Design Automation for Cyber-Physical Production Systems: Lessons Learned from the DeFacto Project

Michele Lora[1], Sebastiano Gaiardelli[1], Chanwook Oh[2], Stefano Spellini[3], Pierluigi Nuzzo[2], Franco Fummi[1]
[1]*Industrial Computer Engineering Laboratory – Dept. of Engineering for Innovation Medicine, University of Verona, Italy*
[2]*Dept. of Electrical and Computer Engineering, University of Southern California, Los Angeles, CA, U.S.A.*
[3]*Factoryal S.r.l., San Giovanni Lupatoto, Italy*

***Abstract*—The *DeFacto* project, supported by the European Commission via a Marie Skłodowska-Curie Global Individual Fellowship, tackles the complexity arising from the transformation of industrial manufacturing systems into intricate cyber-physical systems. This evolution offers unprecedented opportunities but also poses intellectual and engineering challenges. DeFacto aims to advance the design automation of cyber-physical production systems by developing innovative modeling paradigms, scalable algorithms, software architectures, and tools.**

In the *DeFacto* approach, production systems are managed through service-oriented manufacturing software architectures. System-level models capture the features and the requirements of production systems, representing both production and computational processes as services provided by the infrastructure. Methodologies for system analysis and optimization rely on compositional abstractions of system behaviors grounded in assume-guarantee contracts. This paper outlines key research endeavors, findings, and lessons learned from the *DeFacto* project.

***Index Terms*—Smart manufacturing, system-level design, design automation, software architecture, system modeling.**

I. Introduction

Manufacturing is undergoing what has been identified as "a fourth industrial revolution", turning factories into complex, heterogeneous, cyber-physical production systems (CPPS) [1], capable of offering unprecedented opportunities. Production lines become complex networked systems, equipped with teams of robots that manipulate products and take care of the logistics, sensors gathering large amounts of data, and computing platforms providing situational awareness and "intelligent" control. In "smart" factories, reconfigurable production lines will be able to handle the increasing demand for customized products and move from mass production to mass customization [2]. Data analytics will enable the optimization of production costs, supply and logistics chains, and the integration of different aspects of the value production chain.

These opportunities come, however, with a series of intellectual, engineering, and social challenges. A smart manufacturing system must integrate a diverse set of components while offering strong guarantees in terms of functionality, reliability, safety, and cost. This heterogeneity in components and system requirements inevitably calls for models, specification formalisms, and design constraints of different nature. However, it is difficult to extensively explore the design space of such heterogeneous systems in a reasonable amount of time, an issue that is often denoted as the "explosion in complexity" of today's CPPS design. Increased complexity, high design costs, and the lack of the required technological skills may eventually prevent companies from embracing novel production paradigms that may substantially benefit the economy and the society at large. This holds especially true for small and medium-sized enterprises (SMEs) which serve as the backbone of the European economy. Facilitating the adoption of smart manufacturing solutions has, indeed, become strategic for the European Union (EU), and is already part of the policies of major EU countries, such as Germany's initiative for *Industrie 4.0* and the Italian plan *Transizione 4.0*.

Mirroring the success of electronic design automation (EDA) in taming the complexity of microchip design [3], *system design automation* can play a crucial role toward appropriately addressing the complexity of cyber-physical systems (CPS) design. The objective of *DeFacto* (*Design automation for smart Factories*)[1] is to advance the state of the art in system design automation by developing novel modeling paradigms, system architectures, scalable algorithms, and tools to aid the design of smart manufacturing systems. The project had three main research objectives:

1) Identify requirements and architectures for the design of CPPSs, and define representations for the requirements and the components at different abstraction levels.
2) Develop a formal methodology, algorithms, and computational tools for requirement validation, design-space exploration, and model refinement. The methodology enables the derivation of the system architecture and a set of control algorithms from the system requirements.
3) Develop synthesis and mapping algorithms to generate software implementations from higher-level models of the system architecture and the control algorithms.

The project started in October 2020 and ended at the end of September 2023. The collaboration between the *Electronic Systems Design (ESD) Research Group* at the *University of Verona* (UniVR), and the *Cyber-Physical System Design (DesCyPhy) Lab* at the *University of Southern California* (USC) was funded by the European Commission via a Marie Skłodowska-Curie Global Individual Fellowship.

Figure 1 shows the different stakeholders involved in the project and their contributions. Over the years, the consortium had the chance to interact with a diverse set of companies, and collect heterogeneous requirements. At USC, we interacted

Corresponding author: *Michele Lora*, michele.lora@univr.it

[1]https://defacto-h2020.github.io/

979-8-3503-4859-0/DATE24/© 2024 EDAA

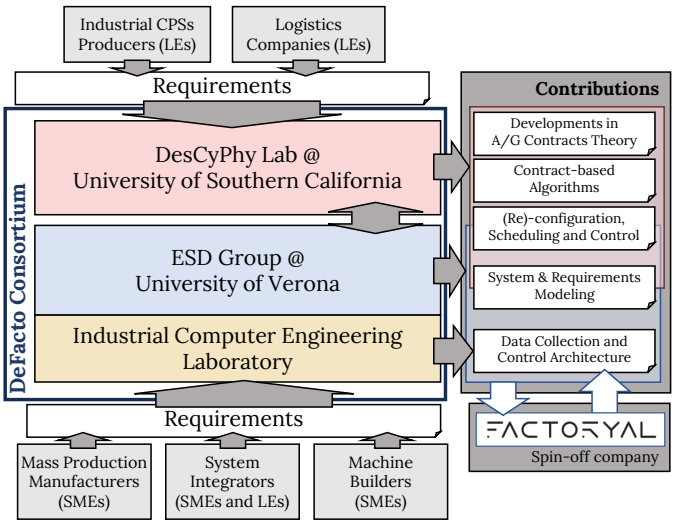

Fig. 1: Project stakeholders and contributions. The consortium collected real-world requirements from a diverse set of companies, from large logistics companies to small machine builders. The results with a higher technology readiness level led to a spin-off company which also contributed in developing the data collection and control architecture.

mostly with large enterprises (LEs), specifically *industrial CPS producers* and *logistics companies*. UniVR interacted with companies via the Industrial Computer Engineering (ICE) laboratory[2], a research laboratory equipped with a fully fledged manufacturing line, meant to support research activities as well as showcase cutting-edge manufacturing technologies to local companies. Thus, UniVR interacted mainly with medium-sized enterprises operating in different fields: *mass production manufacturers*, *system integrators*, and *machine builders*. These interactions allowed us to build a well-rounded view of today's automation requirements.

Driven by the identified industrial requirements, the project advanced the development of design methods based on assume/guarantee (A/G) contracts, including algorithms for architectural exploration and decision making under uncertainty [4]–[7]. The algorithmic advancements were paired with their implementation within the *CHASE* framework [8], [9]. Moreover, the consortium introduced new system-level modeling strategies [4]–[6], [10], [11] and defined a novel software architecture, paired with configuration and scheduling algorithms, to enable service-oriented manufacturing (SoM) [12]. Parts of these results led to a spin-off company, *Factoryal*, operating in the field of factory automation, which engineered the data collection and control SoM architecture.

In this paper we summarizes key results from the project while discussing the lessons learned from applying design automation methodologies to advanced manufacturing systems.

II. Context and Background

The automation pyramid, originally introduced in 1985 [13], is still the centerpiece of most industrial automation software systems, and survived the evolution in automation software we have witnessed in recent years. Despite the recent hype surrounding factory automation systems that enable the Industry 4.0 revolution [1] and Industry 5.0 [14], the adoption of such paradigms by companies seems still distant. Even though they are the backbone of the European economy, and despite the accelerating effect of the COVID-19 pandemics [15], many SMEs are still reluctant to embrace automation and digitalization. Over the past years, we discussed with multiple manufacturing companies, finding out that many companies are not only using software systems that are still based on the original automation pyramid, but they are also overlooking many of the pyramid layers.

By exploring the pyramid from the top to the bottom, we witnessed that most companies rely on Enterprise Resource Planning (ERP) systems, while Manufacturing Execution Systems (MESs) are less frequently used, especially by SMEs. Supervisory Control and Data Acquisition (SCADA) systems are often limited to human-machine interfaces used to start and stop manufacturing processes, and monitoring some field data. Often, we encountered companies relying on office automation tools, such as spreadsheets and word processors, to monitor and manage manufacturing processes, rather than using specialized automation software. At the lowest levels of the pyramid, *i.e.*, control and field levels, many manufacturing enterprises employed machines that were unable to provide network connections or grant easy access to their data or control commands, thus making factory automation even more complicated. Companies may have considered upgrading their systems. However, any upgrade often requires huge investments in terms of costs for updating systems and machines and costs to train the personnel. Moreover, upgrading a productive site will require halting the production, thus introducing downtime costs. These are the main reasons why a significant number of companies tend to postpone the adoption of advanced manufacturing technologies.

We also witnessed a genuine interest in advanced production technologies, not only when interfacing with large enterprises, but also when interfacing with SMEs. Interestingly, many of the stakeholders we interacted with were particularly interested in methodologies and tools that enable abstraction and reasoning about system properties. To this end, we investigated modeling languages as well as formal languages useful to specify and analyze CPPSs. Concerning modeling languages, model-based software engineering (MBSE) is a well-established field enabling the encapsulation and abstraction of system architectures and functionalities into models. These models can encompass diverse components, behaviors, and dynamics, while accommodating a wide range of viewpoints [16]. Consequently, MBSE approaches are well-suited for the design of advanced manufacturing systems [17], [18]. SysML and AutomationML are popular modeling languages in this field [19]. AutomationML is specifically designed for manufacturing systems; SysML is a well-rounded language for generic system modeling. In our analysis, AutomationML fell short when modeling smart manufacturing systems, while SysML was more suitable to create unified models for today's

[2]The ICE laboratory website: https://www.icelab.di.univr.it/

CPPSs with computing and communication capabilities [20].

Concerning formal specification languages, *DeFacto* addresses CPPS design using compositional abstractions of system behaviors based on A/G contracts. Contracts are mathematical models which offer rigorous composition rules and provide mechanisms to validate the design requirements, analyze complex system behaviors, and develop systems in a modular and hierarchical way. Effective contract-based system-level design methodologies for CPSs had already appeared in the literature [16]. Practical tools were also being developed, showcasing the applicability and effectiveness of A/G contracts in dealing with complex systems [8]. However, further advancements were required in both theoretical foundations and automation to support the analysis of complex CPPSs.

III. The DeFacto Approach

Given the context described above, we organized our research according to three main directions:

- We defined a *distributed software architecture* based on the SoM paradigm, which acts as an abstract layer to access the factory resources and functionalities. The proposed software architecture is particularly effective in enabling the digital transition for SMEs;
- We proposed *modeling strategies* to simplify the specification of production system features, machines functionalities, and production requirements. The defined models specify manufacturing operations as services provided by the production infrastructure. As such, they are able to support the main concepts in the SoM paradigm implemented by the proposed software architecture;
- We proposed system design and analysis methodologies to support the *automated exploration of the system architectures*, the *automated generation of control software*, and the *system optimization*.

Figure 2 depicts the overview of the project's contributions and their relationships. Production systems must be equipped to support SoM. To this end, a system is equipped with a distributed software layer that abstracts the functionalities provided by each piece of equipment as a set of services provided by the infrastructure. The Advanced Manufacturing Controller (AMC) is the software that schedules system executions. It takes scheduling decisions by monitoring the data coming from the field and analyzing the information stored in the models. Section IV will delve into the developed architecture.

Descriptions of the system's features and functionalities may be stored in different types of models, such as AutomationML and BPMN models. In *DeFacto*, the information contained in such models is unified into SysML representations, which capture both the cyber and the physical aspects of a CPPS. Furthermore, a user may complete the unified SysML model by specifying missing information or production requirements. Thus, SysML models enclose the information necessary to configure and manage the SoM architecture, and consequently control the system. Section V focuses on the modeling strategies proposed in the project.

As the information about the system is encapsulated by models, it is formalized in terms of A/G contracts. The

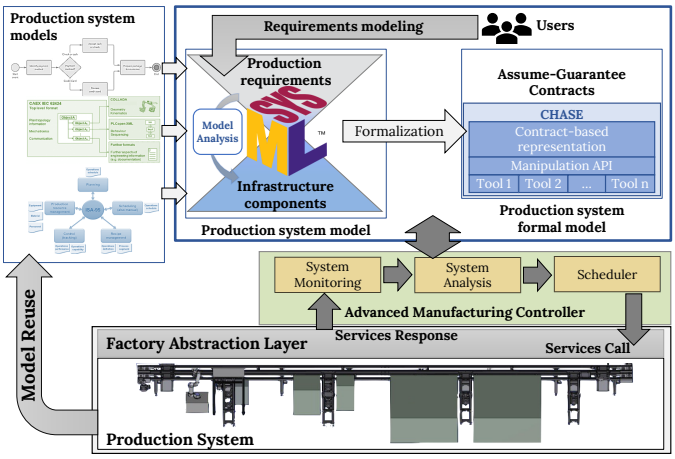

Fig. 2: Overview of the *DeFacto* approach. The information available about the production system is expressed in SysML models and formalized in CHASE using A/G contracts. The AMC processes the information, analyzes the data from the field, and takes control decisions. The AMC communicates with the machinery through a factory abstraction layer by invoking services and collecting the service responses.

CHASE (Contract-based Heterogeneous Analysis and System Exploration) framework [8] automates the contract-based methodologies that enable exploration of the reconfigurable system architectures, design of the control software, and scheduling of the production lines. The methodologies can be used by designers to design the system, as well as by the AMC to govern the production line. Section VI summarizes the results achieved in the area of contract-based design.

IV. Service-Oriented Manufacturing (SoM)

SoM is an emerging concept in industrial automation [21], aiming at making production lines more flexible. Figure 3 summarizes the main idea behind SoM. Production recipes describe the tasks necessary to obtain the desired product. While the manufacturing tasks are usually allocated directly to machines or resources available in the facility, in SoM, each task is interpreted as a partially ordered set of actions that must be performed by the production system. Each operation that a machine can perform is considered as a service provided by the infrastructure. As such, the factory equipment and their functionalities are seen as Application Programming Interfaces (APIs), while the execution of the recipes is regarded as a sequence of primitives in the APIs. In *DeFacto*, we defined an architecture to implement such a paradigm.

The functionalities provided by each machine are accessed via an *Open Platform Communication Unified Architecture (OPC UA)* server as a service provided by the machine. The OPC UA servers are part of the Factory Communication Platform (FCP), a distributed software layer abstracting the factory and managing the communication between all the components in the system. A software component, called AMC, maps the tasks specified in the production recipes to the services provided by the infrastructure. Then, through FCP, it requests the execution of the services to the production system.

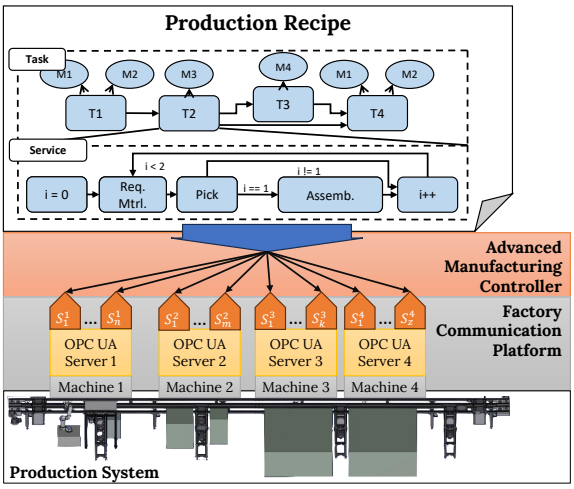

Fig. 3: Service-oriented manufacturing: the production recipes are interpreted as a partially ordered set of services to be executed by the production system. The production system is considered as a service provider. Each machine M_i implements a set of services S_n^i; the OPC UA server associated to the machine exposes the services to the FCP. The AMC maps the tasks of production recipes on the services.

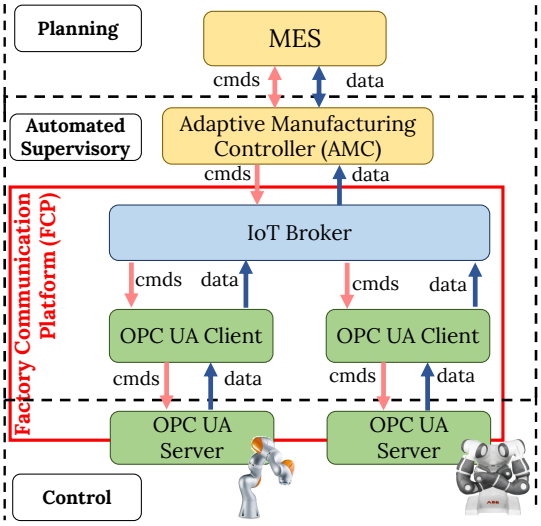

Fig. 4: General structure of the SoM-enabled software architecture. The AMC, together with the FCP, automates the supervisory level of the automation pyramid.

Since the same service may be provided by different machines in the factory, this approach decouples the execution of a task from the executor. Therefore, it increases systems flexibility.

The FCP and AMC allow our architecture to implement the SoM paradigm, while also automating the supervisory level. Figure 4 shows the complete structure of the proposed architecture [12]. From the bottom to the top, each machine is equipped with an OPC UA server exposing the services provided by the machine in a remote procedure call fashion. Furthermore, the OPC UA server collects field data. The services exposed by the OPC UA servers can be used to send commands to the machine or request data from the field. At the supervisory level, a set of OPC UA clients communicates with the AMC via a communication broker to propagate and collect the data from the field and invoke the services provided by the OPC UA servers. The AMC takes decisions on the services to be invoked, thus automating the supervisory level. The decisions are taken by analyzing the data from the field and the information coming from the upper levels, where the information about the production recipes and requirements are stored. At the planning level, we assume a MES which interfaces with the ERP system.

V. Capturing Features and Requirements: SysML

Typically, the information about a production system is spread among a large amount of documents and models, managed by different teams. In *DeFacto* we developed methods to create models describing the production and the computational aspects of a CPPS within the same SysML macro-model [10]. The *DeFacto* methodology allows reusing existing models, such as AutomationML models describing the manufacturing infrastructure and machinery, or the BPMN diagrams describing the production recipes and the manufacturing processes. Furthermore, the methodology produces models using the terminology defined in well-known industrial standards, such as the ANSI/ISA-95 standard, thus providing a common reference for all the models. The advantage of having rigorous terminology is twofold. On the one hand, models are easily interpretable by all the stakeholders. On the other hand, a rigorous terminology facilitates the development of model-based design methods and algorithms [22].

The models must be compatible with their use in the SoM architecture described above. The machines' behaviors are expressed as the set of offered services. The production recipes are expressed as the set of services necessary to obtain the desired product. To this end, the project proposes a hierarchical recipe modeling approach, increasing the granularity of the services, and therefore, the flexibility of the production planning, scheduling, and allocation operations [11], [23].

In *DeFacto*, SysML models are used for multiple purposes. Other than having a documentation purpose, they serve as the input for the automatic generation and deployment of the software components of the SoM architecture described in Section IV. As the SoM architecture governs a CPPS, the AMC exploits models to schedule and allocate the production tasks. Furthermore, models can be used to analyze the system, once requirements and system behaviors are both expressed within a formal framework, such as the one of A/G contracts.

VI. Contract-Based Design

Due to the size, heterogeneity, and distributed nature of CPPSs, their design and analysis are especially complicated [16]. In *DeFacto*, we utilized A/G contracts to formally capture heterogeneous requirements and behaviors of CPPSs, given the assumptions about their environment. Contracts provide effective mechanisms for compositionality and orthogonalization of design concerns. Specifically, we focused on three main aspects to support design and analysis of CPPSs.

Architectural exploration is a crucial step when designing and re-configuring a CPPS. However, the design space of a

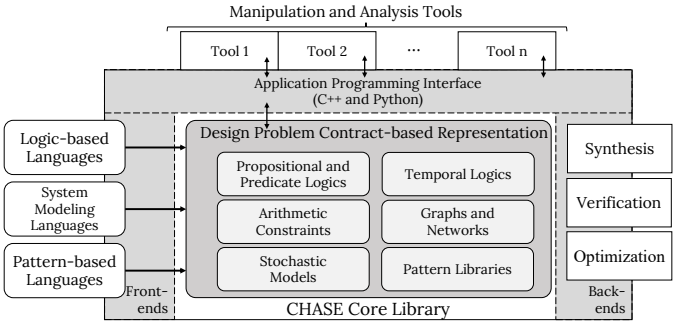

Fig. 5: Structure of the CHASE framework.

CPPS is typically heterogeneous and too large to be thoroughly explored. To address this problem, we introduced *ContrArc* [4], a contract-based methodology for the architectural exploration of CPSs. *ContrArc* offers an efficient approach for the exploration of multi-dimensional design spaces while ensuring the satisfaction of critical system requirements. It uses contracts to formally capture the system requirements and an optimization-based formulation to allocate them to the corresponding components and identify candidate architectures. By employing contract-based decomposition and a method based on subgraph isomorphism to efficiently explore the design space, *ContrArc* achieves up to two-orders-of-magnitude speed-up compared to alternative state-of-art approaches.

Even the most efficient production systems may not significantly contribute to the productivity of the overall CPPS pipeline if logistics are not properly addressed. In this context, we addressed the warehouse servicing problem (WSP) for automated warehouses, aiming to find a feasible plan for a team of autonomous agents such that a list of products can be moved from their respective starting locations to destinations within the warehouse facility. First, we addressed the problem of co-designing the warehouse topology, the scheduling algorithm, and the path planning algorithm for the set of agents, under a set of simplifying assumptions [5]. Then, we extended our approach to also account for the continuous dynamics of the agents and perform motion planning [6]. In both approaches, A/G contracts were used to model the traffic system and the workload. Task allocation and traffic plans were automatically computed using satisfiability modulo theory (SMT) solvers. Our approach allowed performing task assignment, scheduling and motion planning for hundreds of autonomous agents that could transport a million products [6]. Moreover, it outperformed alternative methodologies [24], [25] for solving the WSP by up to 2.9×.

In the architectural exploration problem and the WSP, we dealt with deterministic models of the requirements. However, in reality, CPSs must operate correctly in unpredictable or uncertain environments, and CPPSs are not an exception. For this reason, we proposed an approach for the quantitative verification and design space exploration of CPSs under uncertainty [7]. We introduced *parametric stochastic contracts* expressed in Stochastic Signal Temporal Logic (StSTL) to formally specify probabilistic behaviors of CPSs and their environments subject to uncertainty. We then proposed novel quantitative semantics of StSTL, enabling the formulation and resolution of quantitative verification and design space exploration problems as bi-level optimization problems. We illustrated the effectiveness of the proposed approach on the design of a multi-sensor perception system with several hundreds sensors and an automatic cruise control system.

The CHASE Framework: We implemented the above extensions to the theory of contracts and contract-based design within the *CHASE (Contract-based Heterogeneous Analysis and System Exploration)* framework[3] [8], [9]. As shown in Figure 5, CHASE's main component is the *Core Representation Library* which provides a set of classes for representing requirements, components, and system models as A/G contracts. Assumptions and guarantees can be modeled using a heterogeneous set of formalisms, ranging from inequalities, expressing constraints, to graphs, expressing architectural structures. The library provides functions implementing the basic operations of the contract algebra. The library is equipped with APIs, both in C++ and Python, which allow building design tools by manipulating the internal contract-based representations of a design problem. A set of front-end tools enables the automated encoding of the design problem using contracts, possibly expressed using different specification languages. A set of back-end tools interfaces with different solvers to verify certain properties, such as contract refinement, synthesize implementations, and perform optimization.

VII. Use Case: the ICE Laboratory

The ICE Laboratory is a research facility which serves as a demonstrator for a wide set of computational technologies applied to the industrial manufacturing field. Its purpose in *DeFacto* was twofold: it served as the main testbed for the project, and also as a contact point with many companies, whose requirements guided our research.

The centerpiece of the laboratory is a complete and reconfigurable manufacturing line. The production line accommodates multiple working cells: a *robotic assembly cell* made of two collaborative robots; an *additive manufacturing cell* with multiple 3D printers; a *subtractive manufacturing cell* equipped with a four-axis computer numerical control (CNC) multi-tool milling machine; a *functional control cell* for testing electronic boards; and a *vertical automated warehouse* storing the materials, semi-finished, and finished products.

The ICE laboratory logistics are managed by two autonomous ground vehicles to move objects and materials in the shop-floor, and a reconfigurable system of conveyors that moves a set of mini-pallets between the different working cells.

Factoryal engineered the SoM software architecture presented in Section IV, creating the FCP and AMC currently deployed in the ICE laboratory, which acts as the facility "operating system". Each piece of equipment is equipped with an OPC UA server, which exposes the services provided by the machine. The AMC monitors the production advancement while being aware of the production recipes, production requirements, reconfiguration policies, and other information stored in the SysML models. The AMC is also responsible

[3]CHASE open-source release: https://chase-cps.github.io/

for executing the scheduling, reconfiguration and optimization algorithms, which are used to control the production facility.

VIII. Lessons Learned

DeFacto represents our initial endeavor to incorporate system-level design and design automation principles into CPPSs. This initiative provided us with a unique opportunity to delve into the everyday needs of production companies, placing a particular emphasis on SMEs. From the outset of the project, it became evident that, despite the widespread enthusiasm for industrial trends like Industry 4.0, the actual state of production facilities frequently diverges significantly from the envisioned smart factory paradigm.

Most of the efforts spent so far by companies to enable the digital transformation consists in investments in new connected machines. Thus, the digital transformation problem is mainly approached from an industrial engineering perspective. In *DeFacto* we tried to change this perspective. We considered production systems as CPPSs, enriching them with a computer engineering perspective. This led us to embrace the SoM paradigm and develop a software architecture able to abstract the functionalities of a factory and present them as an API. In our experience, this view allows making legacy systems more flexible and extensible.

A model-based design approach is major component of the project. The increased system complexity requires rigorous methods to organize the available information. A/G contracts played a crucial role. By providing rigorous support to requirement engineering and compositional reasoning, contracts allowed us to tackle larger scale, heterogeneous system-level design problems more efficiently within a unifying framework.

Relying on graphical languages, such as SysML, to organize the information seems helpful in guiding all the stakeholders in the transition. Building models using a standardized terminology, such as the one defined in the ANSI/ISA-95 standard, allows setting a common ground to communicate with the actors managing the CPPS. Finally, organizing the information into structured models supported by formal semantics allows automating many operations required by the digital transition, such as the generation and deployment of the distributed software architectures governing the CPPSs.

Overall, the validation of the proposed methodologies, including the demonstrations in the ICE laboratory, and the proofs of concept built so far show the applicability of *DeFacto*'s concepts to real-world scenarios.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement no. 894237. The authors are grateful to all collaborators, both at the University of Verona and the University of Southern California[4], whose contributions were crucial to the success of the project.

[4]The research carried out by the collaborators at USC was also supported by the National Science Foundation (NSF) under Awards 1846524 and 2139982, the Office of Naval Research (ONR) under Award N00014-20-1-2258, the Defense Advanced Research Projects Agency (DARPA) under Award HR00112010003, the Okawa Research Grant, and Siemens via the USC Center for Autonomy and Artificial Intelligence.

References

[1] R. Drath and A. Horch, "Industrie 4.0: Hit or hype?[industry forum]," *IEEE industrial electronics magazine*, vol. 8, no. 2, pp. 56–58, 2014.

[2] Y. Koren, W. Wang, and X. Gu, "Value creation through design for scalability of reconfigurable manufacturing systems," *International Journal of Production Research*, vol. 55, no. 5, pp. 1227–1242, 2017.

[3] A. Sangiovanni-Vincentelli, "Corsi e ricorsi: The EDA story," *IEEE Solid-State Circuits Magazine*, vol. 2, no. 3, pp. 6–25, 2010.

[4] Y. Xiao, C. Oh, M. Lora, and P. Nuzzo, "Efficient Exploration of Cyber-Physical System Architectures Using Contracts and Subgraph Isomorphism," in *Proc. of IEEE/ACM DATE 2024*, pp. 1–6.

[5] C. Leet, C. Oh, M. Lora, S. Koenig, and P. Nuzzo, "Co-Design of Topology, Scheduling, and Path Planning in Automated Warehouses," in *Proc. of IEEE/ACM DATE 2023*, pp. 1–6.

[6] ——, "Task Assignment, Scheduling, and Motion Planning for Automated Warehouses for Million Product Workloads," in *Proc. of IEEE/RSJ IROS 2023*, pp. 1–6.

[7] C. Oh, M. Lora, and P. Nuzzo, "Quantitative Verification and Design Space Exploration Under Uncertainty with Parametric Stochastic Contracts," in *Proc. of IEEE/ACM ICCAD 2022*, pp. 1–9.

[8] P. Nuzzo, M. Lora, Y. A. Feldman, and A. L. Sangiovanni-Vincentelli, "CHASE: Contract-based requirement engineering for cyber-physical system design," in *Proc. of IEEE/ACM DATE 2018*, pp. 839–844.

[9] M. Lora and P. Nuzzo, "A Contract-Based Requirement Engineering Framework for the Design of Industrial Cyber-Physical Systems," in *Proc. of ACM/IEEE ICCPS 2022*, pp. 310–311.

[10] S. Spellini, S. Gaiardelli, M. Lora, and F. Fummi, "Enabling component reuse in model-based system engineering of cyber-physical production systems," in *Proc. of IEEE/EIS ETFA 2021*, pp. 1–8.

[11] S. Gaiardelli, S. Spellini, M. Lora, and F. Fummi, "A hierarchical modeling approach to improve scheduling of manufacturing processes," in *Proc. of the IEEE ISIE 2022*, pp. 226–232.

[12] S. Gaiardelli, S. Spellini, M. Panato, M. Lora, and F. Fummi, "A software architecture to control service-oriented manufacturing systems," in *Proc. of IEEE/ACM DATE 2022*, pp. 40–43.

[13] W. Babel, "Automation Pyramid and Solutions Business," in *Industry 4.0, China 2025, IoT: The Hype Around the World of Automation*. Springer, 2022, pp. 75–147.

[14] J. Leng, W. Sha, B. Wang, P. Zheng, C. Zhuang, Q. Liu, T. Wuest, D. Mourtzis, and L. Wang, "Industry 5.0: Prospect and retrospect," *Journal of Manufacturing Systems*, vol. 65, pp. 279–295, 2022.

[15] J. Amankwah-Amoah, Z. Khan, G. Wood, and G. Knight, "COVID-19 and digitalization: The great acceleration," *Journal of business research*, vol. 136, pp. 602–611, 2021.

[16] P. Nuzzo *et al.*, "A platform-based design methodology with contracts and related tools for the design of cyber-physical systems," *Proceedings of the IEEE*, vol. 103, no. 11, pp. 2104–2132, 2015.

[17] M. Obermeier, S. Braun, and B. Vogel-Heuser, "A Model-Driven Approach on Object-Oriented PLC Programming for Manufacturing Systems with Regard to Usability," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 790–800, 2015.

[18] A. Kocher *et al.*, "Model-Based Engineering of CPPS Functions and Code Generation for Skills," in *Proc. of IEEE ICPS 2022*.

[19] A. Wortmann, O. Barais, B. Combemale, and M. Wimmer, "Modeling languages in Industry 4.0: an extended systematic mapping study," *Software and Systems Modeling*, vol. 19, no. 1, pp. 67–94, Jan 2020.

[20] S. Gaiardelli, S. Spellini, M. Lora, and F. Fummi, "Modeling in Industry 5.0: What Is There and What Is Missing: Special Session 1: Languages for Industry 5.0," in *Proc. of IEEE FDL 2021*.

[21] T. Lojka, M. Bundzel, and I. Zolotová, "Service-oriented architecture and cloud manufacturing," *Acta polytechnica hungarica*, vol. 13, no. 6, pp. 25–44, 2016.

[22] S. Spellini, R. Chirico, M. Panato, M. Lora, and F. Fummi, "Virtual Prototyping a Production Line Using Assume–Guarantee Contracts," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 9, pp. 6294–6302, 2020.

[23] S. Gaiardelli, M. Lora, S. Spellini, and F. Fummi, "RRPDG: A Graph Model to enable AI-based Production Reconfiguration and Optimization," *IEEE Transactions on Industrial Informatics*, 2024.

[24] J. Li, W. Ruml, and S. Koenig, "EECBS: A Bounded-Suboptimal Search for Multi-Agent Path Finding," *The AAAI Conference on Artificial Intelligence*, vol. 35, pp. 12 353–12 362, 2021.

[25] J. Li, A. Tinka, S. Kiesel, J. W. Durham, S. T. K. Kumar, and S. Koenig, "Lifelong Multi-Agent Path Finding in Large-Scale Warehouses," in *The AAAI Conference on Artificial Intelligence*, 2021, pp. 11 272–11 281.