Understand and Control Complexity in Cyber-Physical Systems by Analyzing Complexity Drivers

Michael Riesener
Chair of Production Engineering
Laboratory for Machine Tools and Production Engineering (WZL) at RWTH Aachen University
Aachen, Germany

Christian Dölle
Department Innovation Management
Laboratory for Machine Tools and Production Engineering (WZL) at RWTH Aachen University
Aachen, Germany

Alexander Keuper
Department Innovation Management
Laboratory for Machine Tools and Production Engineering (WZL) at RWTH Aachen University
Aachen, Germany
a.keuper@wzl.rwth-aachen.de

Günther Schuh
Chair of Production Engineering
Laboratory for Machine Tools and Production Engineering (WZL) at RWTH Aachen University
Aachen, Germany

2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS) | 978-1-7281-6207-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICPS49255.2021.9468255

***Abstract*— Manufacturing companies are confronted with an increased customer demand for innovation and individuality. Leading to new products, which are realized by the merging of physical mechatronic elements with virtual software elements. These systems are cyber-physical systems (CPS) and have the advantage to increase the functionality and enable additional services. The combination into one system offers more than the sum of its components. However, the merge of physical and virtual domains and their domain-specific development methods introduces new and unconsidered complexity into the product and its developing process. It is important to understand and control the additional complexity in order to improve the cost-benefit ratio of CPS. This paper is analyzing the system complexity of CPS by investigating its complexity drivers and their relations with the aim to identify the biggest levers to manage complexity in CPS and providing a basis for deriving complexity-oriented measures.**

Keywords*— *cyber-physical systems, complexity drivers, complexity management

I. Introduction

Global trends like digitization, service-orientation and individualization lead to higher customer expectations regarding the functionality as well as more heterogeneous and service-orientated customer requirements. The challenge for manufacturing companies is to have a higher product variety on the one side and on the other side to offer more functionality and services [1]. In order to fulfill these expectations, products started to cross the boundaries of purely mechanical domains, resulting in cyber-physical Systems (CPS) [2]. CPS are able to increase the functionality, customizability and enable additional services [3]. CPS are complex systems connecting the physical world of mechatronic parts with the virtual world of software applications. The connection between the physical and virtual domain is realized by connective elements, e.g. information networks. Additional components in the physical dimension, e.g. sensors, enable the collection of data, which is analyzed in the virtual dimension, e.g. on analytics platforms. New tools have to emerge in order to manage interdisciplinarity and new challenges in the development of CPS due to the combination of mechanic, electronic and software domains [4]. The linkage of these domains and their components lead to an increase of relations and interdependencies within the products, resulting in an increase of complexity, which is accompanied by higher development efforts and higher costs [5]. Since the importance of CPS will continue to increase, the control of complexity in CPS is a critical factor to lower costs and gain a competitive advantage [6]. In order to improve the cost-benefit ratio of CPS, this paper analyses the complexity in CPS by identifying their most relevant complexity drivers. Through the analysis of complexity drivers' interdependencies, it is possible to unveil complexity drivers that cause complexity propagation and therefore need to be controlled first. Knowledge about roots and symptoms of complexity enables a more efficient complexity-oriented development of CPS.

This paper is structured in five sections. After the introduction, the relevant terms regarding CPS and complexity are defined and explained in the second section. In the third section, related work is discussed to analyze existing solutions and define the research gap. Section four presents the most relevant complexity drivers, which are identified through literature analysis. An influence analysis will determine the complexity drivers with the strongest influence and the complexity drivers with the greatest influenceability. Finally, a conclusion is derived in section five.

II. Terminology

In this section, the relevant terms will be outlined in order to have a common understanding. In that matter, a description for CPS (A), system complexity (B) and complexity drivers (C) is given based on literature.

A. Cyber-physical system

CPS describe interdisciplinary systems combining the fields of mechanics, electronics and computer science [7]. CPS are comparable to mechatronic systems, with the difference that mechatronics are concentrating on the physical elements with additional aspects of electronics and software applications, while CPS emphasize that value is added through the network-like connection of the physical and virtual world [5, 8]. Berger et al. describe that CPS "consist of embedded systems, are based on powerful software systems, enable the integration in digital networks and create completely new system functionalities" [9]. Given their nature to connect individual products and enable the exchange of data

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE
978-1-7281-6207-2/21/$31.00 ©2021 IEEE

and information, CPS often appear in the context of "Industry 4.0" or the "Internet of Things" [10]. Based on an extensive literature review, LIU ET AL. propose a three-layer structure as a general structure of CPS [11]. The first layer is the physical layer, responsible for the generation of data through sensors and the realization of control responses through actuators. Software systems process data in the second layer, the information layer. The last layer is the system user, who is in direct exchange with the CPS and is consequently considered as a part of the CPS architecture [11].

The understanding in this paper is that CPS are network-like systems combining physical and virtual elements. They are distinguished by a high degree of interconnectedness within the systems themselves, across system boundaries and with the system users.

B. *System Complexity*

According to GRUSSENMEYER ET AL., complexity is defined as "characteristic of a system, which is determined by the number and diversity of elements, their relationships as well as the variance over time" [12]. In accordance to this understanding, WESTPHAL describes that a system is complex, if one "cannot fully grasp and control it mentally" [13]. In system theory, complexity is a property of a system describing the effect, which makes the system unpredictable, dynamic and "not easy to analyze" [13]. From an economical point of view, increased complexity can have different impacts on a system [14]. For example, a system with more variety is more likely to meet individual customer preferences resulting in additional value [14]. On the other hand, a more dynamic system with more variety, is less predictable or controllable, resulting in an increase in costs (resources) to manage it.

In the context of this paper, system complexity can be considered as constitutive characteristic of CPS. By enhancing this characteristic, additional functionality can be implemented but also additional cost arise. The active management of complexity ensures positive benefit-cost ratio.

C. *Complexity Drivers*

Complex systems do not allow typical cause-and-effect thinking, nor does a certain measure lead with certainty to a certain result [15]. This creates the need for a different way to characterize and understand complexity. Therefore, complexity drivers are used to investigate complexity. They can be understood as a "phenomenon that causes the (increasing) complexity of a system" [16].

In order to evaluate system complexity it is required to be able to measure complexity and therefore the complexity drivers quantitatively [17]. Enabling faster identification of possible causes of negative financial effects. By analyzing and controlling the complexity drivers of a specific system, the complexity of that system becomes controllable [16]. The step of identifying complexity drivers in order to control system complexity has already proven to be useful in classical product development (e.g. in [17] or [18]) and needs to be expanded to the specifications of CPS.

III. RELATED WORK

In this section, related work is presented and discussed with regard to the identification of complexity drivers in CPS. From this, a theory deficit is derived. An extract of the reviewed literature is presented in Fig. 1.

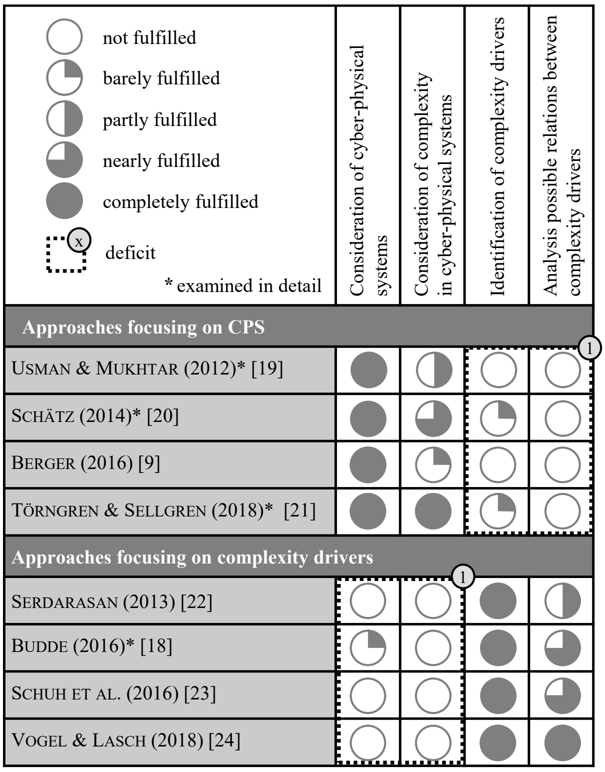


Fig. 1. Investigation of existing research

USMAN & MUKHTAR focus on the requirements of CPS during the design phase [19]. On the one hand, CPS need to be adaptable in many ways and be robust to errors, on the other hand, an operating CPS cannot be modified easily. The authors claim that due to this fact, additional thoughts on how to fulfil future requirements need to be considered while designing a CPS. The authors propose that four main factors need to be considered when designing a CPS: robust to change, responsiveness, capability as well as safety and security. Furthermore, each of these factors is also divided into a more defined structure. The authors are comprehensively looking at the design and structure of a CPS as well as stating them as complex. Although considering the complexity, the complexity itself is not further investigated, thus complexity drivers and their impact on the complexity are not shown.

The main objective of SCHÄTZ is to discuss how the mechanisms of modelling, developed for embedded systems, have to be changed or extended in order to be applicable for CPS [20]. He proposes three main domains as key principles of CPS, differentiating CPS from networked embedded systems. These domains are described as cross-, live-, and self-dimension. Each of these dimensions is also described as a complexity driver for CPS. The cross-dimension covers the interdisciplinary processes of CPS including different domains, disciplines and organizations. The live-dimension describes the requirements for updates and enhancement during the operating-time of a CPS, since CPS often cannot be turned off in whole. The self-control-functions like autonomous documenting and reporting, which are generally implemented in CPS, are part of the self-dimension. SCHÄTZ defines these dimensions as complexity drivers for CPS. A detailed analysis of these complexity drivers e.g. a root-cause analysis is not included.

TÖRNGREN & SELLGREN propose a detailed description of CPS, focusing on a better understanding of complexity in CPS

[21]. Their approach consists of a comparison of complexity aspects in CPS with the limitations in the current development process for CPS. Their goal is to understand what current methods are lacking. A detailed description of facets of complexity in CPS is given and four complexity dimensions are considered: variety, diversity, dynamics and uncertainty. Under these aspects, engineering methods are examined. However, there is no further evaluation of these dimensions, thus the scope is very generic and not detailed enough to explain the complexity emerging by CPS.

BUDDE approaches the complexity management of manufacturing companies [18]. The field of complexity within a company is separated into "positive" complexity, which helps satisfying customer requirements and "negative" complexity, which results in declining sales and profits per product. The goal of his work is to control the different aspects of complexity by understanding and evaluating the complexity. BUDDE presents an extensive method to make a benefit-cost quantification for complexity. In that regard, complexity drivers as well as different complexity dimensions within a company are identified. While complexity and complexity drivers are covered in depth, the author does not consider CPS, the combination of physical products with software and the complexity evolving from these integrated systems.

While there is a significant amount of literature that analyzes complexity drivers in manufacturing companies in general, there are only few approaches that focus on complexity in CPS. Therefore, the identified research deficit lies in the identification of complexity drivers in CPS.

IV. METHODOLOGY

In this section, the identified research deficit is addressed by deriving complexity drivers from literature and checking if they are applicable to CPS. In addition, from literature describing CPS, further CPS-specific complexity drivers are derived. In the next step, relations between the identified complexity drivers are investigated in order to determine those with a strong influence and those with a high influenceability. This analysis unveils the biggest levers to control complexity.

In the first step, a literature analysis is conducted to identify general complexity drivers. Numerous scientific publications have collected complexity drivers in different areas of application. For example, BUDDE identifies a total of 51 complexity drivers in the fields of products, technologies, customers, competitors and processes [18].

VOGEL & LASCH conduct an extensive literature review on complexity drivers in manufacturing companies [24]. A total of 486 complexity drivers are identified. The complexity drivers are categorized in three main groups (external complexity, internal complexity and general complexity) and four subcategories (society complexity, market complexity, internal correlated complexity and internal autonomous complexity).

ADAM presents 26 generic complexity drivers, which are divided into external and internal complexity drivers [25]. Further complexity drivers can be derived from SCHUH [26], GIEßMANN [27], GRÜBNER [28], KISSEL [29], SCHOENEBERG [30] and BLISS [31].

Since none of these approaches specifically investigates complexity drivers in CPS, it is possible that complexity drivers, which are exclusive to CPS, have not been considered yet. Therefore, an analysis of literature about CPS is conducted in order to derive additional complexity drivers and supplement the existing collection.

For example, BROY describes the networking of a system with the environment as a key characteristic of CPS as well as a factor bringing up new requirements for data security and reliability [32]. In this respect, for CPS, the *degree of interconnectedness with the environment* is relevant for the complexity and considered as a complexity driver. In addition, the *requirements for security* and *reliability* can be considered as complexity drivers.

NIGGEMANN ET AL. address the challenges for implementing a data driven controlling and monitoring system [33]. Challenges include the data-acquisition through different sources, certain human-machine-interfaces (HMI) and data storage, which can also be impacted by increasing complexity. Thus, the drivers *data volume* and *extend of HMI* are added to the collection.

Additional CPS-specific complexity drivers were identified by investigating the reviewed CPS literature in KEUPER ET AL. [34]. The combination of general and CPS-specific complexity drivers forms an extensive collection of 685 complexity drivers. Since many different sources are considered, a clean-up and consolidation is conducted in order to eliminate duplicates and summarize similar ones. The remaining complexity drivers are narrowed down to the ones which are relevant in the context of CPS. Finally, the set of complexity drivers is further reduced to the most important complexity drivers by aid of pairwise comparison. A summary of this procedure is shown in Fig. 2.

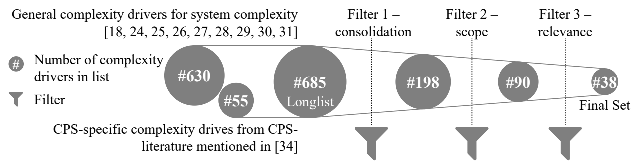


Fig. 2. Identification method for the final set of 38 complexity drivers

In order to structure the final set of 38 complexity drivers, CPS and the system complexity are divided into three dimensions and thus form a 3x3 matrix. CPS are divided into the physical, virtual and connectivity dimension in accordance with [11] and [35]. System complexity is divided into interdependence, variety and dynamic in accordance with [16] and [34]. The definition of every cell of the matrix is given in Fig. 3.

	interdependence	variety	dynamic
physical dimension	Describes the dependencies between the physical elements to each other and to the environment.	Describes the number and heterogeneity of physical elements.	Describes the change, development and different states of the physical elements over time.
virtual dimension	Describes the dependencies between the virtual elements to each other and to the environment.	Describes the number and heterogeneity of virtual elements.	Describes the change, development and different states of the virtual elements over time.
connectivity dimension	Describes the dependencies between the connectivity elements to each other and to the environment.	Describes the number and heterogeneity of the connectivity elements.	Describes the change, development and different states of the connectivity elements over time.

Fig. 3. Definition of the 3x3 matrix dimensions

In the next step, every complexity driver is allocated to one of the three dimensions of CPS and one of the three dimensions of system complexity. The final allocation of the 38 complexity drivers is shown in Fig. 4.

	interdependence	variety	dynamic
physical dimension	• customer requirements (hardware) • physical product interfaces • size and complexity of supply chain	• technology variety • no. of hardware functions • no. of modules with data interaction • no. of modules without data interaction • degree of functional integration (hardware)	• product life cycle • hardware technology and innovation cycles • hardware development effort
virtual dimension	• customer requirements (software) • software interfaces • size of the network of platform partner • compatibility with existing software	• no. of software functions • no. of software variants • degree of software integration • heterogeneity of data structure • data volume	• learning ability of the system • degree of autonomy • demand (-fluctuation) of processing capacity • software technology and innovation cycles • software development effort
connectivity dimension	• customer requirements (connectivity) • data flow beyond system boundary • degree of interconnectedness with the environment • extend of HMI • compatibility with existing platforms • data security requirements • reliability requirements	• geographical distribution of subsystems • no. of functions through connectivity • no. of services • data transfer & storage	• frequency of regulatory changes • degree of real-time capability

Fig. 4. Complexity drivers assigned to the 3x3 matrix

After the relevant complexity drivers have been identified and structured, the next step is to determine their relations and investigate the influence and influenceability of the complexity drivers. The used method has been applied before and is described in Fig. 5 [36].

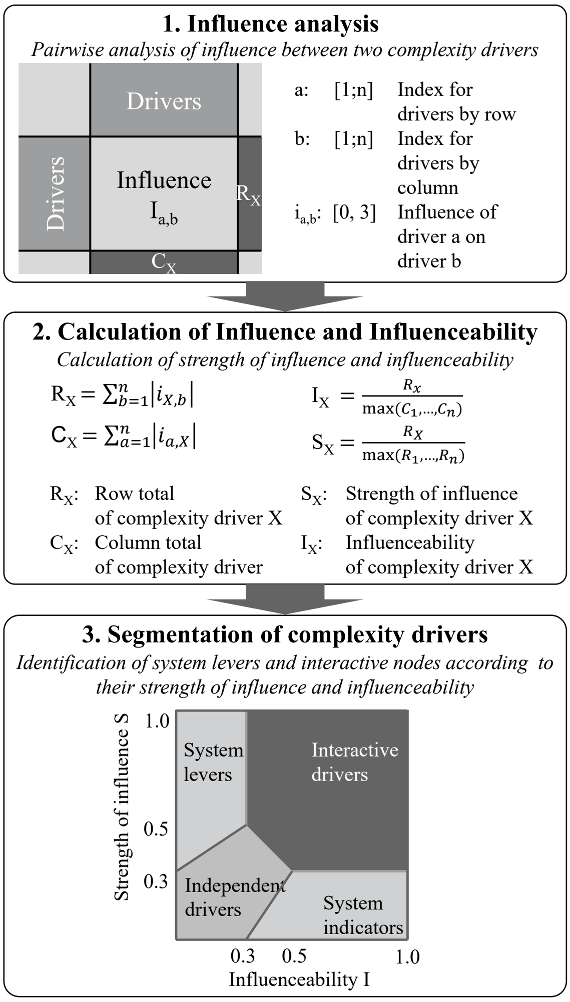


Fig. 5. Process for the relation analysis

The first step of the method analyzes the complexity drivers' influence on each other and documents it in the influence matrix. By calculating the row and column sums in this matrix, the strength of influence and influenceability can be determined. These values are then visualized and segmented in the influence portfolio.

The influence can be determined by answering the question: *"If the measure of complexity driver A increases (and therefore the system complexity caused by A increases), how strong will it influence the measure of complexity driver B (and therefore the system complexity caused by B)"*. The evaluation of the complexity drivers' influence on each other is conducted by analyzing CPS literature and identifying causal chains.

For example, GUNES ET AL. state that more homogeneity of components and systems lead to a reduction in necessary computing capacities [37]. This causal chain indicates that there is an influence of the *number of modules* and the *number of software variants* on the *demand of processing capacity*.

Another example can be derived from BROY and the fact that a higher degree of decision autonomy, e.g. an autonomous driving car, increase the requirements in security and reliability [32]. This causal chain indicates that the *degree of autonomy* influences the *security* and *reliability demands*.

Conclusive thinking based on practical experiences completes cells of the matrix, which cannot be filled by causal chains from the literature analysis. An extract of the final influence matrix is presented in Fig. 6.

Influence matrix: A cell answers the question: "How strong is the Impact of driver A(row) on driver B (column)?" e.g. an increase in the complexity of software interfaces increases the complexity of being compatible with other existing software resulting in a rating of 3. Scale: 0 = no impact 1 = weak impact 2 = medium impact 3 = strong impact	**software interfaces**	**size of the network of platform partner**	**compatibility with existing software**	**customer requirements (connectivity)**	**data flow beyond system boundary**	**degree of interconnectedness with the environment**	**…**	**active sum**
software interfaces		1	3	2	3	2	…	25
size of the network of platform partner	2		1	0	2	1	…	25
compatibility with existing software	3	0		0	1	1	…	14
customer requirements (connectivity)	1	1	1		2	3	…	26
data flow beyond system boundary	0	0	0	0		2	…	12
degree of interconnectedness with the environment	1	0	0	0	2		…	9
…	…	…	…	…	…	…	…	
passive sum	18	9	12	8	30	24		

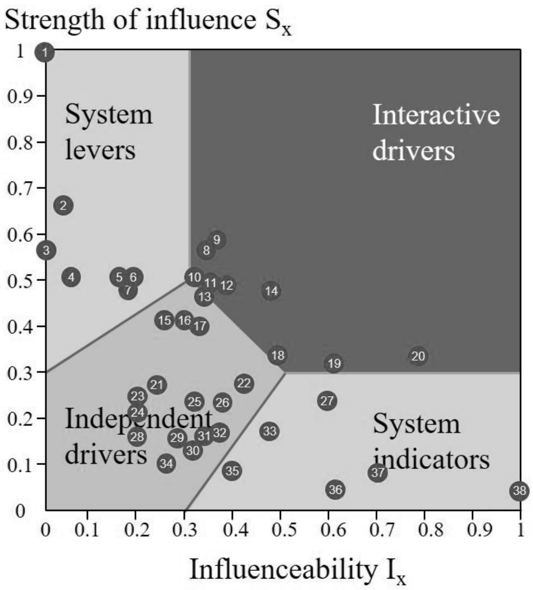
Fig. 6. Extract of the influence matrix

After the influence of the complexity drivers on each other is evaluated in the influence matrix, the column and row sums are used to identify system levers, system indicators, interactive and independent complexity drivers in the influence portfolio [38].

The influence of a complexity driver on other complexity drivers can be determined by the row sum R_x. A high row sum indicates that the complexity driver has a reinforcing effect on other complexity drivers and the system complexity will react sensitively on changes of the complexity driver. The column sum C_x indicates how much the complexity driver is influenced by others. A high column sum means that the complexity driver is influenced by many different other complexity drivers and is therefore highly influenceable.

System levers are complexity drivers that have a strong influence S_x on others while being mostly independent from changes in other complexity drivers. In contrast, systems indicators have a high influenceability I_x and are easily influenced by other complexity drivers but do not affect other complexity drivers. Interactive complexity drivers are both strong in influencing others but also being influenced easily. Independent complexity drives have barley any relations with other complexity drivers. There is no distinctive cut between these categories of complexity drivers and the boundaries are adjustable for clustering. A possible clustering is outlined in Fig. 7.

The first point of action should always focus on system levers and interactive complexity drivers. Independent complexity drivers and system indicators are less relevant due to their lower impact on the overall system complexity. The final allocation of the 38 complexity drivers in the influence portfolio is shown in Fig. 7.



1. customer requirements (software)
2. frequency of regulatory changes
3. customer requirements (hardware)
4. geographical distribution of subsystems
5. customer requirements (connectivity)
6. physical product interfaces
7. size of the network of platform partner
8. degree of real-time capability
9. degree of autonomy
10. technology variety
11. software interfaces
12. data volume
13. no. of functions through connectivity
14. no. of modules with data interaction
15. learning ability of the system
16. no. of software functions
17. no. of hardware functions
18. extend of HMI
19. reliability requirements
20. data security requirements
21. compatibility with existing software
22. demand (-fluctuation) of processing capacity
23. no. of services
24. degree of functional integration (hardware)
25. no. software variants
26. heterogeneity of data structure
27. data flow beyond system boundary
28. degree of software integration
29. no. of modules without data interaction
30. size and complexity of supply chain
31. software technology and innovation cycles
32. degree of interconnectedness with the environment
33. product life cycle
34. compatibility with existing platforms
35. software technology and innovation cycles
36. hardware development effort
37. data transfer & storage
38. software development effort

Fig. 7. Classification of the complexity drivers

V. Conclusion

Higher functionality and more variety is needed to fulfil increased customer expectations and more individual customer requirements. This leads to an increasing relevance of CPS, since CPS enable products with higher functionality through the connection of physical elements with virtual elements, creating a system capable of more than the sum of its individual elements. However, the interdisciplinary nature of CPS as well as the combination of many heterogeneous elements also lead to a higher system complexity. In order to be able to consider complexity during the development of a CPS, this paper identifies and analyzes complexity drivers in CPS. Taking both general complexity drivers and CPS-specific features into account, a set of 38 CPS-relevant complexity drives is identified. Furthermore, the influence of these complexity drivers on each other is analyzed, resulting in an overview of the system levers, system indicators, interactive and independent complexity drivers. In order to use the complexity drivers and their relations in the development of CPS, they need to be combined with a quantification method. Based on the quantification of complexity and the purpose of the CPS a benefit-cost ratio can be derived. Concrete measures to manage the system complexity and improve the benefit-cost ratio should address the system levers and interactive drivers first.

Future work should focus on breaking down the complexity drivers to indicators and connect these indicators with directly measureable key figures. Enabling the quantification of system complexity in CPS. In the next step, the purpose and value of the CPS should be evaluated and broken down into the 9 cells of the 3x3 matrix from Fig. 3, enabling a direct comparison of where the benefits and the complexity arise. Subsequently, concrete actions should be derived to adjust the relative complexity distribution to the relative benefit distribution in the 3x3 matrix. Furthermore, complexity management needs to be integrated in the development of CPS to maximize the added value by reducing the complexity to a necessary minimum.

Acknowledgment

The presented results have been developed within the research project "Complexity oriented design of cyber-physical systems" [GZ: SCHU 1495/146-1] funded by the Deutsche Forschungsgemeinschaft (DFG).

References

[1] G. Schuh, S. Rudolf, M. Riesener and J. Kantelberg, "Application of highly-iterative product development in automotive and manufacturing industry," 2016, pp.1.

[2] W.-G. Drossel, S. Ihlenfeldt, T. Langer and R. Dumitrescu, "Cyber-Physische Systeme," 2018, pp.197–222.

[3] M. Herterich, F. Uebernickel and W. Brenner, "Nutzenpotentiale cyber-physischer Systeme für industrielle Dienstleistungen 4.0," HMD Praxis der Wirtschaftsinformatik, vol. 5, 2015, pp.665–680.

[4] A. Hellinger and H. Seeger, "Cyber-Physical Systems. Driving force for innovation in mobility, health, energy and production," Acatech Position Paper, National Academy of Science and Engineering, vol. 2, 2011

[5] E. A. Lee, "Cyber Physical Systems: Design Challenges," IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC) vol.11, 2008, pp. 363-369.

[6] G. Schuh, E. Rebentisch, M. Riesener, C. Mattern and P. Fey, "Method for the Evaluation and Adaptation of New Product Development Project Complexity," Procedia CIRP, vol. 4, 2017, pp.338–343.

[7] T. Bauernhansl, M. ten Hompel and B. Vogel-Heuser, "Industrie 4.0 in Produktion, Automatisierung und Logistik," Springer Fachmedien Wiesbaden, 2014.

[8] B. Guerineau, M. Bricogne, A. Durupt and L. Rivest, "Mechatronics vs. cyber physical systems: Towards a conceptual framework for a suitable design methodology," IEEE 11th France-Japan & 9th Europe-Asia Congress on Mechatronics (MECATRONICS)/17th International

Conference on Research and Education in Mechatronics, 2016, pp. 314-320.

[9] C. Berger, A. Hees, S. Braunreuther and G. Reinhart, “Characterization of cyber-physical sensor systems,” Procedia CIRP, 2016, pp. 638-643.

[10] L. Monostori, “Cyber-physical Production Systems: Roots, Expectations and R&D Challenges,” Procedia CIRP, 2014, pp.9–13.

[11] Y. Liu, Y. Peng, B. Wang, S. Yao and Z. Liu, “Review on cyber-physical systems,” IEEE/CAA Journal of Automatica Sinica, vol. 1, 2017, pp.27–40.

[12] R. Grussenmeyer and T. Blecker, “Requirements for the design of a complexity management method in new product development of integral and modular products,” International Journal of Engineering, Science and Technology, vol. 2, 2013, pp.132–149.

[13] J. R. Westphal, “Komplexitätsmanagement in der Produktionslogistik,“ Gabler Edition Wissenschaft, Deutscher Universitätsverlag, 2001.

[14] M. Götzfried, “Managing complexity induced by product variety” in Manufacturing companies: Complexity evaluation and integration in decision-making, 2013.

[15] H. Luczak and A. Fricker, “Komplexitätsmanagement — ein Mittel der strategischen Unternehmensgestaltung,“ Komplexität und Agilität, vol. 1-2, 1997, pp.309–323.

[16] G. Schuh, T. Potente, C. Thomas and A. Hauptvogel, “Steigerung der Kollaborationsproduktivität durch cyber-physische Systeme,“ in Industrie 4.0 in Produktion, Automatisierung und Logistik, vol. II, 2014, pp.277–295.

[17] L. Budde, “Integriertes Komplexitätsmanagement in produzierenden Unternehmen: ein Modell zur Bewertung von Komplexität,” Universität St. Gallen, 2016.

[18] J. Arnoscht, “Beherrschung von Komplexität bei der Gestaltung von Baukastensystemen“, Apprimus Verlag Aachen, 2011

[19] A. Usman and H. Mukhtar, ”Design Time Considedrations for Cyber Physical Systems,” IEEE 10th International Conference on Frontiers of Information Technology, 2012, pp.275–281.

[20] B. Schätz, ”The role of models in engineering of cyber-physical systems - challenges and possibilities,” in Tagungsband des Dagstuhl-Workshop, 2014, pp.91.

[21] M. Törngren and U. Sellgren, ”Complexity Challenges in Development of Cyber-Physical Systems,” in Principles of Modeling, 2018, pp.478–503.

[22] S. Serdarasan, ”A review of supply chain complexity drivers,” Computers & Industrial Engineering, vol. 3, 2013, pp.533–540.

[23] G. Schuh and H.-P. Wiendahl, “Komplexität und Agilität,“ Springer Berlin Heidelberg, 1997.

[24] W. Vogel and R. Lasch, ”Complexity drivers in manufacturing companies: a literature review,” Logistics Research, vol. 1, 2016, pp.1399.

[25] D. Adam, “Komplexitätsmanagement,“ Gabler Verlag, 1998,

[26] G. Schuh, “Produktkomplexität managen,“ Carl Hanser Fachbuchverlag, 2014

[27] M. Gießmann, “Komplexitätsmanagement in der Logistik,“ Eul, 2010.

[28] A. Grübner, “Bewältigung marktinduzierter Komplexität in der industriellen Fertigung,“ Lang, 2007.

[29] M. Kissel, “Mustererkennung in komplexen Produktportfolios,“ Technische Universität München, 2014.

[30] K.-P. Schoeneberg, “Komplexitätsmanagement in Unternehmen,“ Springer Gabler, 2014.

[31] C. Bliss, “Management von Komplexität,“ vol. 35, Gabler Verlag, 2013.

[32] M. Broy, “Cyber-Physical Systems — Wissenschaftliche Herausforderungen Bei Der Entwicklung,“ in Cyber-Physical Systems, Springer , 2010, pp.17–31.

[33] O. Niggemann, G. Biswas, J. Kinnebrew, H. Khorasgani, S. Volgmann and A. Bunte, ”Data-Driven Monitoring of Cyber-Physical Systems Leveraging on Big Data and the Internet-of-Things for Diagnosis and Control,” 2015, pp.185-192.

[34] A. Keuper, C. Dölle, M. Riesener and G. Schuh, ”Complexity-Oriented Description of Cyber-Physical Systems,” pp.602–610.

[35] M. E. Porter and J. E. Heppelmann, ”How smart, connected products are transforming competition,” Harvard business review, vol. 11, 2014, pp.64–88.

[36] G. Schuh, M. Riesener and C. Mattern, ”Approach to evaluate complexity in new product development projects,” International Journal of Design & Nature and Ecodynamics, vol. 4, 2016, pp.573–583.

[37] V. Gunes, S. Peter, T. Givargis and F. Vahid, “A Survey on Concepts, Applications, and Challenges in Cyber-Physical Systems,” KSII Transactions on Internet & Information Systems, vol. 8, 2014.

[38] A. Fink, O. Schlake and A. Siebe, “Erfolg durch Szenario-Management: Prinzip und Werkzeuge der strategischen Vorausschau,“ Campus-Verl., 2002.