Guest Editorial
Industry 4.0–Prerequisites and Visions

MOST design principles and enabling technologies of Industry 4.0 have been an active area of research for five to ten years now, but provide in their combination a visionary concept of self-aware, cooperating Cyber Physical Production Systems. From its origin, Industry 4.0–derived from the German term Industrie 4.0–is used as a synonym for Cyber-Physical Production Systems (CPPS), i.e., Cyber-Physical Systems applied in the domain of manufacturing/production. To enable CPPS their automation systems need to be enabled to fulfill the requirements. The term Industrie 4.0 was first used in 2011 at the Hannover Fair and the topic has grown every year not only on the fair. There are still several definitions of Industrie 4.0 (I4.0). Most of them agree on the following design principles [1]:

- Service Orientation: CPPS offering services via the Internet based on a service oriented reference architecture.
- Intelligent self-organizing CPPS providing:
  - the ability of CPPS to make decisions on their own (decentralization).
- The ability of CPS, humans and CPPS to connect and communicate with each other (interoperability):
  - information aggregation and representation for the human in the loop during engineering and maintenance of aPS;
  - a virtual copy of CPPS on different levels of detail, e.g., from sensors and actuators to the entire CPPS (virtualization);
  - relevant process and engineering information for data analysis (real time capability);
- The ability to flexible adaptation to changing requirements by replacing or expanding individual modules (cross-disciplinary modularity).
- Big Data algorithm and technologies provided in real-time (real-time capability).
- Optimization of the manufacturing process based on these algorithms and data to increase Overall Equipment Effectiveness (OEE).
- Data integration cross disciplines and along the life cycle based on standardized data models and a model driven modular engineering process.
- Secure communication enabling a worldwide network of aPS supporting economic industrial partnership across companies borders.
- Access to data securely stored in a Cloud/Intranet.

The economic impact of the 4$^{\text{th}}$ industrial evolution (in the U.S. especially artificial intelligence, robotics, and 3D printing) was also discussed during the world economic forum 2016 [2]. Governments worldwide support of the development regarding I4.0 with scientific grants economic and development schemes. In its beginning, I4.0 has been evolved in Germany initially by three associations, i.e., BITKOM representing IT companies, VDMA representing machine and plant manufacturers, and ZVEI representing suppliers of electric and electronic devices and is strongly supported by the German government. Since June 3, 2015, the platform I4.0 coordinates these activities and is strongly supported by the Federal Ministry for Economic Affairs and Energy and the Federal Ministry of Education and Research. By now, 208 application examples (among others, the *MyJoghurt* demonstrator operated by the Technical University of Munich) as well as 28 test and competence centers are affiliated in the roadmap I4.0. The Reference Architecture Model (RAMI 4.0) introduces three dimensions: *Layer*, *Life Cycle*, and *Value Stream* as well as *Hierarchy Levels* in April 2015. RAMI names real-time capability, reliability, and the possession of the required QoS-attributes as characteristics for I4.0.

Research on most I4.0 design principles and enabling technologies mentioned above delivered already promising results for single aspects, but the challenge and the benefit will be only reached combining all aspects synergistically and considering the constraints from automation especially automated production systems as real-time, dependable, safety standard compliant systems providing concepts for diagnosis and fault handling.

Three important challenges in automation of CPPS will be discussed exemplarily: first, modularity and interfaces of control software as a prerequisite for adaptation to changing requirements; second, modularity of the mechatronic system managing and identifying inconsistencies with semantic technologies; and third, real-time capabilities to collect and analyze process data. At first, modularity application software and as a prerequisite adaptable interfaces will be discussed focusing on software. aPS are long living systems (up to $\sim$ 30 years) and have up to now mostly limited computing capacity. Common Programmable Logic Controllers (PLC) and industrially accepted programming languages lack in object oriented mechanisms [3]. Additionally, often software engineering is the last activity after mechanical and electrical design facing a lack of information and limited development time because of delays in the other disciplines. On the other hand, bugs created in other disciplines need to be fixed by means of software on-site. As a consequence, software engineering for aPS is still struggling with a multitude of challenges.

- Transition to modularity and maintainable interfaces are a fundamental basis for adaptable and evolvable systems.

Digital Object Identifier 10.1109/TASE.2016.2523639

- Tracking of changes in hundreds of machines or plants on different operation sites (mostly distributed globally and often optimized by operating companies) operated over decades.
- Management of consistent software variants and versions, especially if we additionally assume to include self-adaptation and reconfiguration during runtime.
- Adaptation of Big Data algorithm and technologies already well known for fleet management related to consumer products for aPS.

As a basis for a more efficient management of variants and versions during operation machine and plant suppliers start to improve their code structure intending automatic code configuration based on optimized module libraries. That allows to reuse already existing and tested code and, subsequently, to configure the application software instead of programming it. Evaluation in pilot projects show that up 70% until 80% of the code may be configured automatically out of engineering information and module libraries. This gain in efficiency compensates easily the efforts for optimizing and maintaining the module libraries.

Addressing the second challenge we widen our focus from software to modularity of the mechatronic system; the integration of engineering data cross-disciplinary and along the life cycle. Flexible adaptation to changing requirements by replacing or expanding individual modules (modularity) requires to identify and manage inconsistencies.

Semantic Web technologies [1], [5] are promising technologies to handle this challenge. They are beneficial identifying and managing inconsistencies in the following use cases.

— Structural compatibility check between mechatronic modules after a change based on a system model, e.g., in Systems Modeling Language (SysML) combined with the Web Ontology Language (OWL).

— Consistency between models along the engineering life cycle of aPS, e.g., between requirement and test case [5] will be reached by inconsistency check of the respective attributes.

— Inconsistencies between interdisciplinary engineering models of aPS by evaluation of, e.g., attribute types' equivalence and furthermore resolving of such inconsistencies [1].

However, as a basis for inconsistency management at first appropriate and accepted vocabularies are required describing the semantics of the attributes, e.g., like NIST function definition [6]. One enabler to implement such concepts is the availability of a multitude of so-called triple stores that support Semantic Web technologies and provide higher flexibility than traditional relational databases. After a replacement of an electrical part in an operating aPS this change needs to be analyzed regarding potential inconsistencies to the mechanical or software part, e.g., whether the newly integrated device fulfills the interface of the replaced one, e.g., analogue output (port, query 4, Fig. 1) and a maximum value, e.g., current of all related system components, may not be exceeded due to constraints of a bus coupler (query 5, Fig. 1). For the evaluation of scalability of such structural inconsistencies we used 800 model instances of a lab size SysML based model of an aPS with 100 input/output values each (we assume real-world plants having 800 digital and analogue input, output). The calculation time lasts from 2 to 16 sec[1] (Fig. 1).

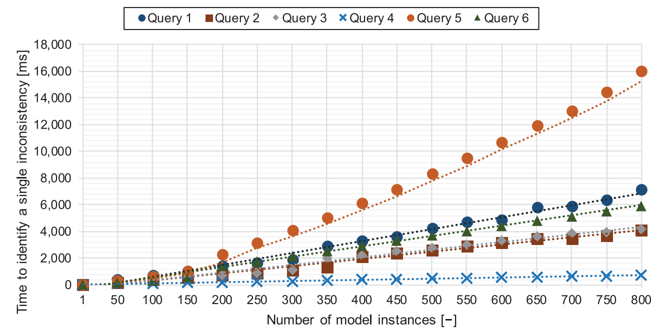

Fig. 1. Time to identify a single structural inconsistency (represented as a SPARQL query) depending on the number of model instances. Each model instance represents a single model of an aPS.

Additionally, the detection of behavioral inconsistencies in the components' interface require, for example, analyzing the specified moment over time, e.g., regarding the acceleration or deceleration of a transportation unit, compared to the physical laws of the unit and the specified throughput of the transported workpieces. These questions are up to now solved by physical simulation and/or by model checking approaches in case of discrete-event systems. I4.0 compliant CPPS shall support adaption during design as well as during operation phase. Therefore, structural and behavioral inconsistencies need to be checked in case of change.

Since plant data from CPPS, in general, are technically accessible due to increased connectivity of automation systems (cp. real-time capability) and Big Data algorithm and technologies are available, too, the basis for the I4.0 design principle real-time capability of data is given theoretically and will be discussed as a third challenge for automation. As application example in the following alarm pattern recognition in aPS is discussed giving an estimation of required calculation effort. Alarm pattern recognition supports human operators to identify the most hazardous alarm at once instead of browsing through alarm floods. To identify alarm patterns a huge amount of industrial alarm data logs have to be analyzed by a frequent pattern recognition algorithm (cp. [7]). Assuming alarm data logs consist of n messages, an adjacent matrix (size n $\times$ n) can be determined representing the state space of possible connected messages. Each message in the alarm log is represented by one state. For pairwise states it has to be calculated if this pair is included in the alarm log and additionally it is proved if this pair is holding a recognition criteria, e.g., the alarm is related to the same device in the piping and instrumentation diagram. If this pair holds a recognition criteria a third alarm is added to the sequence and the new sequence is evaluated by the recognition criteria in the same way as the pairs before. For each recognition criteria, there may be alarm pattern with length 1 to nxn. Assuming there are k recognition criteria the computational effort can be estimated as $Q(k \cdot n^2)$. Using the sizes of alarm log sizes presented in [7], a minimal computation is 4*12.707.161

[1]For the implementation, Fuseki was used, which is part of the Apache Jena Framework. Evaluations were run on a standard office PC (Windows 7 $\times$ 64 platform, 16 GB RAM, 4 cores, 3.6 GHz).

resulting in 94.828.644 and the maximal computational effort is 4*141.61 M equals to 566.44 M, which may lead in the worst case to 2–3 days of calculation time.

Besides technological issues, both industry and academia are well aware that best practices are necessary to establish I4.0 capable systems broadly. Especially small and medium size companies require to identify a concrete financial benefit, e.g., Return of Investments, before they decide to buy an I4.0 compliant aPS. Classification types for adaptivity are introduced, e.g., adaptable via add-ons, adaptable via modification, adaptable via parameterization, and adaptability via self-configuration as well as selected metrics, for e.g., real-time capabilities of self-configuration, restartability,[2] programming effort, and operator interaction [8]. However, widely accepted definitions of adaptivity, flexibility, and according metrics are missing.

As described, enabling technologies and design principles of I4.0 are already well known: Among others these are: real-time capabilities applying Big Data analytics and technologies, and as a basis data exchange formats, e.g., AutomationML, machine learning algorithm, and Semantic Web technologies. The semantics, i.e., a joint minimum understanding of these different disciplines, is required—described as coupled local discipline specific vocabularies as well as a global vocabulary—is still a challenge. Therefore, more powerful classified attributes to describe complex and customer-specific systems like aPS are required. Three challenges for automation were discussed, i.e., modularity of control software and mechatronic systems avoiding inconsistencies as well as data analytics and its real-time capabilities, in more detail providing future areas of research.

The expected benefit of I4.0 will be fully gained only combining most of the above mentioned design principles. Cooperation on an equal footing between involved scientific communities as well as companies working on I4.0, IoT, and CPS is required. Scientist and companies from computer science, product development, production systems, automation and control as well as ergonomics, human machine, psychology need to work cooperatively on these approaches to include all views and competencies. Multidisciplinarity is a real challenge when it comes to publishing joined results. The TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING (TASE) could provide the platform for such interdisciplinary research results, because automation and control overlaps with most aspects already.

Most important new technologies should not only be applicable in academia but also in industrial companies and supported by or connected to best-of-breed tools already used in industry. Finally, metrics to evaluate I4.0 compliance and benefit of adpativity need to be developed allowing to benchmark different solutions.

BIRGIT VOGEL-HEUSER, *Guest Editor*
Technical University of Munich
Munich 80333, Germany
vogel-heuser@tum.de

DIETER HESS, *Guest Editor*
3S-Smart Software Solution GmbH
Kempten 87439, Germany
d.hess@codesys.com

[2]Restarting is the procedure to resynchronize the control system and the physical system, such that the production can be restarted and eventually be completed.

REFERENCES

[1] [Online]. Available: https://en.m.wikipedia.org/wiki/Industry_4.0 (accessed Jan. 27, 2016)
[2] [Online]. Available: http://www.weforum.org/events/world-economic-forum-annual-meeting-2016 (accessed Jan. 27, 2016)
[3] B. Werner, "Object-oriented extensions for iec 61131–1," *IEEE Electron. Mag.*, vol. 3, no. 4, pp. 36–39, Dec. 2009.
[4] S. Feldmann, S. J. I. Herzig, K. Kernschmidt, T. Wolfenstetter, D. Kammerl, A. Qamar, U. Lindemann, H. Krcmar, C. J. J. Paredis, and B. Vogel-Heuser, "Towards effective management of inconsistencies in model-based engineering of automated production systems," in *Proc. 15th IFAC Symp. Inform. Control Manuf.*, Ottawa, ON, Canada, 2015, pp. 916–923.
[5] S. Feldmann, S. Rösch, C. Legat, and B. Vogel-Heuser, "Keeping requirements and test cases consistent: Towards an ontology-based approach," in *Proc. 12th IEEE Int. Conf. Ind. Informat.*, Porto Alegre, Brazil, 2014, pp. 726–732.
[6] J. Hirtz, R. B. Stone, D. A. McAdams, S. Szykman, and K. L. Wood, "A functional basis for engineering design: Reconciling and evolving previous efforts," National Inst. Standards Technol., Gaithersburg, MD, USA, NIST Tech. Note 1447, 2002.
[7] B. Vogel-Heuser, D. Schütz, and J. Folmer, "Criteria-based alarm flood pattern recognition using historical data from automated production systems (aPS)," *Mechatronics*, vol. 31, pp. 89–100, Oct. 2015.
[8] B. Vogel-Heuser, S. Rösch, J. Fischer, T. Simon, S. Ulewicz, and J. Folmer, "Fault handling in PLC-based industry 4.0 automated production systems as a basis for restart and self-configuration and its evaluation," *J. Softw. Eng. Appl.*, 2016.