Knowledge engineering and machine learning for design and use in cyber-physical environments

Michael Walch
University of Vienna
Universittsring 1
1010 Vienna, Austria

Abstract

The demand to make human knowledge and design explicit emerges when cyber-physical systems (CPS) are engineered with people and business aspects in the loop. While there is no doubt about the usefulness of human knowledge and design throughout the life-cycle of CPS, there is a question about how human knowledge and design is decomposed in artifacts and how these artifacts are related. In particular, one problem is to connect conceptual models and operation environments that realize CPS. The former focuses on encoding human knowledge and design about people, businesses, and CPS using semi-formal concepts that can be executed through procedures for sequential semantics, while the latter surrounds continuous-time models and CPS that exist in the concurrent physical world at run-time. Connecting both aspects is necessary throughout the life-cycle of CPS. By connecting both aspects in an intelligent manner, s*IoT is able to support the life-cycle of CPS. Therby, s*IoT supersedes the approach of developing application-specific interfaces between conceptual models and operation environments. Rather, s*IoT employs the semantic web stack to reduce the human effort for developing application-specific interfaces. While this is a promising approach, the question is if the integration of machine-learning approaches offers additional benefits for s*IoT, as machine-learning approaches can presumably further eliminate human effort associated with technologies from the semantic web stack. This paper presents an arguable opinion about the issue.

Introduction

While most cyber-physical systems (CPS) are intended to enhance the capabilities of people and businesses, this is a problem because it is difficult for CPS to know people's and businesses' requirements (Sowe et al. 2016). Making human knowledge and design accessible can help CPS to make intelligent decisions and achieve their goals which are ultimately the goals of people and businesses. Conceptual modeling is an approach that has the potential to make human design and knowledge explicit in a semi-formal manner that can be understood by humans at design-time and CPS at run-time. As a consequence, specializing the conceptual modeling approach is necessary to enable CPS with people and businesses in the loop. Correspondingly, the s*IoT conceptual modeling approach has been proposed to bring together in an intelligent manner (1) conceptual models that decompose human knowledge and design and (2) operation environments that provide abstract functional capabilities of CPS (Walch and Karagiannis 2019). The result thereof are "smart" models that can be understood by humans and CPS.

Connectivity between conceptual models and operation environments can be engineered by different means throughout the life-cycle of CPS. One option is to develop conceptual models and operation environments by hand, which means that different stakeholders have to invest a great amount of effort. Connecting these manually developed artifacts is possible by developing application-specific interfaces, which again requires human effort for each interface. Another option is to employ the semantic web stack to connect conceptual models and operation environments. As the semantic web stack provides benefits for topics that require diversity, synthesis, and definiteness (Janowicz et al. 2014), technologies from the semantic web stack are adopted in the current version of s*IoT. Likewise, ontologies and reasoning are employed by the s*IoT modeling method and tools. This enables to further decompose conceptual models step-wise into elements with formal semantics that can be matched to the formal semantics of operation environments. While employing the semantic web stack allows for the elimination of a large portion of manual work, some aspects still have to be largely developed by hand in a labor-intensive and error-prone process that has become a key bottleneck (Doan et al. 2004). Therefore, a third potential option is to employ machine learning. Thereby, the focus is on opportunities for advanced automation.

The methodology of this paper is to present an arguable opinion about combining machine learning with knowledge engineering. In particular, a potential update for s*IoT is presented by researching the opportunities of machine learning. Therefore, three cases are discussed on the topic of automating the connection between conceptual models and operation environments. The goal is to describe a direction along which future research can progress. This direction is framed by the conceptual framework of specializing the design science paradigm with a model-based approach. Additionally, a meta-level view is applied that considers the resulting models as systems under study. Research questions

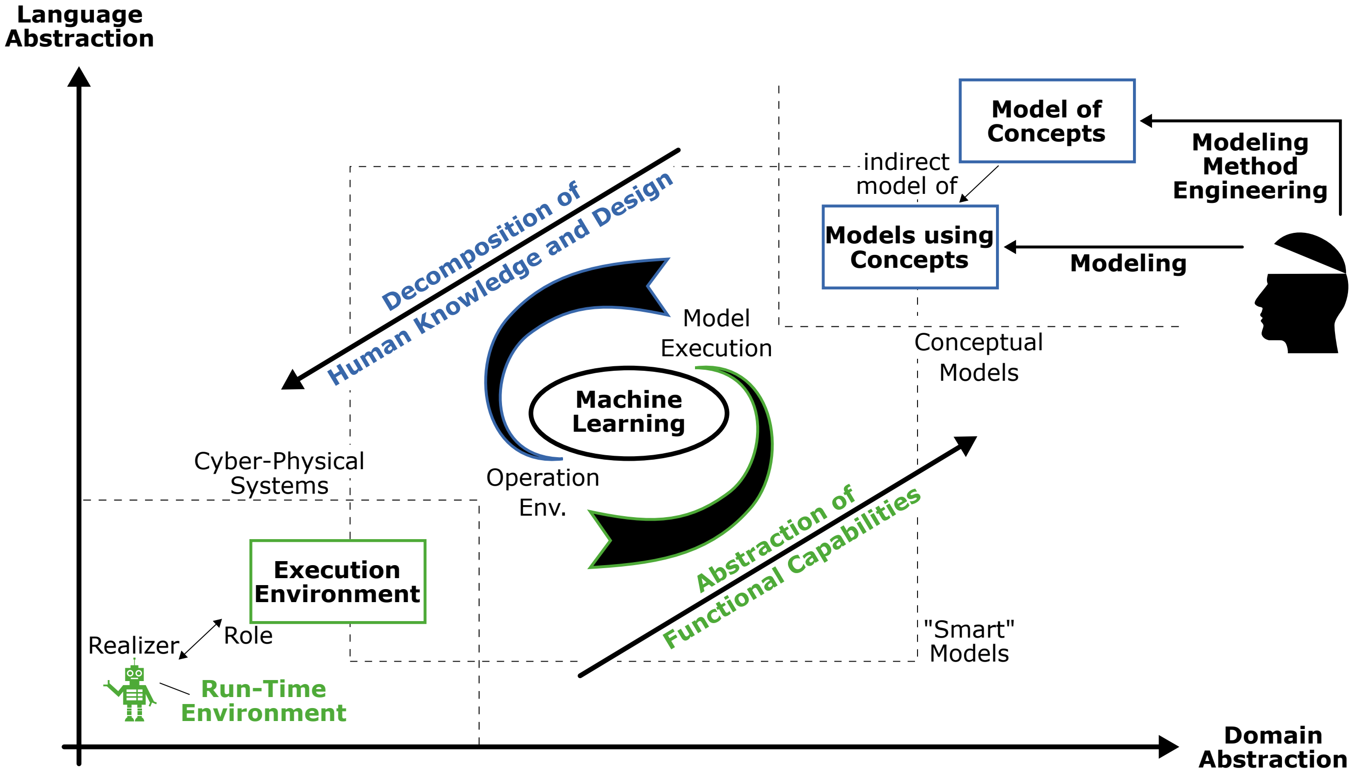

Figure 1: Topic for which the applicability of machine learning is analyzed.

for this paper are to map the opportunities of machine learning for connecting conceptual models and operation environments, to structure information from concrete cases in which machine learning is needed, and to conclude future research directions. To answer the research questions, the method of conceptual analysis is applied. An analysis of the results in terms of strengths, weaknesses, opportunities and threats (SWOT) is conducted for validation purposes.

Following the introduction, the paper is structured in five sections. First, foundations and related work are summarized on conceptual modeling, CPSs, and their connection. Afterwards, the s*IoT conceptual modeling approach is introduced briefly, also in terms of how it benefits from employing the semantic web stack. Based on these two sections, an update is provided on how s*IoT can be combined with machine learning. The results are critically reflected in a discussion section before the conclusion.

Foundations and Related Work

Figure 1 shows the topic addressed by the s*IoT conceptual modeling approach. This topic is analyzed in this paper regarding the applicability of machine learning. Therefore, foundations of s*IoT and related work are briefly discussed.

The topic under scrutiny can be structured in three main parts: conceptual modeling, CPS, and the connection between the two. Conceptual modeling can be employed in a distilling cycle to make human knowledge and design explicit (Karagiannis, Buchmann, and Walch 2017). The result is explicit knowledge and design that is decomposed by human-oriented and machine-oriented representations. These representations are conceptual models. Conceptual models are semi-formal in the sense that they can be processed by ICT systems but also contain semantics that require human interpretation. To build conceptual models, modeling methods and tools are required. A linguistic, procedural, and algorithmic abstraction of conceptual models, their modeling methods, and modeling tools is provided by metamodels (Karagiannis and Kühn 2002). Together, metamodels and models support the engineering of knowledge and design in an agile cycle (Karagiannis 2015). Thereby, engineering can be viewed as a task of assembling representational components rather than axiom-writing (Clark et al. 2001). As a consequence, an engineer is not always necessary when knowledge and design of subject matter experts is made explicit, as the latter can directly interact with conceptual models using representations familiar or intuitive to them. Cyber-physical systems are feedback systems involving cyber and physical components, which enables innovative applications for, e.g., Industrie 4.0, Society 5.0, and Smart Cities. The difference to traditional ICT systems is that there is no clear separation, but rather an intersection of physical processes and software (Shi et al. 2011). However, modeling is required to enable different multidisciplinary teams to work together on the problem of designing and using CPS. As a consequence, CPSs create new challenges for modeling not covered by traditional modeling methods for ICT systems (Derler, Lee, and Sangiovanni-Vincentelli 2011; Sharma et al. 2014). That is because traditional ICT systems rely on models that encoding knowledge and design through sequential steps, while CPSs are deeply rooted in the physical world, which requires continuous-time models that are working with, e.g., solvers that numerically approximate the solutions to differential equations. Connectivity between conceptual models and CPS requires an integration of design-time and run-time aspects. Therefore, conceptual models can be extended by operational semantics (Lehmann et al. 2010) on the one end of the connection. On the other, CPS realize a run-time environment for executable models. The run-time environment can be encapsulated by an execution environment that provides interfaces on the same level of abstraction as executable models. Together, run-time environment and execution environment make up the operation environment of executable models. However, in reality the connection between executable models and execution environments is a complex issue, as there is no fixed point of alignment (Walch and Karagiannis 2019).

After this short introduction to the foundations of the topic under scrutiny, related work for the connection between conceptual models and operation environments is discussed. Regarding the execution of conceptual models, there is a benefit for conceptual models that are cognitively adequate for humans and processable by machines, as such models could, e.g., enable communication and collaboration, support decision makers through analysis and simulation, and automate enterprise operations through model execution (Hinkelmann et al. 2018). To harness these benefits, formal semantics of conceptual models are essential (Hinkelmann et al. 2016). Examples of conceptual models that are extended by formal operational semantics are model types like UML which are extended by fUML (Dévai et al. 2015), SysML which requires dedicated execution environments (Wolny 2017), and BPMN which can be put to use by workflow engines (De Giacomo et al. 2017). However, only few types of models can be executed (Thalheim 2018), which is a problem due to agile and fast changing modeling requirements and especially considering that CPS could be employed to operationalize models. Regarding the abstraction of CPS in conceptual models, the PRINTEPS project is a recent example (Morita et al. 2018). PRINTEPS commits to the robot operating system (ROS) as an abstraction of the run-time environment that different robots offer. This execution environment is reflected in domain-specific conceptual models that are extended with operational semantics for model execution. Further model abstraction allows for conceptual models that are intuitive for domain experts. In PRINTEPS, some of the abstraction and decomposition mechanisms that relate different conceptual models and ROS are automated. However, one problem is that the commitment to ROS is not applicable to all kinds of CPS, especially as CPS architectures change from hierarchical to service-oriented (Foehr et al. 2017; Gruettner, Richter, and Basten 2017).

The s*IoT Conceptual Modeling Approach

The s*IoT conceptual modeling approach has been proposed due to new requirements that emerged from AMME and changing architectures of CPS. In particular, the problem manifests itself when conceptual models are put to use, as the manual alignment of conceptual models and operation environments by application-specific interfaces requires human development effort that does not scale. To alleviate this issue, the s*IoT modeling method integrates technologies from the semantic web stack.

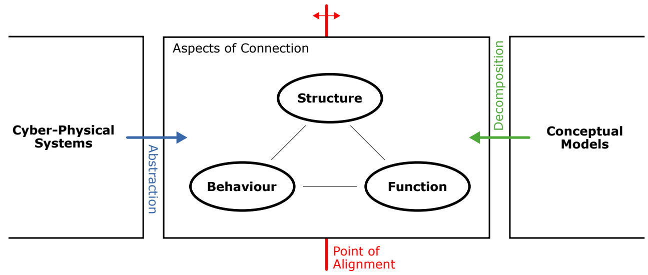

Figure 2: Connection of conceptual models and operation environments supported by s*IoT.

Figure 2 shows aspects of connecting conceptual models and operation environments. Structural aspects refer to components and their relationships, functional aspects to the hierarchy of abstract roles and concrete realizations (i.e., goals and measurable effects), and behavioural aspects to the performance over time. In s*IoT, these aspect are supported by technologies from the semantic web stack. The resulting benefit is that the point of alignment is not fixed for specific applications, but rather it allows for flexibility, intelligence, and automation when connecting different kinds of conceptual models and operation environments. This is possible because the semantic web stack provides technologies that elevate the connection from application-specific interfaces to semantic mappings between the involved elements.

An example for a concrete application case of s*IoT is the matching of execution requirements from conceptual models to functional capabilities of CPS. Therefore, execution requirements of conceptual models have to be extended by semantic information to be matched to semantic services of CPS that are being discovered. Likewise, it is possible to model human knowledge and design about, e.g., an Industrie 4.0 production process, to annotate the resulting conceptual model with formal semantics, and to discover suitable services of CPS for model execution. Another example is the configuration of CPS based on requirements from conceptual models.

s*IoT and Machine Learning

To improve the s*IoT conceptual modeling approach, the benefits of machine learning are examined with regards to the issue of connecting conceptual models and operation environments. Three cases are presented in which s*IoT is applied. As the current version of the s*IoT modeling method and tools already makes use of technologies from the semantic web stack, the results are ”smart” models. In the three presented cases existing ”smart” models were extended by machine learning on a proof-of-concept basis.

Case One - Recognizing the Structure of Cyber-Physical Environments: In this case, the s*IoT modeling method and tools are applied to model a mock-up coffee making process and to execute that process in a cyber-physical environment that contains a robotic arm and coffee ingredients. To enable model execution, the structure of the cyber-physical environment is abstracted to the modeling layer. This is done manually by humans who created an ontology that extends the model of the mock-up coffee making process. The ontology contains information about objects in the cyber-physical environment like the coffee ingredients and the robotic arm, e.g., their x, y, and z positions. By combining all these elements in ”smart” models, the execution of the process becomes possible. Currently, the options that machine learning provides to this case are being evaluated. In particular, image recognition was used to update the ontology of objects based on real-time data. As a consequence, it is feasible that no manual intervention would be necessary in case the amount, position, or size of coffee ingredients changes, if machine learning approaches were to be integrated in s*IoT.

Case Two - Reasoning Functional Capabilities from Structure: In this case, the s*IoT modeling method and tools are applied to model functional capabilities of CPS. By using technologies from the semantic web stack, it is possible to reason functional capabilities of CPS from their structure. This requires knowledge engineers and domain experts to define the relation between function and structure, e.g., a robotic vehicle that can drive and steer has - among other things - two independent motors, wheels, and motor controllers. Currently, it is evaluated how this kind of reasoning can be supported by machine learning. Previously, the structure of a CPS had to be modeled by hand, as well as the relation between function and structure. Existing models of that kind were used to structure training sets for machine learning. Based on these training sets and machine learning technologies, it was possible to identify the structure of CPS from images and to classify CPS by their functional capabilities. A thorough comparison of benefits and drawbacks between the currently employed technologies from the semantic web stack an machine learning should be able to provide further insights.

Case Three - Modelers Assistant based on CPS Behaviour: In this case, the goal is to reduce the time and cognitive effort modelers spent, by providing intelligent assistants to modelers. These assistants should actively classify the modeler’s activities, predict future tasks, and proactively perform those tasks automatically (Panton et al. 2006). One example for this is case-based reasoning, where knowledge of previously experienced cases is used to propose solutions to changing requirements (Martin and Hinkelmann 2018). Currently, s*IoT offers no intelligent assistants for modelers. Therefore, machine learning is explored to fill this gap. The concept is that, as processes are being put to use by CPS, the feedback from CPS behaviour can be collected. This feedback can be used in machine learning to classify good and bad patterns of processes. Based on this classification, it should be possible to predict how newly modeled processes will behave. This prediction could be made available to the modelers of processes during their modeling task. After reviewing the necessary machine learning technologies, it is feasible that progress can be made towards developing a prototype for this case.

Discussion

A conclusive SWOT analysis is an effective approach for rationalization. Therefore, a SWOT analysis is conducted to validate the opinion formed in this paper about the potential of machine learning for s*IoT. Furthermore, the SWOT analysis generalizes from the three presented cases.

The strengths of machine learning for s*IoT are: (1) Human effort associated with technologies from the semantic web stack can be reduced. This allows for greater flexibility when connecting conceptual models and CPS. (2) New application scenarios become possible as modeling methods and tools evolve. (3) The quality of conceptual models and CPS is increased as machine learning enables a tighter connection between the two. The weaknesses of machine learning for s*IoT are: (1) Additional complexity is introduced as the workload of human stakeholders gets automated. New sources of error and a lack of tractability are a problem for modeling method engineers and modelers. (2) Machine learning requires human effort to select machine learning paradigms, prepare training data, and supervise learning algorithms. (3) The applicability of machine learning is related to the availability of training data. This is somewhat contradictory to conceptual modeling which is often used to capture innovative and creative ideas. The opportunities of machine learning for s*IoT are: (1) Collaboration is facilitated among the machine learning community, the conceptual modeling community, and the CPS community. This creates new chances for research, application, and education. (2) The dissemination of the s*IoT modeling method and tools can be accelerated by embracing the current trend of machine learning. (3) By automating human effort, human resources become available. These human resources can be used for creative and innovative tasks. The threats of machine learning for s*IoT are: (1) Machine learning is a complex topic and human resources are sparse. Furthermore, projects that involve machine learning are often difficult to plan due to the lack of previous results. (2) It is possible that the trend of machine learning changes as it did before. The danger is to focus on soon to be outdated aspects of machine learning. (3) A social and ethical perspective has to be considered when tasks of humans are automated. Furthermore, all kinds of risks have to be considered when humans are replaced by automation.

Conclusion

Knowledge engineering is necessary in the life-cycle of CPS, as human knowledge and design is essential for CPS with people and businesses in the loop. In the life-cycle of CPS, conceptual models are knowledge engineering artifacts that have to be connected to operation environments. Connecting conceptual models and operation environments is elevated by s*IoT from an application-specific development effort towards a systematic approach that makes use of technologies from the semantic web stack. While this is a promising endeavor, this paper is exploring advanced options for elevating the connection of conceptual models and operation environments even further. In particular, the reemerging trend of machine learning is evaluated regarding benefits it could provide to s*IoT.

Three cases are presented in which machine learning supports connecting conceptual models and operation environments. In the first, the recognized structure of a cyber-physical environment is made available for conceptual models. In the second, functional capabilities of CPS are classified based on the structure of CPS components. In the third, the behaviour of processes is predicted during modeling based on their previous execution by CPS. Preliminary results from the three cases are promising. The next step is to integrate the machine learning technologies used in those three cases as part of the s*IoT modeling method and tools, which will allow modelers unfamiliar with the technologies to make use of them. Furthermore, this allows other modeling method engineers to integrate them into their modeling methods as well.

References

Clark, P.; Thompson, J.; Barker, K.; Porter, B.; Chaudhri, V.; Rodriguez, A.; Thomere, J.; Mishra, S.; Gil, Y.; Hayes, P.; et al. 2001. Knowledge entry as the graphical assembly of components. In *Proceedings of the 1st international conference on Knowledge capture*, 22–29. ACM.

De Giacomo, G.; Oriol, X.; Estanol, M.; and Teniente, E. 2017. Linking data and bpmn processes to achieve executable models. In *International Conference on Advanced Information Systems Engineering*, 612–628. Springer.

Derler, P.; Lee, E. A.; and Sangiovanni-Vincentelli, A. L. 2011. Addressing modeling challenges in cyber-physical systems. Technical report, California Univ. Berkeley Dept. of Electrical Engineering and Computer Science.

Dévai, G.; Karácsony, M.; Németh, B.; Kitlei, R.; and Kozsik, T. 2015. UML model execution via code generation. In *EXE@ MoDELS*, 9–15.

Doan, A.; Madhavan, J.; Domingos, P.; and Halevy, A. 2004. Ontology matching: A machine learning approach. In *Handbook on ontologies*. Springer. 385–403.

Foehr, M.; Vollmar, J.; Calà, A.; Leitão, P.; Karnouskos, S.; and Colombo, A. W. 2017. Engineering of next generation cyber-physical automation system architectures. In *Multi-Disciplinary Engineering for Cyber-Physical Production Systems*. Springer. 185–206.

Gruettner, A.; Richter, J.; and Basten, D. 2017. Explaining the role of service-oriented architecture for cyber-physical systems by establishing logical links. In *Information Systems (ECIS), 25th European Conference on*.

Hinkelmann, K.; Gerber, A.; Karagiannis, D.; Thoenssen, B.; van der Merwe, A.; and Woitsch, R. 2016. A new paradigm for the continuous alignment of business and IT: Combining enterprise architecture modelling and enterprise ontology. *Computers in Industry* 79:77 – 86. Special Issue on Future Perspectives On Next Generation Enterprise Information Systems.

Hinkelmann, K.; Laurenzi, E.; Martin, A.; and Thönssen, B. 2018. *Ontology-Based Metamodeling*. Cham: Springer International Publishing. 177–194.

Janowicz, K.; Van Harmelen, F.; Hendler, J. A.; and Hitzler, P. 2014. Why the data train needs semantic rails. *AI Magazine*.

Karagiannis, D., and Kühn, H. 2002. Metamodelling platforms. In Bauknecht, K.; Tjoa, A. M.; and Quirchmayr, G., eds., *E-Commerce and Web Technologies*, 182–182. Berlin, Heidelberg: Springer Berlin Heidelberg.

Karagiannis, D.; Buchmann, R.; and Walch, M. 2017. How can diagrammatic conceptual modelling support knowledge management? In *25th European Cenference on Information Systems, ECIS 2017*, Proceedings of the 25th European Conference on Information Systems (ECIS), 1568–1583.

Karagiannis, D. 2015. Agile modeling method engineering. In *Proceedings of the 19th Panhellenic Conference on Informatics*, PCI '15, 5–10. New York, NY, USA: ACM.

Lehmann, G.; Blumendorf, M.; Trollmann, F.; and Albayrak, S. 2010. Meta-modeling runtime models. In *International Conference on Model Driven Engineering Languages and Systems*, 209–223. Springer.

Martin, A., and Hinkelmann, K. 2018. *Case-Based Reasoning for Process Experience*. Cham: Springer International Publishing. 47–63.

Morita, T.; Kashiwagi, N.; Yorozu, A.; Walch, M.; Suzuki, H.; Karagiannis, D.; and Yamaguchi, T. 2018. Practice of multi-robot teahouse based on PRINTEPS and evaluation of service quality. In *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, 147–152. IEEE.

Panton, K.; Matuszek, C.; Lenat, D.; Schneider, D.; Witbrock, M.; Siegel, N.; and Shepard, B. 2006. Common sense reasoning–from cyc to intelligent assistant. In *Ambient Intelligence in Everyday Life*. Springer. 1–31.

Sharma, A. B.; Ivančić, F.; Niculescu-Mizil, A.; Chen, H.; and Jiang, G. 2014. Modeling and analytics for cyber-physical systems in the age of big data. *ACM SIGMETRICS Performance Evaluation Review* 41(4):74–77.

Shi, J.; Wan, J.; Yan, H.; and Suo, H. 2011. A survey of cyber-physical systems. In *Wireless Communications and Signal Processing (WCSP), 2011 International Conference on*, 1–6. IEEE.

Sowe, S. K.; Simmon, E.; Zettsu, K.; de Vaulx, F.; and Bojanova, I. 2016. Cyber-physical-human systems: Putting people in the loop. *IT professional* 18(1):10–13.

Thalheim, B. 2018. Conceptual model notions–a matter of controversy: Conceptual modelling and its lacunas. *Enterprise Modelling and Information Systems Architectures* 13:9–27.

Walch, M., and Karagiannis, D. 2019. How to connect design thinking and cyber-physical systems: the s*IoT conceptual modelling approach. In *Proceedings of the 52nd Hawaii International Conference on System Sciences (HICSS-52)*, forthcoming.

Wolny, S. 2017. A runtime model for SysML. *Doctoral College Cyber-Physical Production Systems* 43.