYBERFELDT [a]
[a] *School of Engineering Science, University of Skövde, Högskolevägen, Box 408 Skövde, 541 28, Sweden*

Abstract. Knowledge graphs are generating significant interest in industry and research. These graphs can be enriched with data to represent aspects of production systems such as their structure, component interrelationships, and conditions. This provides opportunities to gain insights into system behavior, performance, and states. Such insights could potentially be leveraged by a wide range of technologies for a multitude of purposes and applications such as system control, process optimization, and informed decision making. However, the existing literature addressing industrial applications of knowledge graphs related to production systems remains limited in scope and depth. This underscores the importance of developing methods for exploring the potential use and implementation of knowledge graphs in such systems. The primary focus of this study centers on facilitating such exploration by developing a virtual commissioning simulation framework. A modular production system is modelled that leverages physics, moving product dynamics, and incorporates authentic PLC and robot programs. A knowledge graph is integrated and enriched with data representing various aspects of the system. An application is developed to facilitate product routing and prioritization. A service-oriented approach is used that leverages graph data processing and exchange for service registration and matching. System simulations are conducted and subsequently the framework is evaluated for outcomes and findings. This study demonstrates the successful design and implementation of a production system simulation framework that uses knowledge graphs for system functionality. It demonstrates the exploration of knowledge graph applications through the development of a modular and service-oriented system that includes system functionality supported by the graph. The results highlight the potential of simulation suggesting its capacity for valuable exploration regarding potential applications of knowledge graphs within production systems.

Keywords. Knowledge Graphs, Industrial Applications, Manufacturing Simulation, System Representation

1. Introduction

As a paradigm, Industry 4.0 (I4.0) raises high expectations for production systems. Among other things, the challenges consist of factory digitization, reaching even higher

---

[1] Corresponding Author: Martin Birtic, martin.birtic@his.se

levels of automation, rapid reconfigurability, flexibility, interoperability and general intelligent smartness [1]. To meet these challenges, research has been initiated on many fronts which has led to larger technical concepts such as cyber-physical production systems and digital twins which both seek to exploit the combination of physical and virtual elements to achieve a variety of functions [2]. An additional multitude of supporting concepts and technologies have emerged to facilitate the realization of I4.0 production strategies such as advancements in artificial intelligence, knowledge techniques, data analytics, simulation, and smart production.

Knowledge graphs (KG) have recently generated a lot of interest in both academic and industrial domains [3] as a useful technology that can potentially be useful in advancing a variety of concepts such as knowledge automation [4], manufacturing process planning [5], maintenance planning [6], system reconfiguration [7], collaborative manufacturing [8], manufacturing service matchmaking [9], intelligent manufacturing equipment [10], resource allocation and decision-making [11], factory data integration [12], digital twin modelling [13], and cyber-physical system and component description and integration [14].

However, despite their potential benefits KGs in industrial settings are still preliminary and more work is motivated to investigate their applications [15] and hence we propose to further investigate their use through the utilization of virtual environments.

Simulation has been identified as an important tool within the I4.0 paradigm [16] where certain types of simulation methods effectively support the development, testing and commissioning of complex production systems. In particular, Virtual Commissioning (VC) as a simulation method has been identified as a cornerstone for working with these types of systems [17]. VC simulations include the software and control programs expected to be used in the real physical system. The simulation model includes virtual representations of the system's components and their control and interactions with each other and with products. By building a virtual representation of the system, design decisions and control programs can be continuously and iteratively tested and evaluated while interacting with a representative virtual system and its dynamics. [18].

Section 2 considers the research efforts of various researchers investigating KG applications in industrial contexts. Notably, the authors are unaware of any instances where a VC environment integrates a KG database as a core element. This perceived gap, in the authors' view, limits experimentation and exploration of the capabilities of these systems.

The aim of this study is to develop a simulation framework based on VC modeling and simulation that also uses KG for system functions to address the exploration of KG potential in production systems. The simulation specifically includes service matching and scheduling functions previously tested separately using KGs by other researchers [9, 5] as well as a modular approach and a service-oriented architecture.

The following details are arranged in the corresponding order. Section 2 reviews relevant concepts from the literature and briefly presents ways in which other researchers are using KG technology. Section 3 outlines the simulation environment which is discussed in section 4. Finally, conclusions and future work are presented in section 5.

2. Related concepts

2.1. Virtual engineering and digital system development

In this article, virtual engineering and system development refers to the use of specialized software in order to explore and develop concepts and solutions related to complex production systems. Simulation is the central key component in this context and different simulation methods have their specific strengths, weaknesses and fields of application [16]. This study operates under the assumption that simulations that mimic and represent real systems more authentically also provide a better virtual development environment for experimentation and development of new features and technology. Consequently, a methodology based on VC modeling and simulation has been selected as the approach. This method has been shown to accurately reproduce the behavior of a production system in such detail that faults can be identified before physical commissioning [17].

Methods for VC have been developed over time based on the need to develop, test, and debug production system control programs prior to downloading and deployment into a real physical system. Before virtual tools were widely adopted, programs could be debugged on a mock-up variant of the physical system. Such a system could reproduce certain parts of the dynamics under which control programs for PLCs and robots operate. However, such systems could be costly, unavailable or occupied and could also be damaged if unexpected program sequences were executed. This type of mock-up system can frequently be replaced by a virtual equivalent whose dynamics and behavior are coded using timers, counters and logic. By using these virtual mechanisms with the system's inputs, outputs and logic programs, tests could be performed to investigate the interaction of the PLC or robot program with the dynamics simulated by the virtual mock-up system.

Gradually, VC has evolved from using virtual mock-up systems to using more realistic 3D models. Today, there are several off-the-shelf software that can simulate a variety of PLCs, robots, HMIs and other peripherals combined into sophisticated simulation models. With the help of realistic 3D visualization and simulation of physical laws, an accurate imitation of system dynamic behavior is enabled.

During this development, the VC method has also been used in other areas, in addition to PLC and robotic control code development and troubleshooting, including prototyping production system solutions, planning updates, operator training, and system optimization. And more recently, it has been suggested that this type of VC model also has the potential to be turned into a digital twin [17].

Developing a VC-type model and simulation involves establishing a software framework that reflects the set of software that will be used in the real situation, and also includes software that can realistically represent the physical system. Physical systems consist of sensors and actuators, which need to be virtualized and complemented with realistic behavior to interact with virtual products and production resources in a way that reflects their dynamics. During simulation, various events lead to the triggering of virtual sensors whose signals are received by the control system. The control programs process the signals and produce output signals that in turn activate the model's virtual actuators.

In the ultimate situation, developers have more or less a theoretically possible production ecosystem where the main difference is that the physical system is digitized and lives inside a computer. Such a situation allows for progressive exploration and innovation in a safe but realistic environment where ideas can be iteratively developed, and have their feasibility evaluated.

2.2. Knowledge graphs

KGs are a way to structure knowledge and concepts (often human knowledge) using the mathematical structure of a graph. In a KG concepts become nodes of a graph and edges connect the nodes/concepts indicating a relationship between those nodes. Edges and nodes in a KG can (and usually do) have additional data most importantly being the name or type of the edge, indicating how two nodes are related. How raw data is encoded varies by implementation but the underlying concept is that data, such as names, identifiers and values are attached to nodes.

KGs have their roots in the concept of semantic networks and the philosophical study of ontologies for describing classes of things and the relationships between these classes. Within the fields of computing and AI (Artificial Intelligence) more specifically this approach has provided a powerful underlying model for knowledge representation and identifying patterns within such data.

Semantic networks are also a core concept within the field of the *semantic web* [19]. The Semantic Web as a concept appeared in the early 2000s as an enhancement to the internet. The concept was that the existing internet connected structured text documents (HTML) through hyperlinks but if each document represented concepts and relationships then the information or knowledge itself would be linked. The network of knowledge would then be exposed and searchable. Additional rules and reasoning can then be expressed through representing abstractions of knowledge as ontologies, which can themselves be expressed in the same structured documents. This lends itself to a wider definition of what a KG is, that it is the connections and integration of information being drawn from different sources into a conceptual network as proposed in [20].

The vision of the original semantic web has not yet come about, however the approach has been found to be general enough to express almost any data and the relationships between them. This is then very powerful for data interchange and integration between heterogeneous systems. For example, in a production system there might be a variety of machines from different providers, each storing their data internally in different formats. The semantic web formats allow each to present their data and capabilities using a standard format and this can then either be interacted with directly or gathered and integrated into a database. The model is that of a KG formed of simple *triples* formed of;

- a subject; a node or identifier for a node
- a predicate; a relationship
- an object; a node that the subject has a relationship with

This model can be stored in a range of formats but the usage of the Resource Description Framework (RDF) and XML is common [21].

Several database implementations have developed following the approach of KGs, all placed within the NoSQL category. Those closest to the semantic web's infrastructure are sometimes called triple stores, because they store and present information in the semantic web's subject-predicate-object pattern. Triple stores will usually return information in RDF and integrate the SPARQL query language as the primary method for interaction with the tool. They will usually have the capability to include a formal ontology within the database and link data to it as instances, using the ontology as both a database schema and for the inference of connections that are believed to exist. Some examples of triplestores are; Ontotext's GraphDB and Apache Jena.

An alternative approach that has gained significant traction in the last decade are based on the concept of *property graphs*. In this model the nodes and edges are still present but data can be stored on nodes and edges either to enhance and specialise the meaning of an edge or more usually to store data on a node. This model is the basis for Graph Databases which usually includes the triplestores but also includes tools such as Neo4J and TerminusDB. Neither of these examples provide SPARQL as standard instead using other query languages, Cypher in the case of Neo4J and GraphQL in the case of Terminus.

KGs provide benefits over traditional databases in terms of flexibility of the structures but also the ability to search for patterns and by the relationships between different instances of the data, including finding unexpected relationships and connections. Many of the tools also take advantage of the model of the data as a *graph* to provide powerful visualisation tools to explain the results of searches or the inferences being made. These advantages are leading to growing use in industry and research into more sophisticated uses. Most commonly in terms of application are those related to knowledge management and analytics, either of company documentation [22] or to improve the representation of data found in models of products or machines [23]. Another common use of KGs, especially using RDF and SPARQL type technologies has been in integration between heterogeneous data and devices, a common problem in industrial settings and one that needs to be overcome to realise the benefits of industry 4.0 [24,25]. Related to the integration of data is how to effectively query or understand the data being seen. This has given rise to the concept of an industrial *semantic shell* as discussed in [26]. This usage has been extended to the integration between DTs and the capability of systems to discover the related real physical assets using semantic technologies [27]. Several researchers have also been considering the use of KGs in support of digital twins, either in automating connection towards real world devices [28] or to provide ways to supply statistical data from the real machines to support the simulation aspect of digital twins [29]. Still others have begun to investigate using KGs in support of production planning by executing graph algorithms directly on models of data within graph databases [30].

This illustrates the range of different applications of KGs in industry being investigated by researchers at the current time. However, these examples do not include clear examples of VC systems that integrate KGs to speed the integration and research into this technology in future factory developments. Hence this is the direction the remainder of this paper will take.

3. Simulation

The virtual simulation environment consists of four software tools that communicate using the OPC UA protocol, see figure 1. The digital modeling software used in this study is RobotStudio, where all system components are virtualized. Here, robots are emulated and controlled internally, while PLC-controlled virtual components execute their control programs in CoDeSys. Knowledge graphs are managed in Neo4J and the central service orchestration is implemented in Python. Communication between the three pieces of software and the orchestrator enables the modules to access and participate in the system's orchestration.

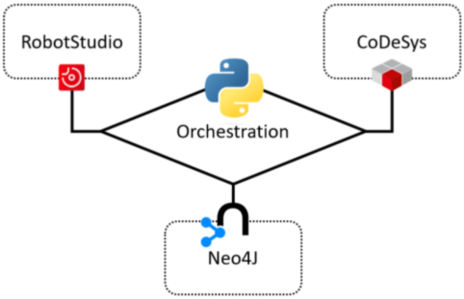

Figure 1. Software framework. All virtual equipment is modeled in RobotStudio where the robot control also takes place. PLC controlled equipment executes control programs in CoDeSys. The orchestrator use a service oriented approach to read and write information to the system's knowledge graph, which is implemented in Neo4j.

The basic element of the production system a standardized module. This module follows strict physical standardization, which means a conformity to pre-decided dimensions and placement of material transport equipment. This enables a seamless physical connection of modules enabling a quick establishment of product flow. Two material transport inputs and two outputs are used in the standardized main configuration, which form the horizontal material flow, FLOW2, and the vertical, FLOW1, see figure 2a. The system handles standardized product pallets which carry unique identifiers encoded in RFID tags. Pallets positioned where the two flows converge can be processed by the module. A transfer conveyor at this position can send the pallet in both directions. There are two main module types each instantiated three times in the system. Module 110, 130, and 220 has a robot with its own decentralized robot controller used to control the module's peripherals such as conveyors and beacons. Modules 120, 210, and 230 have no robot and are completely controlled by PLC programs. Two modules in the system, 100 and 500 are not based on this physical layout but conform to physical interfaces nonetheless. In module 100, pallets are introduced and removed, while the 500 module is used to recirculate pallets if necessary. To achieve seamless interoperability, standardization of logical interfaces for intramodular communication is also implemented. In addition, standardized communication mechanisms are established, using handshake and feedback mechanisms to ensure the robust flow of information and materials between modules.

The knowledge graph is populated with details of modules and their physical interconnections (see figure 2), encompassing data on each module's offered services. Product-related information, including a unique identifier, service requirements, arrival times at modules, and transfer details, is recorded in the KG. As product pallets move between modules, the KG is continuously updated to reflect these changes to maintain an overview.

Modules offer process and transportation services. At a granular level, these services are encapsulated as micro-services and composite services. Micro-services are small-

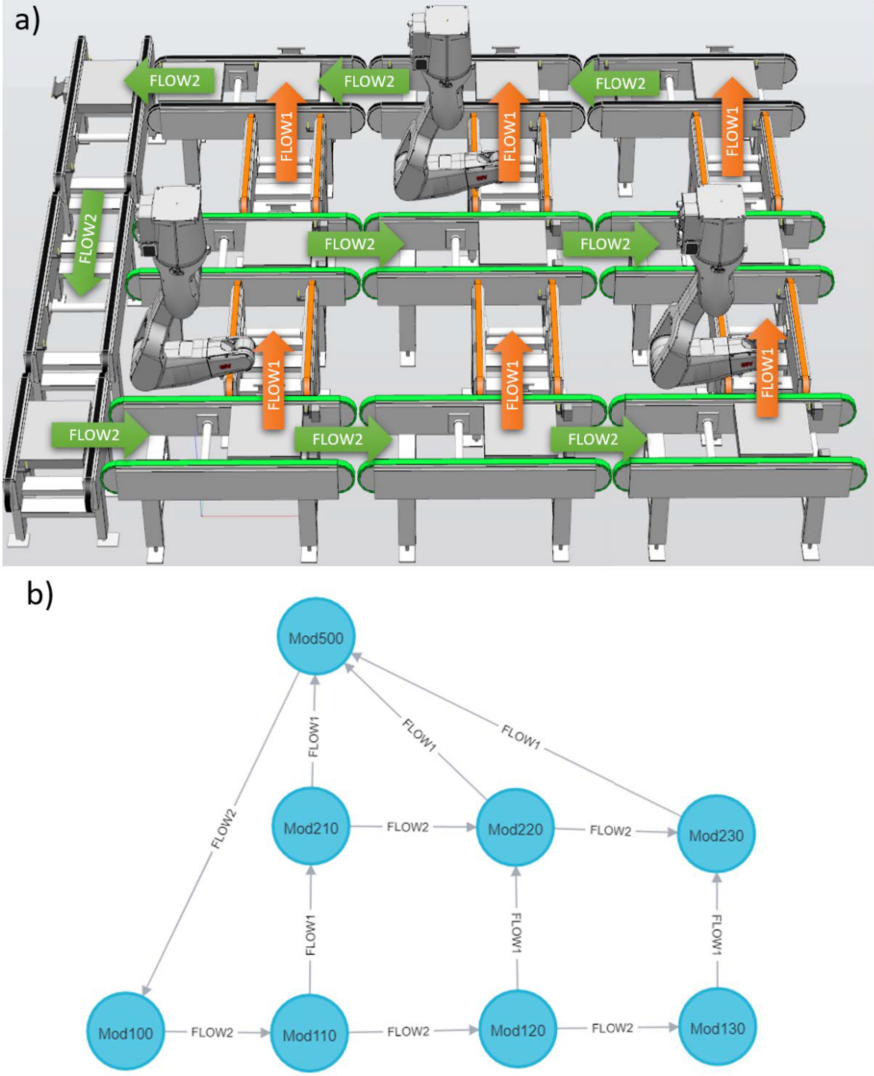

Figure 2. Image illustrating a) virtual model and flows, b) knowledge graph representation of modules and physical pallet flows.

scale operations, such as simple robot movements, discrete tasks, or short conveyor transportation. Composite services on the other hand, integrate micro-services and standalone code to execute more complex functions. In this simulation, micro-services are only used locally within each module, while only composite-services are registered and available via the system's orchestrator.

A key assumption made in the simulation is that product pallets can hold relevant products and that modules can perform specific process services on them. In the simulation however, specific products and relevant process services are not used, but empty pallets and placeholder services perform dummy actions. A more detailed interaction diagram between modules, products, orchestrator, and KG can be seen in figure 3.

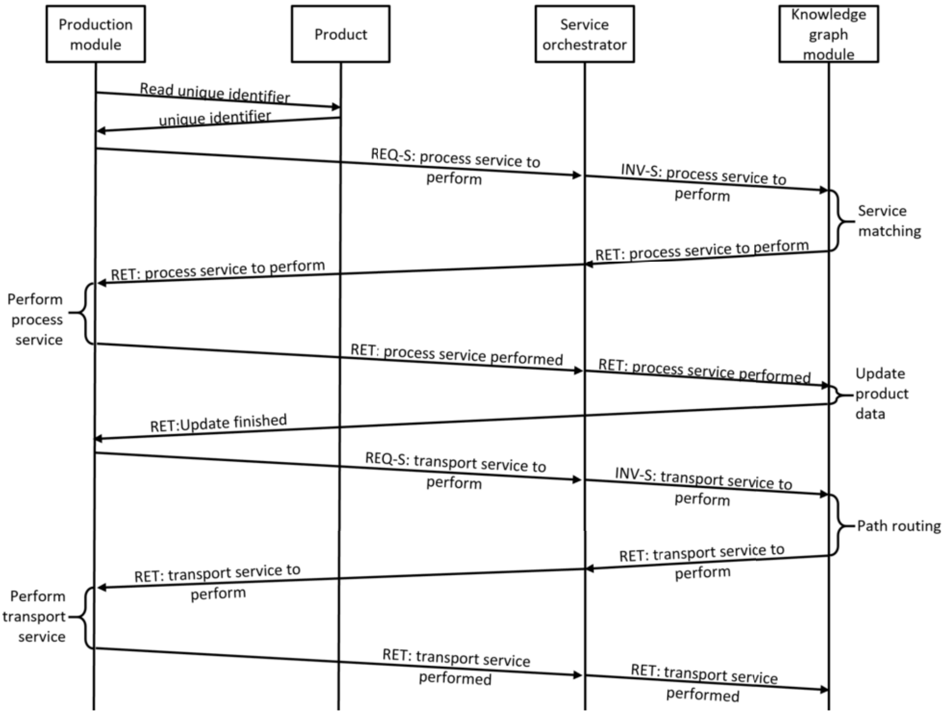

Figure 3. Signal diagram illustrating service-oriented signals and KG. Requesting service is REQ-S, invoking service is INV-S, return if data is RET.

3.1. Simulation operation and KG services

The module's operational sequence is illustrated in Figure 4a. Upon startup, each module undergoes an initialization procedure, initiating signals and informing its status through a light beacon. An empty module then enters an idle state until a preceding module signals its readiness to transfer a pallet. The receiving module transitions into a loading state until the product pallet is loaded and positioned in a processing position. A loaded module initiates a request to identify if any process services are to be performed to the pallet. This is done by reading the pallet's RFID to extract its unique identifier. The module contacts the orchestrator, which queries the knowledge graph to match the pallet's next service need with the current module's service capabilities. If there's a match, the orchestrator return the service identifier causing the module to enter a state for performing the process service. Upon completion, the module communicates the service execution to the orchestrator, which updates the knowledge graph regarding the pallet's service needs. The module then enters a state of requesting transport services to initiate the transport of the pallet to the next module. The orchestrator, once again, queries the knowledge graph, finding the nearest module offering the next required service. The shortest path is identified, and the correct flow (FLOW1 or FLOW2) is communicated to the module for transport initiation to the next module on the path. Figure 4b illustrates how product 1022's next need can be met by module 230, while figure 4c shows how product 2024 needs to circulate back to module 130 via module 500.

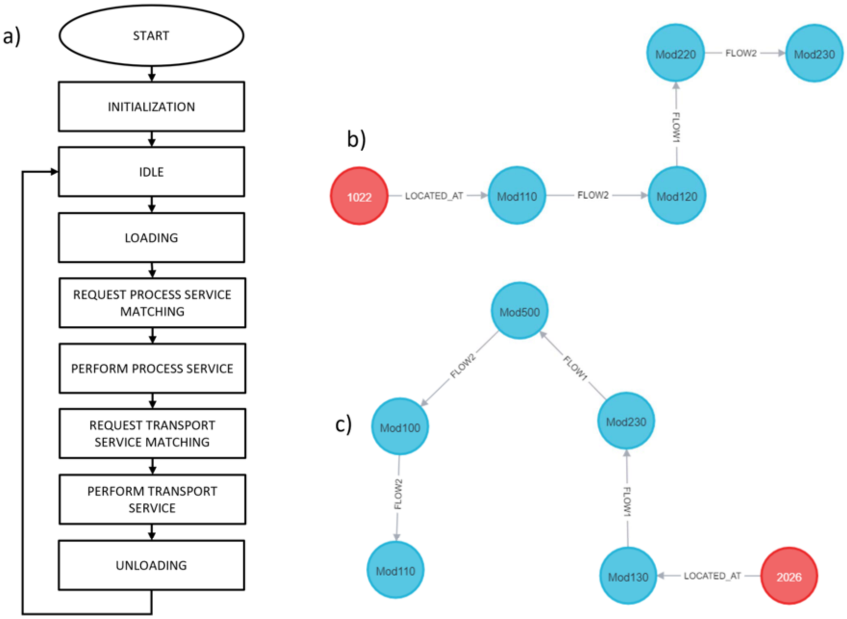

Figure 4. a) operational sequence of modules, b) forwards path to next service, c) using re-circulation module 500 to re-enter the system to reach next service.

Upon receiving flow information, the module transitions into a transport performance state, activating the corresponding transport service and passing the pallet to the next module, while entering an unloading state. Upon reception by the receiving module, the sending module returns to the idle state. In cases where entering the idle state involves waiting pallets on both input flows, a priority service is requested through the orchestrator. Arrival times retrieved from the knowledge graph are used for comparison before communicationg to the module to load the pallet that has been waiting the longest.

4. Discussion

A production simulation framework has been developed that integrate and use knowledge graphs. The main purpose is to enable the exploration of KG potential to be used in industrial applications. A modular production system was modeled, and a knowledge graph was populated with an abstracted version of the system, continuously updated with new data. The system model is dynamic and its virtual equipment is controlled by control programs executed in robot and PLC emulators. This means that KG is embedded in this environment and interacts with these systems.

With integrated KG technology, loaded and updated with carefully selected data from the production system, the system can have ”a voice”. Through tailored queries, relevant system facts or data that are otherwise tricky to access can be requested by system actors using services. In the study, this is demonstrated by production modules independently executing their programmed sequence up to the point that external information is necessary. With the help of a service request, information is extracted by querying the

KG system overview. Once the information is received back, the modules can continue the sequence now operating on data derived from a system-wide context. In this setup, on the one hand, there is a complex control system that needs information in order to operate effectively. On the other hand, there is KG technology that can synthesize information based on its content. Bringing these together is a promising combination with great potential that needs to be explored. The general picture that emerges is that this type of exploratory simulation can contribute to the exploration of interesting possibilities for KG. However, there are several limitations to this study, including that the simulation model is not primarily designed to represent a real production system with realistic products, processing services, or safety, etc. The model has been created primarily with the exploration of KG in mind, which makes the results difficult to validate in this study. Furthermore, assumptions and simplifications have been made which affect the dynamics and the situations that the system exhibits in the simulation. This means that important events and problems are not detected and are not influencing the development of the system's functions. Nevertheless, we believe that development and exploration using this type of virtual simulation methodology is the next best thing compared to also having access to a physical environment for validation, as this allows exploration and innovation without access to costly hardware systems. However, validation through implementation in hardware is beneficial and is inevitable to achieve solutions of value.

5. Conclusion and Future Work

A KG was integrated into a VC-style simulation framework where it had to interact with PLCs, robot programs and product dynamics in a modular production system. The study shows how the exploratory use of KGs in VC simulation can support the study of their potential for industrial applications. The implemented KG was a key component of the system's service-oriented architecture as it was used for service registration and matching as well as product routing and prioritization. By continuously loading the graph with updated system information, it is possible to extract otherwise hard-to-access information from it using queries which can be seen as the system acquiring ”a voice”. In addition, otherwise implicit or explicit system details and patterns can be powerfully filtered and visualized.

The study's exploratory simulation method, while valuable, has some limitations. The simulation model, primarily designed for knowledge graph exploration, lacks representation in real life, which affects its validation. Assumptions and simplifications affect system dynamics and overlook potentially decisive events.

Despite these challenges, virtual simulation remains a valuable tool that enables exploration and innovation without expensive hardware. Nevertheless, hardware implementation is critical to achieving practical and valuable solutions.

References

[1] Lu Y. Industry 4.0: A survey on technologies, applications and open research issues. Journal of industrial information integration. 2017;6:1-10.

[2] Tao F, Qi Q, Wang L, Nee A. Digital twins and cyber–physical systems toward smart manufacturing and industry 4.0: Correlation and comparison. Engineering. 2019;5(4):653-61.

[3] Kejriwal M. Knowledge graphs: A practical review of the research landscape. Information. 2022;13(4):161.
[4] Song G, Fu D, Zhang D. From Knowledge Graph Development to Serving Industrial Knowledge Automation: A Review. In: 2022 41st Chinese Control Conference (CCC). IEEE; 2022. p. 4219-26.
[5] Xiao Y, Zheng S, Shi J, Du X, Hong J. Knowledge graph-based manufacturing process planning: A state-of-the-art review. Journal of Manufacturing Systems. 2023;70:417-35.
[6] Xia L, Liang Y, Leng J, Zheng P. Maintenance planning recommendation of complex industrial equipment based on knowledge graph and graph neural network. Reliability Engineering & System Safety. 2023;232:109068.
[7] Mo F, Chaplin JC, Sanderson D, Martínez-Arellano G, Ratchev S. Semantic models and knowledge graphs as manufacturing system reconfiguration enablers. Robotics and Computer-Integrated Manufacturing. 2024;86:102625.
[8] Nagy L, Ruppert T, Abonyi J. Human-centered knowledge graph-based design concept for collaborative manufacturing. In: 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE; 2022. p. 1-8.
[9] Zhao Y, Liu Q, Xu W. Open industrial knowledge graph development for intelligent manufacturing service matchmaking. In: 2017 International Conference on Industrial Informatics-Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII). IEEE; 2017. p. 194-8.
[10] Yan H, Yang J, Wan J. KnowIME: a system to construct a knowledge graph for intelligent manufacturing equipment. Ieee Access. 2020;8:41805-13.
[11] Zhou B, Bao J, Li J, Lu Y, Liu T, Zhang Q. A novel knowledge graph-based optimization approach for resource allocation in discrete manufacturing workshops. Robotics and Computer-Integrated Manufacturing. 2021;71:102160.
[12] Dombrowski U, Reiswich A, Imdahl C. Knowledge graphs for an automated information provision in the factory planning. In: 2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM). IEEE; 2019. p. 1074-8.
[13] Banerjee A, Dalal R, Mittal S, Joshi KP. Generating digital twin models using knowledge graphs for industrial production lines. UMBC information systems department. 2017.
[14] Grangel-González I, Halilaj L, Vidal ME, Rana O, Lohmann S, Auer S, et al. Knowledge graphs for semantically integrating cyber-physical systems. In: Database and Expert Systems Applications: 29th International Conference, DEXA 2018, Regensburg, Germany, September 3–6, 2018, Proceedings, Part I 29. Springer; 2018. p. 184-99.
[15] Buchgeher G, Gabauer D, Martinez-Gil J, Ehrlinger L. Knowledge graphs in manufacturing and production: A systematic literature review. IEEE Access. 2021;9:55537-54.
[16] de Paula Ferreira W, Armellini F, De Santa-Eulalia LA. Simulation in industry 4.0: A state-of-the-art review. Computers & Industrial Engineering. 2020;149:106868.
[17] Lechler T, Fischer E, Metzner M, Mayr A, Franke J. Virtual Commissioning–Scientific review and exploratory use cases in advanced production systems. Procedia CIRP. 2019;81:1125-30.
[18] Albo A, Falkman P. A standardization approach to virtual commissioning strategies in complex production environments. Procedia Manufacturing. 2020;51:1251-8.
[19] Berners-Lee T, Hendler J, Lassila O. The Semantic Web. Scientific American. 2001 May;284(5):34-43. Available from: http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21.
[20] Fensel D, Şimşek U, Angele K, Huaman E, Kärle E, Panasiuk O, et al. Introduction: What Is a Knowledge Graph? In: Fensel D, Şimşek U, Angele K, Huaman E, Kärle E, Panasiuk O, et al., editors. Knowledge Graphs: Methodology, Tools and Selected Use Cases. Cham: Springer International Publishing; 2020. p. 1-10.
[21] Beckett D. RDF/XML Syntax Specification (Revised). W3C; 2004. Http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/. Available from: http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/.
[22] Zhou B, Bao J, Liu Y, Song D. BA-IKG: BiLSTM Embedded ALBERT for Industrial Knowledge Graph Generation and Reuse. In: 2020 IEEE 18th International Conference on Industrial Informatics (INDIN). vol. 1; 2020. p. 63-9.
[23] Bharadwaj AG, Starly B. Knowledge graph construction for product designs from large CAD model repositories. Advanced Engineering Informatics. 2022;53:101680. Available from: https://www.sciencedirect.com/science/article/pii/S1474034622001409.

[24] Diedrich C, Belyaev A, Schröder T, Vialkowitsch J, Willmann A, Usländer T, et al. Semantic interoperability for asset communication within smart factories. In: 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA); 2017. p. 1-8.

[25] Grangel-González I, Lösch F, ul Mehdi A. Knowledge Graphs for Efficient Integration and Access of Manufacturing Data. In: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). vol. 1; 2020. p. 93-100.

[26] Grangel-González I, Halilaj L, Coskun G, Auer S, Collarana D, Hoffmeister M. Towards a Semantic Administrative Shell for Industry 4.0 Components. In: 2016 IEEE Tenth International Conference on Semantic Computing (ICSC); 2016. p. 230-7.

[27] Rongen S, Nikolova N, van der Pas M. Modelling with AAS and RDF in Industry 4.0. Computers in Industry. 2023;148:103910. Available from: https://www.sciencedirect.com/science/article/pii/S016636152300060X.

[28] Ploennigs J, Semertzidis K, Lorenzi F, Mihindukulasooriya N. Scaling Knowledge Graphs for Automating AI of Digital Twins. In: Sattler U, Hogan A, Keet M, Presutti V, Almeida JPA, Takeda H, et al., editors. The Semantic Web – ISWC 2022. Cham: Springer International Publishing; 2022. p. 810-26.

[29] Senington R, Baumeister F, Ng A, Oscarsson J. A linked data approach for the connection of manufacturing processes with production simulation models. Procedia CIRP. 2018;70:440-5. 28th CIRP Design Conference 2018, 23-25 May 2018, Nantes, France. Available from: https://www.sciencedirect.com/science/article/pii/S221282711830413X.

[30] Zhu Z, Zhou X, Shao K. A novel approach based on Neo4j for multi-constrained flexible job shop scheduling problem. Computers & Industrial Engineering. 2019;130:671-86. Available from: https://www.sciencedirect.com/science/article/pii/S0360835219301585.