PERSONAL COPY - This paper has been presented at the 2021 Forum on Specification and Design Language (FDL).
The published paper is available at the following link: https://doi.org/10.1109/FDL53530.2021.9568371

Modeling in Industry 5.0: What Is There and What Is Missing

Special Session 1: Languages for Industry 5.0

Sebastiano Gaiardelli, Stefano Spellini, Michele Lora, Franco Fummi
Department of Computer Science – University of Verona
name.surname@univr.it

***Abstract*—The Industry 4.0 trend speeds up the adoption of a variety of technologies. In modern manufacturing, system data are collected both from the field through sensors and by exploiting complex simulations. Data analysis techniques became crucial to build and maintain any efficient production line, while autonomous systems and robots are the main focus of researchers and practitioners.**

This pervasive use of artificial intelligence derived technologies pushed humans to the border of production systems. Industry 5.0 aims at bringing the attention back to humans in production lines while magnifying their interactions with intelligent systems. This new trend will impact the design of future manufacturing infrastructures, increasing their complexity.

Engineers will need modeling and developing tools able to capture this complexity. In this paper, we analyze the modeling languages and tools being used, identifying their strengths and weaknesses. Then, we propose some possible directions to provide engineers with the expressive power needed to tackle the challenges posed by Industry 5.0.

***Index Terms*—Industry 5.0, Modeling Languages, Human-Robot Collaboration**

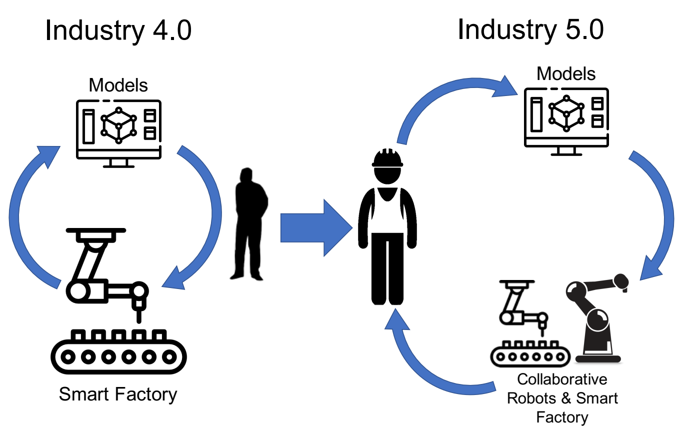

Figure 1: While in the vision proposed by Industry 4.0 the human is mainly supervising and "observing" the system, Industry 5.0 aims at putting the human back at the center of manufacturing. To pursuit such a task, models of manufacturing systems must be adjoined with models of human behaviors and, thus, novel modeling methodologies must be investigated.

I. Introduction

Industrialization underwent four main evolutions in human history, impacting on population and economic growth, and establishing important social changes. Each industrial revolution has brought to humankind new technical innovations, made possible by the acquisition of a a better understanding of the natural environment and its resources: the use of steam, fossil fuels and electrical energy has contributed to lift the burden of moving heavy and complex physical machines from animals and humans. The last iteration of such revolutions, dubbed "Industry 4.0" [1], proposes to transition from mechatronics systems to Cyber-Physical Systems (CPSs) incorporating data and intelligence, that in the context of manufacturing has been dubbed Cyber-Physical Production System (CPPS). Such systems are capable of communicating with each other, acquiring and transmitting real-time production data used to optimize production processes. This main contribution leads to increased production throughput while reducing costs and wastes. All these innovations have been made possible through the adoption of key-enabling technologies such as Internet-of-Things (IoT) and Cloud computing. Furthermore, the development of virtual models of the manufacturing system, *i.e.*, *Digital Twins*, enables the simulation of the entire production and allows performing what-if analysis.

This work is supported by the European Commission through the project DeFacto (grant n. H2020-MSCA-IF-894237), and by the Ministry of Education, University and Research "Dipartimenti di Eccellenza" 2018–2022 grant.

The pursuit for production optimality and efficiency proposed by modern manufacturing often leads to an inevitable conclusion: human labor is neither as efficient nor as cost-effective as machine labor. This is particularly true in repetitive and dull production tasks that characterize the "mass production" trend. Consequently, the cost in terms of manpower will be quite evident in the following years, when the Industry 4.0 principles will be fully implemented. Indeed, on a global scale, human labor is implemented by consumers. Thus, negatively affecting human labor may lead to decreased demand, making the investments to reach full automation not sustainable.

In this context, pushed by human resilience, the concept of Industry 5.0 [2] is emerging intending to find a sustainable trade-off between automation and human labor. As depicted in Figure 1, it proposes to put the human back at the center of manufacturing: rather than exploiting the manpower and the

PERSONAL COPY - This paper has been presented at the 2021 Forum on Specification and Design Language (FDL).
The published paper is available at the following link: https://doi.org/10.1109/FDL53530.2021.9568371

human muscles, it capitalizes on human brainpower, adding to the production loop the creativity and the problem-solving abilities that cannot be transferred to autonomous machines. In this context, nonetheless, autonomous and intelligent systems are a fundamental addition to achieving the maximum process efficiency: collaborative robots will be able to support the human during the production, by observing, learning and offering help when needed.

The complexity of current autonomous production systems has already posed new challenges for their design [3], as it requires the ability to design systems while considering multiple facets and viewpoints such as process optimization, infrastructures, control paradigms and data analysis. System design and engineering methodologies based on models, such as Model-based System Engineering (MBSE), has proved efficient in capturing some aspects of modern CPPSs. A plethora of languages and methodologies have been developed to specify and model particular viewpoints of the system, such as software [4], hardware [5] components and network infrastructures [6]. Therefore, a unifying modeling methodology or language, able to capture and include all facets of CPPSs is still missing [7].

In an Industry 5.0 context, the design of cognitive and collaborative manufacturing systems is even more complex, since the process must deal with uncertainties of humans' behaviors. Furthermore, collaborative robots are not only required to autonomously offer assistance, by recognizing the type of job the human worker is executing, but must also keep a safe working condition by not creating dangerous situations. As such, modeling such systems requires a language and a modeling environment able to capture the complexity of their possible behaviors, to perform in-depth analysis and carry out safe control strategies and AI algorithms.

This paper analyzes "what is there" about modeling tools and languages in manufacturing: we identify the modeling trends of traditional production systems, while we investigate the research work being accomplished on Industry 4.0 CPPSs. This process will clarify the main characteristics of the approaches being proposed, while determining their limitations. Furthermore, by establishing the novel modeling requirements being posed by Industry 5.0 and how collaborative systems should be designed, we define "what is missing" in this field.

II. From Industry 4.0 to Industry 5.0

The starting point of the Industry 4.0 evolution has been the so-called software automation pyramid (see Figure 2). Under this view, there is a continuous bi-directional information flow from the actual plant to the management software (ERP). Level by level, electrical signals become strings of bits, then abstract data types, then data objects and finally services. A huge variety of protocols has been created to model the exchange of information among the different levels. Few of them, are able to act as a unifying protocol managing all kinds of information exchange between all levels of the automation pyramid. The possibility of abstracting actual protocols has been the basis of Industry 4.0 since an entire production plant can become

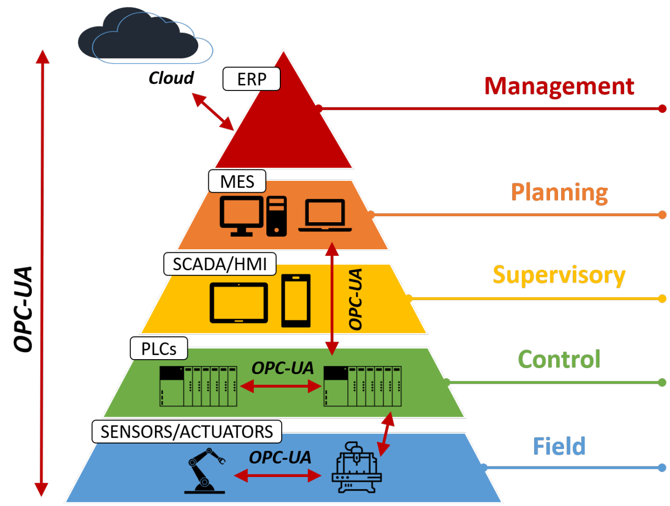

Figure 2: The automation pyramid: the starting point of Industry 4.0. Role of OPC-UA as a unifying protocol focused on data.

a service-oriented software architecture. This sensibly reduces the complexity of developing software applications managing data from the sensor/actuator level up to the cloud infrastructure. Data managing is thus becoming the core problem of a production line and Industry 4.0 focused its revolution on using such data. This allows optimizing the production, make predictions and facilitate reconfiguration, thus moving from mass production in large batches to personalized production in small and differentiated batches. Research and development in this area allowed to define protocols able to implement these requirements, among them OPC-UA [8] is one of the most successful. However, such protocols are useful to manage the heterogeneous information in the system once the system is up and running, while they are not meant to express the information of a system being designed (or redesigned). Thus, alongside protocols, models became necessary to design and implement systems able to fulfill the main design requirements of Industry 4.0 manufacturing systems. In particular, six main design requirements can be listed to implement Industry 4.0 principles:

- **Interoperability**: the ability of cyber-physical systems (*i.e.*, work assembly stations), humans and Smart Factories to connect and communicate with each other via the Internet of Things and the Internet of Services;
- **Reconfigurability**: the set of operations to produce a good (recipe) must no longer be fixed and statically scheduled, the system must adapt to the variations of its surrounding conditions and production requirements;
- **Virtualization**: a virtual copy of the Smart Factory which is created by linking sensor data (from monitoring physical processes) with virtual plant models and simulation models;
- **Decentralization**: the ability of cyber-physical systems within Smart Factories to make decisions on their own Real-Time Capability: the capability to collect and ana-

lyze data and provide the insights immediately;
- **Service Orientation**: offering of services (of cyber-physical systems, humans and Smart Factories) via the Internet of Services;
- **Modularity**: flexible adaptation of Smart Factories for changing requirements of individual modules.

The core (a bit naïf) assumption of Industry 4.0 is that the production is completely automatic, thus the role of humans is limited to control and supervision. However, this is not feasible in the majority of production lines, particularly because flexibility, adaptation and precision of robotic activities are still far away from human abilities. People must still play a productive role in an Industry 5.0 production line.

However, differently from the third industrial era, Industry 5.0 promotes a significant change in the adaptation philosophy: people have not to adapt their behavior to the machines, but **machines must automatically adapt their activities to the human ones**. Thus, there is a new list of design requirements to implement this new Industry 5.0 principle:

- **Uncertainty**: this is related to the unpredictability of the human behaviors that must be represented under some level of uncertainty. Indeed, uncertainty impacts how interoperability and reconfigurability are implemented by Industry 5.0 systems, as subsystems must reconfigure and interoperate by considering that other subsystems may be humans. Facing the non-determinism becomes crucial and this must be taken into consideration by all languages, models and theories;
- **Cognitivity**: an evolved MES is necessary to control the production line; it cannot be based on simple deterministic algorithms, but it has to implement cognitive methods that must be continuously reinforced by the experience; in other words, AI must become the core technology to program this kind of architectures in order to allow the system to co-exist with natural intelligence of human agents;
- **Safety**: a higher level of safety must be reached to guarantee the cooperation between humans and robots, since all operations scheduled by a robot must be checked for safety before their execution by considering the current and the possibly predictable human behaviors.

Indeed, these new requirements require a technological leap, as well as an advancement of the tools necessary to support the design of the new technologies that must be supported by novel modeling and design languages and methods. The next sections analyze how to fulfill these requirements with the current modeling languages and what is missing.

III. What is there: Modeling Industrial Systems

Traditional industrial systems focused on the automation of production processes. Manufacturing systems relied on novel technologies such as Programmable Logic Controllers (PLCs). Nonetheless, systems still had to reckon on human inputs, that manually parametrized the process according to production requirements. Therefore, since 60% of all factory automation costs were caused by engineering and commissioning [9], research interests were centered on aiding the engineering and the implementation of automation components (*i.e.*, robots) in production lines.

A. Modeling the system

One of the main products of this research is Automation Markup Language (AutomationML): a language developed to assist engineers and reduce costs. It is an XML-based data format, created to provide a formal exchange model for heterogeneous engineering tools [10]. In industrial contexts, it is capable to describe plant components from different points of view, from the plant topology to a machine's kinematics. Different standards are strongly intertwined within AutomationML to compose a complete description of the system to be characterized. In particular, the Computer Aided Engineering Exchange (CAEX) (IEC 62424) provides features to represent a topological view of the system, with relations between objects.

CAEX has an object-oriented structure. It is composed of a set of standard and user-defined libraries and object instances in a precise structure. The first library is the *RoleClasses* library, which is the abstract functionality representation of an object, without technical details on its implementation. As an example, a "resource" is a role for an object, and can be further detailed to represent a piece of equipment or material. As such, role classes define objects semantics. The *SystemUnitClasses* library contains vendor-specific AutomationML objects, typically representing particular resources instances with specific characteristics. The *InterfaceClasses* library serves to define the relations between objects, both syntactically and semantically. AutomationML's core is the *InstanceHierarchy*, where objects instances are stored in a well-formed hierarchy of elements and sub-elements. To model the system's behaviors, the COLLAborative Design Activity (COLLADA) interchange standard provides primitives to specify information about kinematics and three-dimensional shapes and, thus, on the mechanical nature of the system. Such a standard is integrated into AutomationML via XML external links. PLCopen is another standard embedded within AutomationML. It allows describing the logical behavior of the system and its components, by specifying impulse diagrams, sequence function charts, logic networks, state charts, Gantt and Pert charts.

B. Modeling the processes

When modeling material and information flows during production, a well-established pencil and paper practice is Value Stream Mapping (VSM) [11]. This modeling methodology provides a better understanding of the product value chain and, therefore, of the production process. VSM, however, does not provide an executable digital model and, thus, stakeholders cannot integrate the knowledge provided by the model in IT systems. Therefore, a language that has been created to solve this issue is Business Process Model and Notation (BPMN). Its main advantage is to support the modeling of business processes by specifying a syntax and an executable semantics. Furthermore, many extensions of such a language have been proposed to represent different aspects of processes and

PERSONAL COPY - This paper has been presented at the 2021 Forum on Specification and Design Language (FDL).
The published paper is available at the following link: https://doi.org/10.1109/FDL53530.2021.9568371

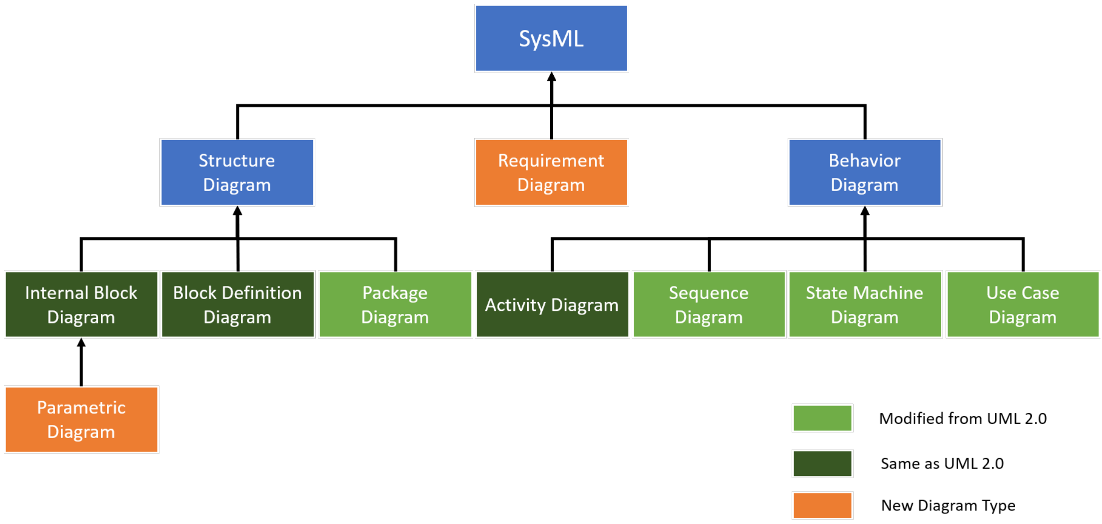

Figure 3: SysML diagrams hierarchy, clarifying the diagram types that are adopted or adapted from Unified Modeling Language (UML).

various viewpoints of manufacturing. Among all, a mapping between VSM and BPMN has been proposed in [12].

To regulate the knowledge integrated into manufacturing information systems, different standards have been created. The most relevant business standard is International Society of Automation (ISA)-95, also known as IEC 62264. Its main contribution is to define the terminology for business to manufacturing integration, allowing the unification of all the software information knowledge within the manufacturing industry. ISA-95 is still under development, and mainly consists of 5 parts:

- Part1: Models and terminology
- Part2: Object model attributes
- Part3: Activity models and attributes of manufacturing operation management
- Part4: Object models and attributes of manufacturing operations management
- Part5: Business to Manufacturing transactions

Part 1 defines the hierarchy of terminologies to define interfaces between manufacture and business processes. Part 2 defines the objects and attributes that can be used in the information exchange between manufacture and business processes. Part 3 defines the main manufacturing functions and activities, such as production, maintenance, warehousing and quality control. It also introduces the concept of Manufacturing Execution System (MES). This piece of software is in charge of allowing bidirectional communication between business software, such as Enterprise Resource Planning (ERP), and manufacturing software, such as PLCs and SCADA. The ISA-95 standard defines also the activities that a typical MES must carry out [13]:

- **Production definition management**: manages all information about the product required for manufacturing;
- **Production resource management**: manages the information about resources required by production operations;
- **Detailed production scheduling**: includes local planning and scheduling of production and resources;
- **Production dispatching**: manages the flow of production by dispatching production to equipment and personnel;
- **Production execution**: directs the performance of work, as specified by the contents of the dispatch list elements;
- **Production data collection**: collects and manages process and equipment information;
- **Production tracking**: prepares the production response for the ERP;
- **Production performance analysis**: provides feedback about production.

These activities create a complete vision of what is happening within the manufacturing industries. A MES that is compliant with the standard can efficiently communicate with other compliant software. Furthermore, by uniquely defining terms and objects, the standard cuts down the necessary effort to model the manufacturing system. The construction of an exhaustive model of the system, from business to control automation viewpoints, conceives a complete vision of the production line, enabling model analysis and design optimization. Therefore, ISA-95 has been integrated in AutomationML [14], by defining particular AutomationML libraries and classes to represent the concepts proposed by the standard. In particular, interesting facets integrated into AutomationML formally detail types of information, such as the *Product definition* model, which describes processes and requirements to make a product, or the *Resource definition* model, which characterizes available resources such as pieces of equipment, materials and also personnel.

IV. What is there: Modeling Advanced Manufacturing

A major effect of the Industry 4.0 trend is the transformation of manufacturing systems into CPPSs. Thus, modeling of advanced manufacturing systems borrows most of the requirements identified when modeling CPS. Models must be able

PERSONAL COPY - This paper has been presented at the 2021 Forum on Specification and Design Language (FDL).
The published paper is available at the following link: https://doi.org/10.1109/FDL53530.2021.9568371

to capture widely heterogeneous components [15], dynamics and behaviors [16] as well as a large variety of different viewpoints [17]. Modeling tools and languages must be able to both represent the system's architecture and behaviors, and also depict the decision process carrying out the reconfiguration processes. Production data analysis is another key ingredient for the production processes optimization proposed by smart manufacturing principles. As such, modeling engineering technologies and languages must be paired with knowledge-representation languages to reach the necessary expressive power for advanced manufacturing systems.

For software and systems model engineering, one of the most used languages is SysML. As depicted in Figure 3, it provides a set of diagrams over UML, to represent systems and systems-of-systems in addition to plain software. As such, it is natively capable of representing manufacturing systems and expressive enough to enable performing analysis over models. Other than native SysML, specializations have been proposed [18] to aid the development of automation software (*i.e.*, PLC software) for smart manufacturing systems. Furthermore, models can be used to automatically generate control software to directly integrate into machines. SysML has also been used to ease the development and the integration of a MES in a production line [19]. AutomationML is also a widely used language for engineering. As an example, AutomationML models can be used for the data exchange between CPPSs engineering viewpoints. Such a language provides the necessary information for a tool to analyze AutomationML models for data analytics on engineering activities [20]. This tool would be particularly useful for engineers because it allows them to browse the interlinked models and to query ongoing project activities. To support the modeling of manufacturing activities on a non-technological level, process modeling languages combine graphical clarity with a well-known specification meta-model. Other than BPMN, semi-formal process modeling language has been proposed [21], able to capture the business aspects of Industry 4.0-compliant systems, such as IoT components. Such methodologies can be used for model validation and simulation, as well as to generate code for software Application Programming Interfaces (APIs). The combination of automated control and business aspects is also an important step to achieve a complete view of production activities. Models combining such viewpoints integrate a wider set of information and, thus, can be more finely verified and simulated. The *design by contract* is a relevant approach for the construction of CPSs [22], [23]. Such a principle has been recently proposed also for the design of CPPSs [24]. To construct contract-based specifications, it is necessary to collect system's information by manipulating state-of-the-practice languages (*i.e.*, AutomationML and ISA-95) and, thus, models of the production plant. Assume-Guarantee (A/G) contracts can be then checked for consistency, composed and synthesized to construct correct control software that represents the behavior of the specified manufacturing component.

Representing and formalizing the knowledge of entities and services involved in manufacturing is also an open challenge. In this regard, the most widely used tool is ontology, which is defined as a formal and explicit specification of a shared conceptualization [25]. Semantic models and ontologies have been used to specify concepts, resources and entities involved in manufacturing, especially production information systems (*i.e.*, MES) [26]. The models serve as logical definitions that can be exploited to infer concept relations between entities, such as between enterprise, automation and logistics domains [27]. For those reasons, different standards and languages have been proposed to carry out information modeling through ontologies. Different languages exist to carry out such a task. The most representative ones are the Web Ontology Language (OWL) and the Semantic Web Rule Language (SWRL). Those languages have been used to define semantic annotations of service-oriented architecture, created to control manufacturing systems based on the type of service it is required to provide [28] [29]. The integration of ontologies and, thus, semantic reasoning within SysML has been proposed in the past [30]. Nonetheless, an ontology can be considered as a meta-metamodel, which is often specified by creating custom SysML profiles. Consequently, the knowledge defined in the ontology is transferred to the model, and their interoperation is guaranteed.

A systematic review [7] of modeling languages for Industry 4.0 systems highlighted that the effort of involved research groups is rather sparse: AutomationML and the System Modeling Language (SysML) are popular languages in the field, but they are rarely coupled together and, thus, a complete modeling methodology is still missing. In particular, the analyzed trends show that interoperation between languages and models is still an open issue. In fact, the substantial ability to carry out the design of a complex manufacturing system reusing models of components or system viewpoints is still pursued. In this regard, SysML seems to be the appropriate language, expressive enough to serve as a collector of information and modeling viewpoints. On this matter, a design flow that goes in this direction has been recently proposed [31]. It contemplates the reuse of already existing manufacturing lines' models while designing novel advanced production systems. As depicted in Figure 4, it is embedded in Platform-Based Design (PBD) approach, where bottom-up reused models are paired with top-down specified requirements and functionalities. The methodology is actually limited to reusing AutomationML models, incorporating information about plant topology and components hierarchy. Nonetheless, SysML is proposed as the central container of the system's models and information, which can be then exploited to produce software or carry out model analysis and design exploration iterations.

V. Modeling Industry 5.0: what is missing and possible directions

Industry 5.0 is still in its infancy. To the best of our knowledge, no research effort exists today in proposing modeling tools or methodologies for this trend. In particular, modeling CPSs with humans in the loop is still an open task [32], [33].

As stated in Sec. II, one of the main requirements of Industry 5.0 is reconfigurability: it involves guaranteeing a

PERSONAL COPY - This paper has been presented at the 2021 Forum on Specification and Design Language (FDL).
The published paper is available at the following link: https://doi.org/10.1109/FDL53530.2021.9568371

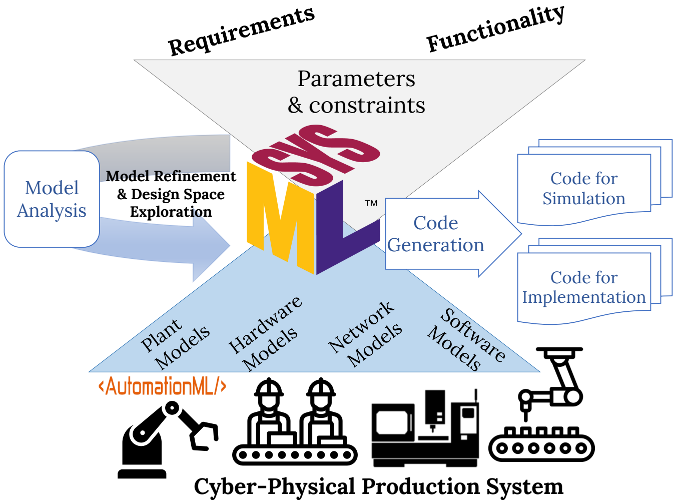

Figure 4: Conceptual view of the design flow for CPPSs proposed in [31], where the model reuse methodology is represented by the cyan triangle.

certain resiliency and adaptation degree of the CPPS to real-time changes in the environment. In this regard, to carry out the best reconfiguration strategy, it would be fundamental to have models spanning over business, processes and control viewpoints: integrating the widest possible set of information increases the quality of the control strategy carried out by the reconfiguration procedure. By also considering humans in the manufacturing loop, a degree of complexity is added on top, since it involves modeling nondeterministic behaviors. Of course, models are not enough in this case to represent all the possible behaviors. It is necessary to employ cognitive methods, which require experimenting with AI techniques to carry out control strategies based on observations. Nonetheless, models can aid the "learning" procedures of AI algorithms, by providing a basic set of information that can be helpful to interpret the current system status. This is particularly true also for safety considerations: the cooperation between humans and robots can be dangerous for human operators. Therefore, possible critical situations must be predicted and avoided. Different techniques can be exploited, most of them based on pattern recognition and AI-based prediction using different types of learning techniques. In all these scenarios, models can aid the process of carrying out the best possible strategy. These techniques introduce in the system different levels of uncertainties, making systems non-deterministic. While multiple mathematical frameworks allow reasoning on non-deterministic systems, no system modeling language at the state of the art provides the designer the tools necessary to specify such systems efficiently.

To support the engineering of Industry 5.0 systems, engineers must be put in the condition to comfortably specify non-deterministic behaviors intrinsic to humans' behaviors, and those introduced when using AI techniques. As such, future system modeling language must allow specifying uncertainties and their semantics must be built on top of probabilistic and stochastic mathematical formalisms. Furthermore, to be effective for system design, they must allow designers to select the appropriate degree of abstraction and is capable of scaling on the level of details.

Reusing models is also a winning strategy in the context of modeling complex manufacturing systems: as stated in Sec. IV, it allows to cut down the modeling effort in a multi-faceted scenario. SysML is a suitable language because it is expressive and it is extensible with profiles and stereotypes. Furthermore, it is XMI-based and, thus, it can be manipulated and translated to other languages (*i.e.*, formal languages, programming languages, etc.).

Figure 5 depicts the set of Industry 5.0 characterizing features introduced in Sec. II, *i.e.*, *reconfigurability*, *uncertainty*, *cognitivity*, and *safety*. The figure relates each feature with the mathematical formalisms required to represent them.

a) Uncertainty: strongly impacts on the interoperability and the riconfigurality of the system. The system must be able to interoperate with non-epistemic agents (*i.e.*, human beings), that may potentially act irrationally or through apparently unexplicable paths. Thus, machines may interpret and predict the behavior of these agents only through models contemplating probabilistic, as well as statistical behaviors. At the same time, uncertainties in the environment surrounding the machines may lead to the necessity of system adaptation, *i.e.*, reconfiguration. This implicates carrying out optimization procedures to maximize the effectiveness of the adaptation. Furthermore, system reconfiguration may still need to consider future possible uncertainties that may be caused by the intervention of human agents, as well as by external causes. As such, stochastic models used to represent the uncertainty of the system may act as triggers for the models specifying the system reconfiguration.

Thus, modeling uncertainty in the context of Industry 5.0 requires the ability of capturing stochastic systems formalisms, which typically involve representations based on probabilistic and statistical methods, as well as optimization specifications. Formal techniques able to capture, within the same framework both optimization and statistical models are gaining maturity [34]–[36]. However, a language capable of modeling these aspects within the same framework, while providing an intuitive syntax and semantics to designers has yet to be proposed.

b) Cognitivity: is not only associated with stochasticity but also to system's dynamics. In fact, the cognition process must implement techniques dedicated to recognizing patterns of the dynamical behavior of a certain component or set of components. On the other hand, recognizing the role of the human actor in Human-Robot Collaboration (HRC) is also a key component of Industry 5.0 systems. Such a process involves determining a set of tasks that the human is capable of carrying out in the manufacturing process. It is also necessary to estimate his/her performances, to evaluate whether the robot's cooperation would be beneficial to the overall production. In HRC tasks, the decision-making procedures of robots must deal with the mental state of the human as well as the sense of trust he/she has in the robotic collaborator.

PERSONAL COPY - This paper has been presented at the 2021 Forum on Specification and Design Language (FDL).
The published paper is available at the following link: https://doi.org/10.1109/FDL53530.2021.9568371

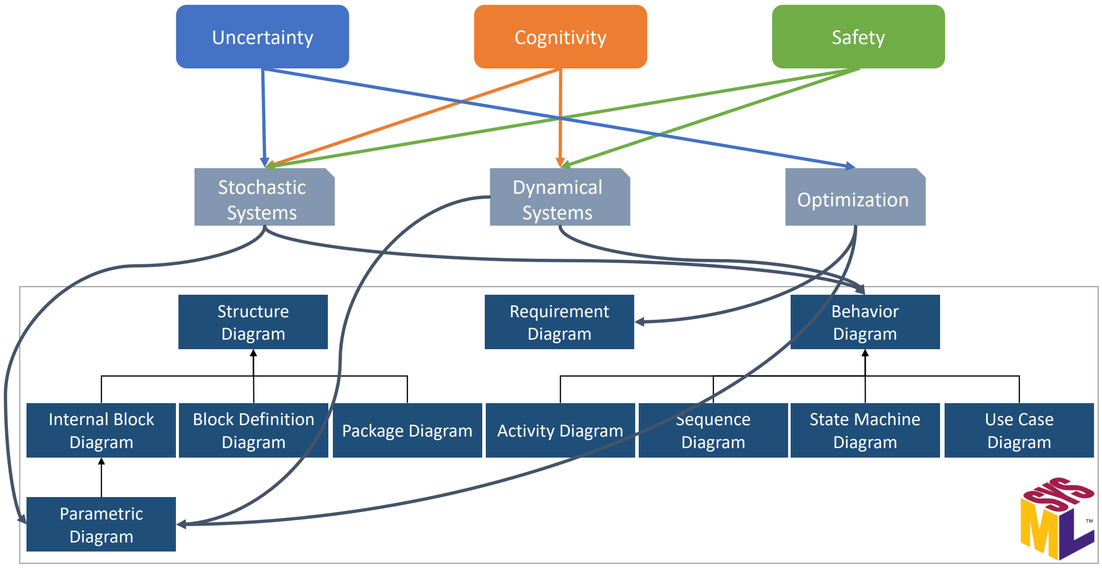

Figure 5: Relations between Industry 5.0 requirements asserted in Sec. II, systems types and possible associations with SysML diagrams.

Such collaboration facets must exploit inference models and reinforcement learning, to correctly perceive the human's status and intentions, dealing also with uncertainty. Furthermore, it is also necessary to strengthen communication models, to provide convincing decisions explanations to human operators.

Modeling cognitive-aware systems, thus, requires methodologies and languages able to capture multiple facets of human behaviors and emotional processes. Mathematical models have been proposed to guide the decision process and control synthesis of HRC, exploiting non-learning techniques such as optimization based on Markov Chains [37] and learning techniques such as Reinforcement Learning [38]. Nonetheless, an inclusive framework for the modeling and specification of cognitive-aware systems has yet to be proposed, limiting the applicability of such techniques to domain experts.

c) Safety: is a feature associated with both stochastic and dynamical systems. In Human-Robot interactions and collaborations, safety can be related to avoiding both physical and psychological harm. Different categories of methodologies have been developed to provide safety in HRC environments [39]. *Safety through control* proposes methods to prevent unwanted contacts between the human and the robot by detecting at run-time unwanted system's states and reacting to recover from a dangerous situation. *Safety through motion planning* category proposes more proactive approaches suggesting human-aware motion planning techniques exploiting dynamical models. Such models exploit differential equations and, thus, a strong mathematical foundation to define the kinematics of the robotic systems. In methodologies related to *Safety through prediction*, the motion planner is adjoined with predicted human behaviors and movements, to proactively propose movements instead of continuously replanning. *Safety through psychological considerations* is more focused on extra-functional motion properties, *i.e.*, acceleration, velocity, distance from the human, etc, since they are the most influential parameters on human's well-being.

In general, safety requires setting the boundaries of unwanted behaviors. At the state of the art, the specification of such requirement still relies mostly on complex mathematical notations, based on complex logic, automata, and equations [40]. The use of formal mathematical frameworks is imposed by the critical importance of the concepts being expressed. However, this makes the life of engineers harder. Some attempts of proposing domain specification languages able to simplify the specification task, while still guaranteeing strong formal support, have been already presented [23], [41]. However, no general designer-friendly language has been proposed so far, able to intuitively and effectively capture different types of safety requirements, while providing the formalisation required to tackle safety issues.

A. SysML support to Industry 5.0 models

While many features of Industry 4.0 may be represented by using SysML and other state-of-the-art modeling and description languages, this is not the case of industry 5.0. Neither SysML nor other modeling languages at the state-of-the-art provide the necessary language constructs to specify every type of Industry 5.0 system. Focusing on SysML, among the different diagrams, it implements the parametric diagram type, that can capture, to a certain degree, the dynamics of a system. However, such a type of diagram is not able to represent, for example, probability distributions or stochastic equations for numerical approximations. As such, the language has to be extended to design such systems. To optimize a system according to the system's requirements is a process involving, as an example, linear programming techniques and an optimization model. Such a model should be designed with the support of SysML diagrams, such as the requirements diagram. However, in requirements diagrams, requirements are specified only in natural language. Therefore, a methodology

PERSONAL COPY - This paper has been presented at the 2021 Forum on Specification and Design Language (FDL).
The published paper is available at the following link: https://doi.org/10.1109/FDL53530.2021.9568371

to extend the expressiveness of SysML able to include formal specifications is still missing.

Thus, a research effort is necessary to extend the existing modeling and specification languages, and to define new ones as well, able to effectively support engineers. This effort should move toward easier ways of specifying models which are supported by stochastic, dynamical and optimization mathematical frameworks underneath, to allow the designer to better tackle reconfigurability, uncertainty, cognitivity and safety requirements of Industry 5.0 systems.

VI. Conclusion

The complexity of Industry 5.0 systems poses new challenges in their design. A unifying modeling methodology, able to thoroughly capture all the aspects and viewpoints does not exist. Therefore, this paper analyzed research trends and state-of-the-art methodologies to model Industry 4.0 systems. Methodologies based on SysML prove that such a language has the appropriate degree of expressiveness to model complex CPPSs. Finally, this paper proposes future directions to model Industry 5.0 systems.

References

[1] M. Hermann *et al.*, "Design principles for industrie 4.0 scenarios," in *Proc. of HICSS*, 2016, pp. 3928–3937.
[2] S. Nahavandi, "Industry 5.0—a human-centric solution," *Sustainability*, vol. 11, no. 16, p. 4371, 2019.
[3] L. Ribeiro, "Cyber-Physical Production Systems' Design Challenges," in *Proc. of IEEE ISIE*, 2017, pp. 1189–1194.
[4] E. Korshunova *et al.*, "CPP2XMI: Reverse engineering of UML class, sequence, and activity diagrams from C++ source code," in *2006 13th Working Conference on Reverse Engineering*, 2006, pp. 297–298.
[5] N. Bombieri *et al.*, "On the reuse of heterogeneous IPs into SysML models for integration validation," *Journal of Electronic Testing*, vol. 29, no. 5, pp. 647–667, 2013.
[6] E. Ebeid *et al.*, "Model-driven design of network aspects of distributed embedded systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 4, pp. 603–614, 2015.
[7] A. Wortmann *et al.*, "Modeling languages in Industry 4.0: an extended systematic mapping study," *Software and Systems Modeling*, vol. 19, no. 1, pp. 67–94, 2020.
[8] OPC Foundation, "OPC Unified Architecture." [Online]. Available: https://opcfoundation.org/developer-tools/specifications-unified-architecture
[9] AutomationML Consortium, "AutomationML - story," 2021. [Online]. Available: https://www.automationml.org/o.red.c/story.html
[10] R. Drath, "Let's talk AutomationML what is the effort of AutomationML programming?" in *Proc. of IEEE ETFA*, Sep. 2012, pp. 1–8.
[11] M. Rother, *Learning to See: Value Stream Mapping to Create Value and Eliminate Muda*, 06 1999.
[12] S. Zor *et al.*, "Using bpmn for modeling manufacturing processes," 2010.
[13] International Society of Automation, "ISA-95 Standard," 2000. [Online]. Available: https://www.isa.org/
[14] B. Wally, "Application Recommendation Provisioning for MES and ERP – Support for IEC 62264 and B2MML," 2018.
[15] M. Lora, S. Vinco, and F. Fummi, "Translation, Abstraction and Integration for Effective Smart System Design," *IEEE Transactions on Computers*, vol. 68, no. 10, pp. 1525–1538, 2019.
[16] P. Derler, E. A. Lee, and A. S. Vincentelli, "Modeling cyber–physical systems," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 13–28, 2011.
[17] D. Broman, E. A. Lee, S. Tripakis, and M. Törngren, "Viewpoints, formalisms, languages, and tools for cyber-physical systems," in *Procs. of MPM*, 2012, pp. 49–54.
[18] B. Vogel-Heuser *et al.*, "Model-driven engineering of manufacturing automation software projects – a sysml-based approach," *Mechatronics*, vol. 24, no. 7, pp. 883 – 897, 2014, 1. Model-Based Mechatronic System Design 2. Model Based Engineering.
[19] L. Piétrac *et al.*, "On the use of sysml for manufacturing execution system design," in *ETFA2011*, 2011, pp. 1–8.
[20] M. Sabou *et al.*, "Supporting the engineering of cyber-physical production systems with the automationml analyzer," in *2016 1st International Workshop on CPPS*, 2016, pp. 1–8.
[21] R. Petrasch and R. Hentschke, "Process modeling for industry 4.0 applications: Towards an industry 4.0 process modeling language and method," in *2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 2016, pp. 1–5.
[22] 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.
[23] P. Nuzzo, M. Lora, Y. Feldman, and A. Sangiovanni-Vincentelli, "CHASE: Contract-based requirement engineering for cyber-physical system design," in *Proc. of IEEE/ACM DATE*, 2018, pp. 839–844.
[24] S. Spellini *et al.*, "Production Recipe Validation through Formalization and Digital Twin Generation," in *Proc. of IEEE/ACM DATE 2020*.
[25] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge Acquisition*, vol. 5, no. 2, pp. 199–220, 1993.
[26] S. Jaskó *et al.*, "Development of manufacturing execution systems in accordance with industry 4.0 requirements: A review of standard- and ontology-based methodologies and tools," *Computers in Industry*, vol. 123, p. 103300, 2020.
[27] O. Harcuba and P. Vrba, "Ontologies for flexible production systems," in *2015 IEEE 20th Conference on Emerging Technologies Factory Automation (ETFA)*, 2015, pp. 1–8.
[28] E. Negri *et al.*, "Requirements and languages for the semantic representation of manufacturing systems," *Computers in Industry*, vol. 81, pp. 55–66, 2016.
[29] C. Hildebrandt *et al.*, "Reasoning on engineering knowledge: Applications and desired features," in *The Semantic Web*. Cham: Springer International Publishing, 2017, pp. 65–78.
[30] S. Feldmann *et al.*, "Combining a sysml-based modeling approach and semantic technologies for analyzing change influences in manufacturing plant models," *Procedia CIRP*, vol. 17, pp. 451–456, 2014, variety Management in Manufacturing.
[31] S. Spellini *et al.*, "Enabling Component Reuse in Model-based System Engineering of Cyber-Physical Production Systems," in *2021 IEEE 26th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2021.
[32] C. Raymond and D. Prun, "Extending mbse methodology and sysml formalism to integrate human considerations," in *Proc. of HCI-Aero*, 2016, pp. 1–4.
[33] Z. Liu and J. Wang, "Human-cyber-physical systems: concepts, challenges, and research opportunities," *Frontiers Inf. Technol. Electron. Eng.*, vol. 21, pp. 1535–1553, 2020.
[34] Y. Shoukry *et al.*, "SMC: Satisfiability modulo convex programming," *Proceedings of the IEEE*, vol. 106, no. 9, pp. 1655–1679, 2018.
[35] P. Nuzzo, J. Li, A. L. Sangiovanni-Vincentelli, Y. Xi, and D. Li, "Stochastic assume-guarantee contracts for cyber-physical system design," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 18, no. 1, pp. 1–26, 2019.
[36] C. Oh, E. Kang, S. Shiraishi, and P. Nuzzo, "Optimizing assume-guarantee contracts for cyber-physical system design," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2019, pp. 246–251.
[37] M. Chen *et al.*, "Trust-aware decision making for human-robot collaboration: Model learning and planning," *J. Hum.-Robot Interact.*, vol. 9, no. 2, 2020.
[38] L. Roveda *et al.*, "Model-based reinforcement learning variable impedance control for human-robot collaboration," *Journal of Intelligent and Robotic Systems*, 11 2020.
[39] P. A. Lasota *et al.*, *A Survey of Methods for Safe Human-Robot Interaction*, 2017.
[40] A. Platzer, *Logical analysis of hybrid systems: proving theorems for complex dynamics*. Springer Science & Business Media, 2010.
[41] A. Pinto and A. L. Sangiovanni Vincentelli, "Csl4p: A contract specification language for platforms," *Systems Engineering*, vol. 20, no. 3, pp. 220–234, 2017.