Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 180 (2021) 50–59

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Industry 4.0 and Smart Manufacturing

Towards Mastering Variability in Software-Intensive Cyber-Physical Production Systems

Rick Rabiser, Alois Zoitl*

CDL VaSiCS, LIT CPS Lab, Johannes Kepler University Linz, Altenberger Str. 69, 4040 Linz, Austria

Abstract

Software-intensive Cyber-Physical Production Systems (SiCPPS), like metallurgical plants or manufacturing plants, are highly variable systems of systems that frequently evolve. They typically involve a large number of heterogeneous components (mechanical, electrical, mechatronic, software) that can be configured and combined in different ways. Variability results not only from hardware and software components but also development processes, disciplines (mechanical, electrical, software engineering), methods, and tools. Dealing with variability in industry currently depends too much on mostly tacit domain expert knowledge and custom-built tools focusing on very specific artifacts and software and hardware platforms. Existing research in the area of SiCPPS does not explicitly and systematically deal with variability. Promising software engineering methods and tools, e.g., from the area of Software Product Lines, need to be adapted for the particular challenges in SiCPPS. In this research preview paper, we discuss open research issues, research goals, and propose a research agenda towards mastering variability in SiCPPS.

Peer-review under responsibility of the scientific committee of the International Conference on Industry 4.0 and Smart Manufacturing

Keywords: variability; cyber-physical production systems; variability modeling; variability mining; evolution.

1. Introduction and Motivation

A Cyber-Physical System (CPS) is a system in which physical and software components are deeply intertwined, each operating on different spatial and temporal scales, exhibiting multiple and distinct behavioral modalities, and interacting with each other in a myriad of ways that change with context[1]. Such systems are software-intensive

* Corresponding author. Tel.: +4373224689480
E-mail address: alois.zoitl@jku.at

[1] US National Science Foundation Cyber-Physical Systems (CPS) Program: https://www.nsf.gov/pubs/2019/nsf19553/nsf19553.htm

Peer-review under responsibility of the scientific committee of the International Conference on Industry 4.0 and Smart Manufacturing
10.1016/j.procs.2021.01.128

systems [16] because software contributes essential influences to the design, construction, deployment, and evolution of the system as a whole. Software-intensive Cyber-Physical Production Systems (SiCPPS), like metallurgical plants or manufacturing plants, reflect the characteristics of CPS to the production domain, building ubiquitous systems that autonomously interact with their environments and are capable of flexibly manufacturing a variety of products based on customer demands.

SiCPPS are highly variable systems of systems [27] that frequently evolve. They are characterized by large numbers of tightly integrated, heterogeneous components in a network, which may expand and contract dynamically [1]. A further characteristic of these components is that they can be configured and combined in different ways [9]. The engineering process of SiCPPS involves multiple engineering disciplines (e.g., mechanical, electrical, mechatronic, software, business), which need to collaborate during system development.

As visualized in Figure 1 (left side), heterogeneous artifacts from these disciplines with different dependencies are currently mostly manually maintained by diverse stakeholders from different fields. Some dependencies between the artifacts result from physical requirements while others result from business decisions. Dependencies are often not documented explicitly, especially between artifacts of different disciplines. Furthermore, different artifacts are maintained using diverse tools with varying underlying concepts and semantics.

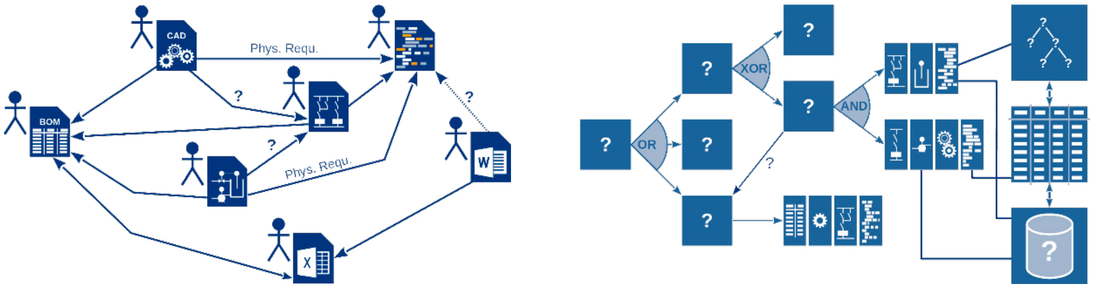

Fig. 1. In SiCPPS, diverse stakeholders mostly manually maintain heterogeneous artifacts from multiple disciplines with diverse dependencies; Variability is typically managed in SiCPPS using diverse mechanisms and tools.

In a recent article, Berger et al. [3] present the results of a study of twelve medium- to large-scale industrial cases regarding the current adoption of variability management techniques. They conclude that hardware is still one of the most significant drivers for variability in industry, directly followed by market pressure for customization. Also, they argue that Industry 4.0 and digitalization challenge variability management even more. Industry is very interested in mastering variability because it is a prerequisite for successful reuse, which helps to reduce the costs of development, certification, and maintenance as well as shorten time to market.

Existing research deals with engineering cyber-physical systems, including versioning and linking of models and artifacts [4]. However, systematic variability management has so far mostly been neglected in existing CPS research [41]. A lot of research on variability has been conducted in software engineering, particularly in the field of software product lines since the 1990ies [29][32]. What can be observed, however, is that only in early years (1990-2000), product line research focused on (and was motivated) mainly by software-intensive systems, i.e., large real-world systems with complex hardware controlled by complex software. Later on (2000-2015) research more and more focused on software, services, and operating systems' variability. Recent work has also focused too much on toy examples and evaluations within the lab instead of real-world systems [32].

It remains unclear what approaches, methods, and tools could be used for practical (SiCPPS-related) variability management problems. Specifically, as indicated in Figure 1 (right side), in SiCPPS the variability of diverse heterogeneous artifacts needs to be managed and a "one-size-fits-all" variability modeling approach is thus deemed to fail. Instead, different disciplines require different approaches to manage variability, which need to be integrated [15]. A key challenge will be to integrate higher-level variability (regarding the overall SiCPPS) and lower-level variability (regarding the detailed configuration of particular artifacts and components) while dealing with dependencies across variation points and ensuring consistency.

In our research, we focus on mastering variability in SiCPPS. Specifically, in cooperation with an industry partner from the metallurgical plant-building domain, we aim to investigate and integrate existing research on variability from the area of software engineering, extend it to address the particular challenges of SiCPPS, and apply it in the field of SiCPPS and systems engineering.

In the remainder of this paper, we first provide a brief overview of the state of the art regarding variability and SiCPPS and also discuss variability challenges in practice. Based on this discussion of the state of the art and practice, we discuss open research issues and our research goals. We conclude with a research agenda outlining our planned research.

2. State of the Art

Our work relies on work from the areas of cyber-physical (production) systems, variability and software product lines, and particularly preliminary work on variability and cyber-physical (production) systems.

Harrison et al. [17] provide a detailed summary of existing engineering standards, methods, and tools for *cyber-physical automation systems*, e.g., existing tools and standards such as AutomationML, ProSTEP iViP, and eCl@ss for end-to-end engineering; OPC-UA/IEC 62541 for communication; electronic device description (EDD) and field device tool (FDT) on the information layer; field device integration (FDI) as integration technology; component-based and model-based systems for developing CPS (see, e.g., Vogel-Heuser et al. [37] and the VDI/VDE RAMI reference model of Industry 4.0); service-oriented engineering [45]; interoperability and information models [5]; and simulation-based approaches (such as virtual factories [40]), virtual commissioning [18], and co-simulation [28]).

As Harrison et al. [17] point out: "much has been published about potential benefits of the adoption of cyber-physical systems in the manufacturing industry. However, less has been said about how such automation systems might be effectively configured and supported through their lifecycles and how application modeling, visualization, and reuse of such systems might be best achieved." They present their engineering environment vueOne to address this gap. In their 2017 book, Biffl et al. [5] summarize existing work on multi-disciplinary engineering for SiCPPS. Biffl et al. [4] also presented a model-driven engineering approach for developing and providing versioning and linking support for SiCPPS modeled using AutomationML.

SiCPPS domain experts in practice, however, still use diverse custom-developed and third-party engineering tools. This results in a heterogeneous tool landscape with a large number of engineering tools operating on different data structures [12]. Model development, maintenance, and integration, as well as data exchange and sharing all still are major challenges in practice [19]. Existing work deals with engineering SiCPPS, including versioning and linking of models and artifacts. However, systematic variability management has so far mostly been neglected [41].

A lot of research on *variability* has been conducted in the area of *software product lines* since the early 1990ies [29][32]. A Software Product Line (SPL) has been defined as “a set of software-intensive systems that share a common, managed set of features satisfying the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way” [7]. Some initial research has also been conducted on variability and CPS. For instance, ter Beek et al. [35] present a feature model of a family of train control systems. Brings et al. [6] report on an exploratory case study on variability modeling in the industrial automation domain. In their position paper, Krüger et al. [20] propose a classification of variability aspects relevant for CPS, point out current challenges in variability modeling, and sketch open research questions (cf. Section 4). Most importantly, they point out that a CPS’ variability typically includes heterogeneous aspects, which have to be integrated into a common model (cf. Section 5). Safdar et al. [33] studied the use of existing variability modeling techniques for CPS. They argue that there is a need to extend existing or to propose new approaches. This is what we aim to do by bringing together our own earlier work on software product lines and variability [10][30][31][36] with our work on industrial control and production (automation) systems [38][42][43][44]. Recently, one of the authors has also started working with colleagues at TU Vienna towards modeling the variability of products, processes and resources in CPPS engineering [13][23][24], which is an important additional input to the work proposed in this paper.

3. State of Practice

In industry the typically applied development process is often based on a so-called "Base Line". The Base Line contains a super-set of all functionalities offered per plant/machine type. For each specific project, the Base Line is taken (following a clone-and-own reuse approach) and converted (extended). In a project execution phase, the generated control software is manually adjusted to the specific plant configuration, tested, and eventually commissioned.

This process already greatly reduces the engineering and development effort. However, there are still critical issues to be addressed. The high variability in plant configurations is still mainly handled manually. Changes in plant configurations during project execution currently also have to be tracked and processed manually. Because of this, the project execution effort is increasing. There currently is no link between Base Line and project-specific code. Therefore, it is very hard to bring fixes and improvements identified during project execution back to the Base Line, i.e., there is no support for round-trip engineering.

The goal for our research is thus to better handle plant variability for the Base Line, to improve the project generation process for reducing the project-specific engineering effort, and to support round-trip engineering. Motivated by this, we plan to conduct research focusing on methodological support for dealing with variability and semantic integration in SiCPPS and apply approaches and tools on use cases from the industry (cf. Section 6).

4. Research Issues

Based on the review of the state of the art [20][41] and practice, our own earlier work, and diverse preparatory workshops we already conducted with our industry partner, we identified the following open research issues:

(RI1) Precarious dependence on tacit expert knowledge and missing requirements. The requirements of SiCPPS are typically not fully documented. Many only exist in the heads of engineers and are thus tacit knowledge. Also, existing information on requirements is distributed among diverse artifacts, e.g., detailed technical specifications, design models, (construction, electrical, …) plans, implementation artifacts, test plans, and even (user/technical) manuals. Due to the heterogeneity of these artifacts – and the fact that different roles from different disciplines, often with quite diverse backgrounds and using different techniques and technologies, create them – it is almost impossible to get a complete overview of the requirements of SiCPPS, especially also regarding the variability of the system. A team of very experienced personnel can still successfully engineer, i.e., plan, design, develop, deploy/commission, and maintain SiCPPS. However, the missing overview of the requirements makes the learning curve for new employees very steep and even slows down experienced personnel. This issue motivates (variability) models explicitly representing tacit expert requirements knowledge.

(RI2) Insufficient automation support. Any modeling approach requiring to manually create, populate, and, particularly, to maintain and evolve models will fail in practice. Even if the additional re-engineering and modeling effort would be accepted by industry, people would still continue to work with their artifacts and not manually check back models anytime they change something in their artifacts (documents, plans, source code, etc.). Existing work, however, typically assumes somebody manually creates and maintains models. Also, as Krüger and Berger [21] point out "empirical data to assess the costs and benefits of such a re-engineering […], despite decades of research on product lines and platforms, […] is scarce."

(RI3) Difficulties in assessing the impact of (changed) requirements and artifacts. Even with complete (variability) models, modeling the problem space alone does not allow assessing the impact of selecting, configuring, and adapting system features. This hinders automation (cf. RI2), e.g., when deploying and commissioning systems, and also affects any maintenance activities. Specifically, engineers responsible for maintenance have a hard time understanding the possible implications of any change they make. At the same time, they have to react fast to keep production systems up and running or to make them running again after failures. Each minute of a machine downtime costs a lot of money. Existing research focuses on modeling the problem space and on modeling the solution space. How to map these two spaces, particularly, how to map to the diversity of artifacts relevant in a SiCPPS context across problem and solution space, considering their semantics, is mostly neglected.

(RI4) Ambiguous and imprecise requirements and ambiguous and conflicting data, standards and regulations. Engineering of SiCPPS is characterized by an environment of highly heterogeneous disciplines, each with their own kind of engineering data [5]. Depending on the discipline, the way this information is represented and maintained

historically grew for many years. Representations are often imprecise and require implicit discipline knowledge for handling and understanding this information. Even further, in different application areas, different variants for different standards or regulations are applied to these models [37]. Discipline-specific engineering models with rich semantics and clear definitions are missing. Still, the engineering data from the different disciplines is highly interdependent [25]. However, data is mostly maintained only within each discipline. A clear formal definition of these interdependencies is lacking and engineers have no guidelines or rules available that aid in ensuring consistency. As a result, currently either a high-effort manual synchronization is applied or workflows are organized sequentially. The latter significantly slows the progress of a project [26]. Approaches from the software engineering domain tried to introduce dedicated languages for modeling this information (e.g., SysML and AADL [14]). However, their acceptance is low, as additional models have to be developed and maintained (cf. RI2). Previous experience in model-based systems engineering of SiCPPS [39] as well as regarding the open tool integration of SiCPPS engineering tools [38] has shown that a key requirement for efficient and effective engineering is to (re-)use the existing engineering data from the different engineering disciplines.

(RI5) Conflicting or contradictory requirements and artifacts. As SiCPPS act within an environment, with humans in the loop as well as with other systems, they must ensure safety and correctness at all times. This also holds true for variability models [8]. Inconsistencies can easily be introduced in variability models, even with a high degree of automation (cf. RI2). There could be mistakes in the heuristics and rules automatically updating the models and users might have manually changed something without the automated support being aware of these changes and their impact (cf. RI3). Existing work on checking the consistency of (design) models and software engineering artifacts [2][11][36] might be a promising starting point. However, it remains unclear how to deal with (non-software) artifacts in a SiCPPS context.

5. Research Goals

Figure 2 provides an overview of our research goals and their interrelations in the context of a SiCPPS development process assuming the availability of a common platform (or Base Line, cf. Section 3) from which products are derived.

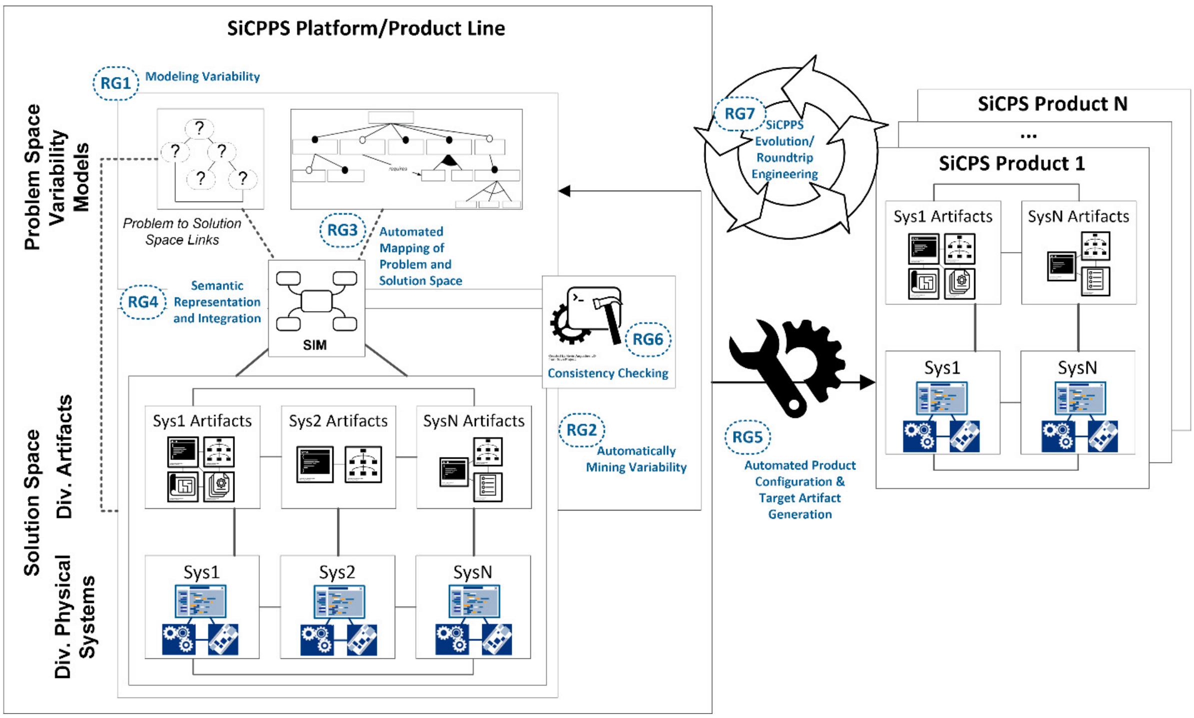

Fig. 2. Research goals towards mastering variability in SiCPPS.

(RG1) Research Goal 1: Modeling Variability in SiCPPS. Along the lines of Lee et al.'s review [22], we aim to support "better engineering of cyber-physical systems through better models". First, an explicit representation of the problem space of SiCPPS is needed. Variability models could be used for this purpose, e.g., feature models or decision models [8]. Feature models capture features – the end users' understanding of the general capabilities of systems in the domain – and the relationships among them. Decision models define the decisions that must be made to specify a member of a domain. While several variability modeling approaches have been developed with software-intensive systems in mind, the specifics of SiCPPS have so far not been explicitly addressed. Specifically, the need to deal with different disciplines in the SiCPPS context and process as well as artifact variability requires further research. Multi-view modeling approaches and orthogonal variability modeling approaches seem to be a promising starting point.

(RG2) Research Goal 2: Automatically Mining Variability in SiCPPS. An approach is needed that automatically populates and updates problem space models by analyzing the artifacts people are used to work with and identify problem-space relevant information. Approaches such as feature facets and, more generally, feature identification and extraction approaches, constraint extraction approaches, and feature model synthesis approaches work in this direction (see related work in Meixner et al. [24]). An approach to automatically mine variability in SiCPPS must be able to parse the heterogeneous artifacts and properly interpret the information based on predefined rules or heuristics. Only in case something cannot be fully automated, domain experts should be involved. Solutions offered by experts must later become programmed rules that automate the task in the future. Further research is required on how to interpret certain types of artifacts from a problem space point of view and on how to define the rules and heuristics to populate and update problem space models.

(RG3) Research Goal 3: Automated Mapping of Problem and Solution Space in SiCPPS. Due to the diversity of solution space artifacts that exist in a SiCPPS context, any "one-size-fits-all approach" is destined to fail. Existing approaches to map problem and solution space need to be adapted to the SiCPPS context and particularly they need to be flexible enough to easily allow integrating new artifacts and models. Existing links between solution space artifacts, e.g., as provided by existing standards such as AutomationML, need to be considered. If an explicit model of engineering models and their relations is available (cf. RG4), links from problem to solution space can be created based on this model in a partly automated manner. When automatically creating problem space models by analyzing solution space artifacts (see RG2 above), links to the originating artifacts can also automatically be created or updated. However, not all possible relations can be derived in this way, as any feature might have different impacts on many other artifacts. An additional mining approach, again with different rules and heuristics, will thus be required to go the "other direction" and analyze for each problem space element the possible impact on different artifacts and then create or at least propose creating problem-to-solution-space links. While it will be hard to fully automate this, proposing links to domain experts to select from will still be a large improvement to manually creating and maintaining such links. Feature location techniques based on static analysis, dynamic analysis, information retrieval or hybrid techniques could be used as a starting point.

(RG4) Research Goal 4: Semantic Representation and Integration. Knowledge plays an essential role in automation systems engineering and especially in the interaction of the different disciplines evolved. It is a vital requirement to extract the knowledge from the existing engineering models of the different disciplines. To make the information from different engineering models and data understandable, we need to model the necessary information about these models and clearly define the domain knowledge and relationships. We plan to do this with a Semantic Integration Model (SIM). Ontologies allow the explicit specification of a domain of discourse, increasing the level of specification of knowledge by incorporating semantics into the data, and promote its exchange in an explicitly understandable form. The starting point for the SIM can be on the one hand the engineering models of the different disciplines and on the other hand existing modeling techniques for systems engineering (e.g., SysML, AADL) and according data integration standards (e.g., AutomationML). As each engineering model only targets a sub-aspect of the overall system, a model alone or the pure model integration is often not sufficient. A lot of implicit knowledge is hidden in the cross-discipline integration of the models. This knowledge is currently tacit and mainly only accessible by human interpretation. We will investigate how implicit knowledge can be derived and utilized for consistency checks and target artifact generation (cf. RG5). This encompasses the development of means and methods for capturing domain knowledge from experts typically performing these tasks in the SIM.

(RG5) Research Goal 5: Generating and Configuring SiCPPS Target Artifacts. A particular importance lies on automating product configuration and artifact generation. Generating target artifacts based on engineering models is

typically done using single input models and developed on a case-by-case basis. This is error prone, hard to maintain, and has to be redone for every new target artifact [34]. The semantic integration of engineering data in the SIM (cf. RG4) will simplify the target artifact generation process. As target artifacts, we envisage, for example, controller configurations, control program fragments, automated testing scenarios, or system-specific multidisciplinary simulation setups. Additionally, utilizing the information contained in variability models will allow quickly deriving multiple variants. In the SiCPPS context a challenge is that the knowledge required for product derivation is often distributed in the heads of different people who might be unavailable. Furthermore, variability models can become rather complex models defining diverse interdependencies among artifacts that no single person can fully understand. In addition, it is hardly possible to satisfy customers' requirements solely by reusing existing assets from the product line. Instead, customers typically articulate new requirements that require additional engineering effort. While software can easily be reused and copied, physical components need to be actually built and commissioned.

(RG6) Research Goal 6: Consistency Checking for SiCPPS. We want to enable reasoning and querying across engineering disciplines so that models can be checked and validated for integrity and consistency and possible problems found can be corrected already in early phases of the SiCPPS engineering process. Specifically, we plan to develop an approach that can check the consistency within models as well as across models. This includes evaluating existing approaches to define and check consistency constraints and adapting them to the SiCPPS context. Defining consistency constraints can be particularly challenging in the context of SiCPPS, especially, if non-software-engineering stakeholders need to define constraints. This motivates a special domain-specific language, maybe even one that is adaptable to different (groups of) users, and to put a special emphasis on its usability.

(RG7) Research Goal 7: SiCPPS Evolution/Round-trip Engineering. As described in RG5, it is hardly possible to satisfy customers' requirements solely by reusing existing artifacts, i.e., the "blue-sky derivation scenario" does not hold in practice. Many newly developed artifacts, however, have the potential to also become useful in future projects. It is thus essential to support the round-trip, i.e., decide which newly developed artifacts have the potential for reuse and then integrate them properly with the product line. This leads to the evolution of the product line. As typically one will not build a product line from scratch, similar evolution support also makes sense for the initial adoption of a product line approach based on already existing artifacts (cf. RG2 and RG4). In a SiCPPS context, support for evolution must further consider the heterogeneity of (newly developed) artifacts and their dependencies. The proposed SIM will serve as base infrastructure, as it allows to compare the diverse solution space models on a semantic level by utilizing the discipline knowledge (cf. RG4 and RG5). A key focus of our research will be on how to identify changes across the engineering models and how to derive variants of existing modules or new modules to support evolution.

6. Research Agenda

We plan to address the discussed research issues and reach the described research goals in a long-term (7 years) industry-academia research collaboration project with a team of 5-10 researchers and at least one industry partner. Based on further, more detailed and systematic analyses of the state of the art and practice and further workshops with our industry partner as well as potentially other companies, mainly also to better understand their SiCPPS' architecture and variability, we will iteratively and incrementally work towards our research goals. We will start with a concrete case study provided by our industry partner, i.e., a plant automation CPPS in the area of continuous casting and hot and cold rolling in the metallurgical industries. It involves stakeholders from multiple disciplines and is based on decades of development using diverse methods, languages and tools. The system is highly variable and typically each deployed plant is individual and customized to specific needs of a heterogeneous customer base. We will base our research on the existing engineering models from this system, individually developed by members of diverse involved CPPS disciplines. Explicit variability models are currently not used in the system. Variability information is available in the heads of the involved engineers, but also documented in different spreadsheets and configuration files available for us to analyze and use. This case study is thus a very good basis for evaluating our research and develop first prototypes.

Specifically, we will develop approaches to support refactoring and modeling existing SiCPPS to better cope with their variability and semantics (cf. RGs 1-4). We will develop approaches for generating software and diverse other SiCPPS artifacts based on the information captured in models (RG5). We will work on tool prototypes to support the modeling process as well as the model-based configuration of SiCPPS (RGs 1-7). We will work on approaches and tools to automatically populate our models (and map artifacts across models) by analyzing existing SiCPPS artifacts (RGs 2 and 3). We will put a particular emphasis on roundtrip engineering and the maintenance and evolution of SiCPPS (RG7) and will develop an approach supporting to check and validate our models for integrity and consistency including automated fixes/suggestions for fixes in case of detected issues (RG6).

We will follow an iterative and incremental approach and continuously refine our plans based on feedback from our industry partner, from scientific collaborations, and from the research community. The following three metrics (to be extended) will be essential to evaluate the success of our research: (1) *Variability modeling effort*: based on existing engineering data models and variability information available in spreadsheets and configuration files, manually creating variability models is not feasible. Thus, one main measure is how much of the variability models can be automatically derived and how much manual effort will remain (see RGs 2 and 3). This will also decide about the industrial applicability of the developed variability modeling approaches. (2) *Product configuration and development effort*: when generating target artifacts (see RG5) based on variability models, the essential metric will be the amount of necessary manual adaptations of generated artifacts and manual development of additional artifacts. We plan to focus, in a first phase of our research, mostly on the control software generation before approaching other artifacts, such as simulation models. (3) *Roundtrip effort*: for any changes introduced to products during commissioning or maintenance, that could be useful for future products too, the effort to eventually include them in the platform/product line (refactoring; cf. RG7) is highly relevant.

7. Conclusion

In this paper, we have described why dealing with variability in software-intensive cyber-physical production systems is challenging in practice and have clarified that there is a lack of systematic approaches, methodologies, and solutions. We discussed the challenges (research issues) involved, specifically, precarious dependence on tacit expert knowledge and missing requirements, insufficient automation support, difficulties in assessing the impact of (changed) requirements and artifacts, ambiguous and imprecise requirements and ambiguous and conflicting data, standards and regulations, and conflicting or contradictory requirements and artifacts. We then outlined research goals and a research agenda. Specifically, in our long-term research project, we plan to develop a tool-supported approach for mastering variability in SiCPPS including support for automated variability mining and mapping, consistency checking, and evolution/roundtrip engineering.

Acknowledgements

The financial support by the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development, and the Christian Doppler Research Association is gratefully acknowledged. We want to explicitly thank our industry partner for their continuous support.

Appendix A. Author biographies

A.1. Rick Rabiser

Rick Rabiser is full professor for Software Engineering in Cyber-Physical Systems at the Linz Institute of Technology CPS Lab at Johannes Kepler University Linz, Austria. He holds a Master's and a Ph.D. degree in Business Informatics as well as the venia docendi (Habilitation) in Practical Computer Science from Johannes Kepler University Linz. His research interests include but are not limited to variability management, systems and software product lines, software maintenance and evolution, automated software engineering, requirements monitoring, and usability and user interface design.

A.2. Alois Zoitl

Alois Zoitl is full professor for Cyber-Physical Systems for Engineering and Production at the Linz Institute of Technology CPS Lab at Johannes Kepler University Linz, Austria. He holds a Master's and a Ph.D. degree in

Electrical Engineering from Technical University of Vienna. His research interests include but are not limited to highly distributed control systems, engineering methodologies for industrial automation systems, component-based modeling of control applications, reconfigurable production systems as well as modeling of dynamic reconfiguration of real-time constrained control applications

References

[1] Antsaklis, P., 2014. Goals and challenges in cyber-physical systems research. Editorial of the editor in chief. IEEE Transactions on Automatic Control, 12(59):3117-3119, 2014.

[2] Blanc, X., Mounier, I., Mougenot, A., Mens, T., 2008. Detecting model inconsistency through operation-based model construction. In Proc. of the 30th International Conference on Software Engineering (ICSE), ACM, 511-520.

[3] Berger, T., Steghöfer, J. P., Ziadi, T., Robin, J., Martinez, J., 2020. The State of Adoption and the Challenges of Systematic Variability Management in Industry. Empirical Software Engineering, 2020 (to appear).

[4] Biffl, S., Maetzler, E., Wimmer, M., Lueder, A., Schmidt, N., 2015. Linking and versioning support for AutomationML: A model-driven engineering perspective. In Proc. of the 2015 IEEE 13th International Conference on Industrial Informatics (INDIN), IEEE, 499-506.

[5] Biffl, S., Lüder, A., Gerhard, D., 2017. Multi-Disciplinary Engineering for Cyber-Physical Production Systems: Data Models and Software Solutions for Handling Complex Engineering Projects. Springer.

[6] Brings, J., Daun, M., Bandyszak, T., Stricker, V., Weyer, T., Mirzaei, E., Neumann, M., Zernickel, J. S., 2019. Model-based documentation of dynamicity constraints for collaborative cyber-physical system architectures: Findings from an industrial case study. Journal of Systems Architecture, 97:153-167.

[7] Clements, P., Northrop, L., 2011. Software Product Lines: Practices and Patterns: SEI Series in Software Engineering. Addison-Wesley.

[8] Czarnecki, K., Grünbacher, P., Rabiser, R., Schmid, K., Wasowski, A. Cool Features and Tough Decisions: A Comparison of Variability Modeling Approaches. In Proc. of the 6th International Workshop on Variability Modeling of Software-Intensive Systems (VaMoS 2012), ACM, 2012, 173-182.

[9] Derler, P., Lee, E.A., Vincentelli, A.S., 2012. Modeling cyber-physical systems. Proceedings of the IEEE, 100(1):13-28.

[10] Dhungana, D., Grünbacher, P., Rabiser, R., 2011. The DOPLER Meta-Tool for Decision-Oriented Variability Modeling: A Multiple Case Study. Automated Software Engineering, 18(1):77-114.

[11] Egyed, A., 2006. Instant Consistency Checking for the UML. In Proc. of the 28th International Conference on Software Engineering (ICSE), IEEE, 381-390.

[12] Fay A., Biffl S., Winkler D., Drath R., Barth M., 2013. A method to evaluate the openness of automation tools for increased interoperability. In Proc. of the 39th Annual Conference of the IEEE Industrial Electronics Society (IECON), IEEE, 6842-6847.

[13] Feichtinger, K. Meixner, K., Rabiser, R., Biffl, S., 2020. Variability Transformation from Industrial Engineering Artifacts: An Example in the Cyber-Physical Production Systems Domain. In Proc. of the 3rd International Workshop on Variability and Evolution of Software-Intensive Systems (VariVolution), collocated with SPLC 2020, Montréal, Canada, ACM, 2020.

[14] Feiler, P. H., Gluch, D. P., 2012. Model-based engineering with AADL: an introduction to the SAE architecture analysis & design language. Addison-Wesley.

[15] Galindo, J. A., Dhungana, D., Rabiser, R., Benavides, D., Botterweck, G., Grünbacher, P., 2015. Supporting Distributed Product Configuration by Integrating Heterogeneous Variability Modeling Approaches. Information and Software Technology, 62(6):78-100.

[16] Gerostathopoulos, I., Bures, T., Hnetynka, P., Keznikl, J., Kit, M., Plasil, F., Plouzeau, N., 2016. Self-adaptation in software-intensive cyber–physical systems: From system goals to architecture configurations. Journal of Systems and Software, 122:378-397.

[17] Harrison, R., Vera, D., Ahmad, B., 2016. Engineering Methods and Tools for Cyber–Physical Automation Systems. Proceedings of the IEEE, 104(5):973-985.

[18] Hoffmann, P., Schumann, R., Maksoud, T.M., Premier, G.C., 2010. Virtual Commissioning of Manufacturing Systems. In Proc. of the 24th European Conference on Modelling and Simulation (ECMS), 175-181.

[19] Kharlamov, E., Grau, B. C., Jiménez-Ruiz, E., Lamparter, S., Mehdi, G., Ringsquandl, M., Nenov, Y., Grimm, S., Roshchin, M., Horrocks, I., 2016. Capturing industrial information models with ontologies and constraints. In Proc. of the International Semantic Web Conference, Springer, 325-343.

[20] Krüger, J., Nielebock, S., Krieter, S., Diedrich, C., Leich, T., Saake, G., Zug, S., Ortmeier, F., 2017. Beyond Software Product Lines: Variability Modeling in Cyber-Physical Systems. In Proc. of the 21st International Systems and Software Product Line Conference (SPLC), ACM, 237-241.

[21] Krüger, J., Berger, T., 2020: Activities and Costs of Re-Engineering Cloned Variants into an Integrated Platform. In Proc. of the 14th International Working Conference on Variability Model-ling of Software-Intensive Systems (VaMoS), ACM, 21:1-21:10.

[22] Lee, J., Bagheri, B., Kao, H.A.: A cyber-physical systems architecture for industry 4.0-based manufacturing systems. Manufacturing Letters, 3:18-23, 2015.

[23] Meixner, K., Rabiser, R., Biffl, S., 2019. Towards modeling variability of products, processes and resources in cyber-physical production systems engineering. In Proc. of the 23rd International Systems and Software Product Line Conference (SPLC), Volume B, ACM, 68:1-68:8.

[24] Meixner, K., Rabiser, R., Biffl, S., 2020. Feature Identification for Engineering Model Variants in Cyber-Physical Production Systems Engineering. In Proc. of the 14th International Working Conference on Variability Modelling of Software-Intensive Systems (VaMoS), ACM, 18:1-18:5.

[25] Mordinyi R., Winkler D., Ekaputra F.J., Wimmer M., Biffl S., 2016. Investigating model slicing capabilities on integrated plant models with AutomationML. In Proc. of the IEEE 21st International Conference on Emerging Technologies and Factory (ETFA), IEEE, pp. 1-8.

[26] Moser, T., Mordinyi, R., Winkler, D., Melik-Merkumians, M., Biffl, S., 2011. Efficient automation systems engineering process support based on semantic integration of engineering knowledge. In Proc. of the IEEE 16th International Conference on Emerging Technologies and Factory (ETFA), IEEE, 1-8.

[27] Nielsen, C. B., Larsen, P. G., Fitzgerald, J., Woodcock, J., Peleska, J., 2015. Systems of Systems Engineering: Basic Concepts, Model-Based Techniques, and Research Directions. ACM Computing Surveys, 48:18:1-18:41.

[28] Oppelt, M., Wolf, G., Drumm, O., Lutz, B., Urbas, L., 2014. Automatic model generation for virtual commissioning based on plant engineering data. In World Congress, 19(1):11635-11640.

[29] Raatikainen, M., Tiihonen, J., Männistö, T., 2019. Software product lines and variability modeling: A tertiary study, Journal of Systems and Software, 149:485-510.

[30] Rabiser, R., Grünbacher, P., Lehofer, M., 2012. A Qualitative Study on User Guidance Capabilities in Product Configuration Tools. In Proc. of the 27th IEEE/ACM International Conference Automated Software Engineering (ASE), ACM, 110-119.

[31] Rabiser, R., Vierhauser, M., Grünbacher, P., Dhungana, D., Schreiner, H., Lehofer, M., 2014. Sup-porting Multiplicity and Hierarchy in Model-based Configuration: Experiences and Lessons Learned. In Proc. of the 17th International Conference on Model Driven Engineering Languages and Systems (MoDELS), Springer, 320-336.

[32] Rabiser, R., Schmid, K., Becker, M., Botterweck, G., Galster, M., Groher, I., Weyns, D., 2018. A Study and Comparison of Industrial vs. Academic Software Product Line Research Published at SPLC. In Proc. of the 22nd International Systems and Software Product Line Conference (SPLC), ACM, 14-24.

[33] Safdar, S.A., Yue, T., Ali, S., Lu, H., 2016: Evaluating variability modeling techniques for supporting cyber-physical system product line engineering. In Proc. of the International Conference on System Analysis and Modeling, Springer, 2016, 1-19.

[34] Steinegger, M., Melik-Merkumians, M., Schitter, G.: Ontology-based framework for the generation of interlock code with redundancy elimination. In Proc. of the 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), IEEE, 2017, 1-5.

[35] ter Beek, M.H., Fantechi, A., Gnesi, S., 2018. Product line models of large cyber-physical systems: the case of ERTMS/ETCS. In Proc. of the 22nd International Systems and Software Product Line Conference (SPLC). ACM, 208-214.

[36] Vierhauser, M., Grünbacher, P., Egyed, A., Rabiser, R., Heider, W., 2010. Flexible and Scalable Consistency Checking on Product Line Variability Models. In Proc. of the 25th IEEE/ACM International Conference on Automated Software Engineering (ASE), ACM, 63-72.

[37] Vogel-Heuser, B., Diedrich, C., Fay, A., Jeschke, S., Kowalewski, S., Wollschlaeger, M., Göhner, P., 2014. Challenges for software engineering in automation. Journal of Software Engineering and Applications, 7(05):440.

[38] Waltersdorfer, F., Moser, T., Zoitl, A., Biffl, S., 2010. Version management and conflict detection across heterogeneous engineering data models. In Proc. of the 2010 8th IEEE International Conference on Industrial Informatics (INDIN), IEEE, 928-935.

[39] Wenger, M., Melik-Merkumians, M., Hegny, I., Hametner, R., Zoitl, A., 2011. Utilizing IEC 61499 in an MDA control application development approach. In Proc. of the IEEE International Conference on Automation Science and Engineering, IEEE, 495-500.

[40] Westkämper, E., Jendoubi, L., 2003. Smart factories-manufacturing environments and systems of the future. In Proc. of the 36th CIRP International Seminar on Manufacturing Systems, University of Stuttgart, 13-16.

[41] Yue, T., Ali, S., Selic, B. 2015. Cyber-physical system product line engineering: comprehensive domain analysis and experience report. In Proc. of the 19th International Systems and Soft-ware Product Line Conference (SPLC), ACM, 338-347.

[42] Zoitl, A., Prähofer, H., 2012. Guidelines and patterns for building hierarchical automation solutions in the IEC 61499 modeling language. IEEE Transactions on Industrial Informatics, 9(4):2387-2396.

[43] Zoitl, A., Vyatkin, V., 2009. IEC 61499 architecture for distributed automation: The 'glass half full' view. IEEE Industrial Electronics Magazine, 3(4):7-23.

[44] Zoitl, A., Lewis, R., 2014: Modeling control systems using IEC 61499. 2nd edition, The Institution of Engineering and Technology.

[45] Zühlke, D., Ollinger, L., 2011. Agile automation systems based on cyber-physical systems and service-oriented architectures. Advances in Automation and Robotics, 1:567-574.