A Conceptual Modeling Framework for Evaluation of Cyber-Physical Systems based on Applied Category Theory and Metamodeling

Vimal Kunnummel[1]

Supervisor: o. Univ.-Prof. Prof.h.c. Dr. Dimitris Karagiannis[1]

[1]RG Knowledge Engineering, Faculty of Computer Science, University of Vienna
Währingerstraße 29, A-1090 Vienna, Austria
vimal.kunnummel@univie.ac.at

Abstract. Cyber-Physical Systems (CPS) are interconnected, computational control systems which directly interact with the non-deterministic physical world. To manage the confronted uncertainty when dealing with non-deterministic environments, a constant feedback loop of monitoring the environment and consequent adjustment of the system is obligatory. Components of a CPS as well as the communication between its components are prone to malfunctioning leading to system failures. Therefore, to enable an effective integration of CPSs into arbitrary business processes, a conceptual modeling framework which enables the evaluation of the capabilities and functionalities of CPSs is needed. In this paper, such a conceptual modeling framework based on applied category theory is proposed to model CPSs in a given environment.

Keywords: Cyber-Physical Systems, Metamodeling, Applied Category Theory, Modeling Languages.

1 Introduction

"Cyber-Physical Systems (CPSs) are, following the definition given in [1], physical and engineered systems whose operations are monitored, coordinated, controlled and integrated by a computing and communication core. Just as the internet transformed how humans interact with one another, Cyber-Physical Systems will transform how we interact with the physical world. Examples of CPS include medical devices and systems, aerospace systems, transportation vehicles, defense systems, robotic systems, process control, factory automation, building and environmental control and smart spaces. CPSs interact with the physical world, must operate dependably, safely, securely and efficiently in real-time. CPS can be considered to be a confluence of embedded systems, real-time systems, distributed sensor systems and controls augmented by the cyber capabilities.

J. Ralyté and Y. Wautelet (Eds.): PoEM 2018 Doctoral Consortium Papers, pp. 74-85, 2018.

CPSs bring together the discrete and powerful logic of computing to monitor and control the continuous dynamics of physical and engineered systems" [1].

One of the main challenges during the ongoing proliferation of CPSs, is the need to create appropriate models of these systems, their interactions with the environment and the interdependencies of their components to understand their combined behavior. Various modeling approaches are discussed in the literature to model CPSs, e.g. using discrete and synchronous models of reactive computation, asynchronous models of computation, timed models of computation and continuous-time models which include differential equations and state-space equations [2,3,4]. But all these models have fundamental limits which stem from the non-deterministic behavior of the physical world which can never be accurately modeled [5,6].

So when a model of a system is used, we need to be aware of its advantages and its restrictions as each model is an abstraction of the real system and using abstraction inherently results in ignoring those parts of the system not included in the model. Each model has therefore a specific purpose. In summary, to understand complex systems like CPSs the choice of the appropriate model, which depends on the specific purpose, is crucial.

This leads to the problem I would like to address in this research project: a conceptual modeling framework which focuses on the evaluation of the capabilities and functionalities of a CPS is still not explored in the literature. Such a model-based evaluation framework would allow to check if the CPSs are available for a specific use-case as they could then be integrated into an arbitrary business process.

Capabilities of a CPS are meant as the set of methods the system has at its disposal to perform a task. For example, a robotic arm would be capable of moving its arm and grabbing objects with that movable arm. A necessary precondition for having these capabilities of moving the arm and grabbing objects, is the positive working condition of all the components which are thereby involved. Using those capabilities, the robotic arm is capable to exert various functions, meaning with the same set of capabilities, a CPS could have different functional capabilities or functionalities. The same robotic arm with the capabilities of grabbing and moving its arm, could be used as a burger-making robotic arm or as a coffee-making robotic arm, meaning it can be deployed in different scenarios and could be serving different functions in those different scenarios.

This paper is structured as follows: After a brief introduction given in section 1, related work regarding modeling of CPSs is presented in section 2. In section 3, the problem statement and the research questions are presented. Furthermore, initial results from the ongoing research on integrating CPSs into business processes, in particular using the s*IoT modelling method [16], is presented to position the research questions addressed in this paper into a broader picture. In section 4, the research methodology is outlined. In section 5, the research approach, preliminary results and the unique contribution of this work is discussed.

2 Related Work

What makes modeling of CPSs challenging is the fact that these systems deal with physical processes which are traditionally modeled using continuous-time models of dynamics (e.g. differential equations) but they also incorporate computational elements which are modeled using finite state machines, dataflow models or synchronous/reactive models [2]. Therefore, conventional models are not fully capable to correctly describe the behavior of CPSs as different models are used to model the physical and computational processes.

Research has been done to develop models which combine discrete and continuous models into one single model which should be able to model hybrid systems like CPSs leading to hybrid automata [7,8]. Using hybrid automata, verification of the properties of a system is reduced to reachability problems but as these can become very complex, solutions are generally obtained only for specific cases and for certain subsets like timed automata [9]. Another issue which is critical for CPSs is the need to model the influence of time which adds another dimension of complexity [10,11]. Research on modeling of CPSs in control and electrical engineering fields focus on these issues of bridging the gap between discrete and continuous-time models to obtain holistic models of the behavior of the CPSs. Other modeling approaches include using agent-based modeling [12] and event-based modeling [11]. These modeling approaches are needed to design CPSs and understand their behavior as they concentrate on the run-time aspects of the systems.

On the other hand to put CPSs into usage in a productive business environment, one has to be able to also model the enterprise requirements and to match those requirements with the capabilities of the CPSs. This is a challenging task, as enterprise models are defined using conceptual models lacking formal semantics and are designed to be interpreted by humans [13,14,15]. And humans also operationalize the enterprise models, meaning humans interpret and perform the given tasks in the models. To achieve the high level of automation as envisioned in age of the digital transformation, machines should also be able to interpret and operationalize conceptual models. To reach that goal the current existing gap between enterprise models only to be interpreted by humans and machine-interpretable models of the CPSs must be bridged. One attempt to tackle this issue is the s*IoT modelling method [16,17]. In this modeling method, a service oriented architecture is used to abstract the functional capabilities of CPSs by using a microservices portal [16]. The goal of the s*IoT methodology is to align enterprise models with CPSs and to create "smart" models using a modeling method tool modeled by a metamodel [17]. The s*IoT research methodology will also be used as the research methodology in this paper.

3 Problem Statement and Research Questions

Cyber-physical systems are all about integration: integration of physical and cyber components, integrating those systems into existing infrastructures but also integration of the systems into business and social contexts. There are various challenges which need to be addressed when integrating these systems. Interoperability of systems must be addressed as the systems and their underlying technologies are very heterogeneous. Platform compatibility should therefore be ensured. Security and privacy issues must also be considered as a growing number of interconnected devices increase the potential targets for malicious actors.

Therefore CPSs need to be resilient and robust, meaning functionalities should not be compromised in case of the sudden malfunctioning of some components of the CPS. There are also other reasons for the malfunctioning of CPS components: in many CPSs, parts are dependent on batteries, so energy consumption and efficiency is always an issue. This could lead to sudden system failures in case the battery runs out. Malfunctioning of the system could also result from wrong measurements. This case must also be taken into account as the physical world is inherently non-deterministic.

So the question of the reliability of the sensed data is an important one in CPSs. Therefore in practice, one has to be able to model these possible malfunctions of CPS components but also the reliability of the sensed data in the conceptual CPS model and should furthermore be able to infer the consequences of such an event. Those other components which depend on that malfunctioning node should be subsequently identified. To ensure this, a profound understanding of the dependencies of the various parts of the CPS is needed to react to sudden events. So again the notions of integration and composition of the system components are paramount to address these issues.

Apart from the technical perspective given above, also other non-functional requirements must be taken into consideration as our goal is to eventually enable integration between the requirements defined in enterprise models and the capabilities of the CPSs. The formal modeling language developed in this work must enable to define models of the CPS incorporating the interdependencies of the components, the hierarchical relations and other dependencies. As mentioned above, in the scenarios where the CPSs are deployed to perform some functions, their success depend on their capabilities which subsequently depend on the components which are prone to malfunctioning and thereby it is crucial to have a monitoring system which reports any changes.

The s*IoT methodology uses a three-layer architecture to tackle the connectivity issue between enterprise models and the CPSs. The three layers are the scenario layer, the modeling layer and the run-time environment of the CPSs [17]. In the scenario layer a concrete use-case in a business-context could be defined, e.g. following a Design Thinking approach and then conceptualized into an enterprise model. To operationalize this enterprise model using CPSs, an abstract model of the functional capabilities of the CPSs must be present in the modeling layer. On the modeling layer those conceptual models and the CPS models are matched. To enable this complex matching, it should be possible to evaluate the capabilities of the CPSs on this abstract modeling layer.

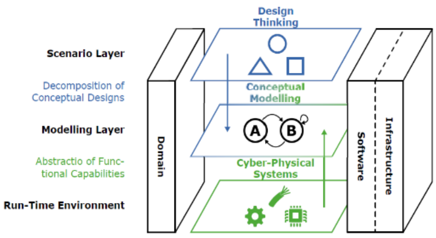

Fig. 1. The three layer s*IoT architecture. Adapted from [17].

This is the focus of this work as part of the broader attempt to develop a framework to integrate CPSs into business processes. Fig. 1 shows the three layered s*IoT architecture. Therefore the goal of this research project is to develop a modeling language which inherently captures the notions of integration and composition of various objects as this is central for modeling CPSs and their interactions with the environment and which enables to evaluate the CPSs. The following research questions which will be addressed in the project are identified to tackle the challenges mentioned above and to define a modeling language:

1. RQ1: What are the requirements for a modeling language which models the capabilities, functionalities and structure of CPSs?
2. RQ2: Which notions of structures and operations thereon must be included in the modeling language to be able to create formal models of CPSs, their interactions leading to state transitions and their interdependencies?
3. RQ3: Is it possible to evaluate the initial conditions which have to be met by a CPS using formal conceptual models to enable integration of CPSs into arbitrary business processes?

4 Research Methodology

The design-science based research approach according to Hevner and Chatterjee [18] is an established research methodology in the field of information systems. The s*IoT methodology [17] based on the design-science research approach will be applied in this project. As shown in [17], this methodology can be used in a variety of research projects. In a first step, the common structures and dependencies between the components of CPSs will be identified and subsequently formally described using applied category theory. These could be hierarchies, relations, types of objects, connections between different types, preconditions for state transitions, post conditions after state transitions and many more.

A formal mathematical framework is needed, as it provides the tools to deliver proofs of the workability of the used model. Based on this mathematical framework a modeling language is developed to evaluate CPSs. The developed modeling language is evaluated through a prototypical implementation of it in a software tool using the OMiLAB artifacts [19,20]. The prototype will be iteratively updated to improve the implementation and in case changes are needed to the modeling language to fulfill the requirements, the language will be refined. To validate the obtained artifacts, experiments are conducted in a laboratory using concrete use-cases. Scenarios in a smart home environment will be used as use-cases. In a smart home environment, interconnected sensors, cameras and other data collecting devices are coupled with data processing capabilities making it into a CPS. These empirical experiments are needed to show that the proposed modeling framework is not only of theoretical interest but to use in practical scenarios.

5 Research Approach and Preliminary Results

In this project, a modeling language based on applied category theory is proposed [21]. Abstraction is needed to find commonalities between different structures and to organize them [22,23,24,30]. Category theory is the mathematical theory of structures and structure-preserving operations between them and is ideal for such abstraction. It is the mathematical framework to study structures and its changes. In category theory, we can define things with similar structures and properties to be objects in a category. Furthermore on can define relationships between those objects as morphisms in that category: these could be functions, relations and other arbitrary operations which fulfill some rules. These morphisms must fulfill the associativity law, furthermore identity morphisms must be defined.

Category theory offers a formal mathematical framework to combine a chain of morphisms into a new morphism and to discuss when they can be combined. This allows us to speak about composition and integration in a clear, formal way. So the question which preconditions must be met to compose two different systems, can be answered. The main goal of this project is to apply these mathematical tools to develop a modeling language to evaluate the capabilities and functionalities of CPSs. To achieve that goal the common requirements encountered in different systems must be identified and formalized while leaving possibilities to add new requirements for specific systems.

Initially, a modeling language based on category theory is developed to model CPSs. This modeling language is then implemented in a second step using the metamodeling platform ADOxx. Using the OMiLAB virtual and technical environments, the developed modeling language is then subsequently validated [19,20].

In the following section 5.1, category theory is introduced and in section 5.2, a brief introduction into metamodeling is given. In section 5.3, the technical environment, which will be used to validate the artifacts produced within this project will be described. As the project deals with CPSs, the obtained results should also be used on real world CPSs to show the practicability of the results.

5.1 Formal Ansatz

Category theory is used as the formal mathematical framework [22,23,24,30]. In particular, the importance of monoidal categories for modeling the CPSs will be stressed as they inherently include composition and parallelism or concurrency of processes.

The ability to model concurrency is central for any modeling language for CPSs. In the following a brief introduction into the most important concepts of category theory which will be used in this dissertation are presented in a phenomenological manner.

To obtain reliability and interoperability of systems, one needs to take care of the structure and coherence of the system components. This is the basic idea of category theory. A category in category theory includes several structures. It has a collection of objects, a collection of morphisms which relate the objects, and a method for combining a chain of morphisms into a single morphism. The important part is to make sure that these structures cohere in a natural way ensuring that they work together.

The central theme in category theory is that the relations or morphisms between objects in a category are the most relevant aspect. Objects are abstract "things", the relationships between those things make them valuable as they give those objects a common organizing principle.

In category theory [25], a morphism $f : a \rightarrow b$ in a category $\boldsymbol{C}$ has a domain a and a codomain b which are both objects of $\boldsymbol{C}$. The morphism f describes one of the many possible relations which could exist between the objects a and b in the context of $\boldsymbol{C}$. It can be easily seen that there must be an underlying directed-graph structure in a category where the objects are nodes and morphisms act as edges but a category is not a graph. There are some crucial differences.

First, there can be more than one morphism in either or both directions between objects, resulting a multigraph structure with arrows in both directions. Therefore the morphisms between two objects a and b are labeled and there is a notion of domain and codomain indicating the source and end of a morphism.

However, the most important difference between a category and a graph, is the notion of *composition*. In a category $\boldsymbol{C}$, we can define for a pair of morphisms of the following form, $f : a \rightarrow b$ and $g : b \rightarrow c$, a composition morphism $g \circ f : a \rightarrow c$ whose domain a is the domain of f and whose codomain c is the codomain of g. We notice that composition is only possible if the "type" of the codomain of a morphism and the domain of the subsequent morphism are the same.

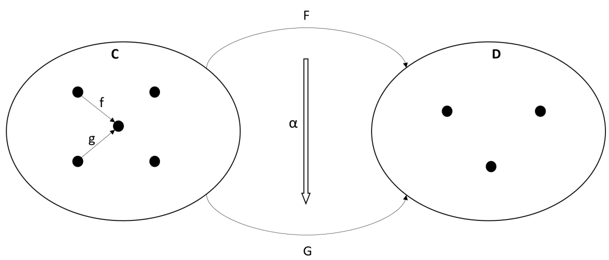

Fig. 2. Schematic of two categories **C** and **D**, two functors F and G from **C** to **D** and a natural transformation $\alpha : F \rightarrow G$. Category **C** contains two morphisms f and g, however, category **D** contains no morphisms, besides the identity morphisms.

Composition must also satisfy the associativity and identity axioms of category theory. The associative law imposes the following rule for three composable morphisms, $f : a \rightarrow b$, $g : b \rightarrow c$ and $h : c \rightarrow d$: the composition is order-independent, $h \circ (g \circ f) = (h \circ g) \circ f$.

Furthermore, for each object a there exists an identity morphism $id_a : a \rightarrow a$. This is the identity axiom. The following must always hold for arbitrary morphisms f and g defined on object a: $id_a \circ g = g$ and $f \circ id_a = f$.

The important result of having this notion of composition is that, there is an equivalency between many morphisms between objects starting from a start object to an end object and the composition of those intermediate morphisms from the start object to the end object along that path, as both lead to the same end object. This is not defined in graphs. In a category we can make statements about the composition of morphisms and furthermore about different paths from the same starting object to the same end object. This case leads to the notion of a commutative diagram.

A *functor* F is a mapping between categories. For example, given categories **C** and **D**, F maps the objects and morphisms in **C** to objects and morphisms in **D**. It sends objects to objects and morphisms to morphisms, but importantly by preserving identities and composition of morphisms. So a functor preserves the overall structure during the mapping. Morphisms between functors are called natural transformations. Given functors F and G, a natural transformation $\alpha : F \rightarrow G$ is mapping between those two functors which also ensures that the structures are preserved. Fig. 2 shows the schematic of two functors between two categories and a natural transformation between the two functors. It is often vital and important to model parallel systems which form a single global system. The concept of monoidal categories are needed to model this notion of parallelism.

The mapping from the mathematical foundations to their applications in concrete cases will be addressed in this project and is one source of the research questions. The focus will be to apply the powerful mathematical tools which are not yet established in the CPS domain but which will prove very useful to solve problems encountered in CPSs.

5.2 Conceptual Metamodeling Approach

The metamodeling approach defined in [26,27,29] identifies three parts of a modeling method: the modeling language that describes the syntax, semantics and notation, the modeling procedure that describes the methodology and the algorithms and mechanisms that provide the functionality to use and evaluate the models described in the modeling language. Metamodeling platforms like ADOxx enable the implementation of the modeling method. The modeling language developed in this project can thus be implemented as a software tool using the metamodeling platform and thereafter tested. This enables the validation of the modeling language by conducting experiments with real world CPSs in the OMiLAB laboratory which will be used as the technical environment.

The advantage of metamodeling is that it enables to abstract the invariants of the domain on a meta-level. Using a metamodeling platform, one can define a metamodel of the modeling method which can then be deployed as a software tool enabling the creation of models. Metamodels are therefore models of modeling methods. One follows five phases to develop the artifacts: Create, Design, Formalize, Develop and Deploy where the steps can be iterated to increase agility.

5.3 Technical Evaluation Environment

The OMiLAB [19,20,28] technical evaluation environment will be used to empirically test the modeling language in Smart Home Environment scenarios. In a Smart Home Environment, the environment is constantly sensed using sensors, cameras and smart devices. In such an interconnected environment, different CPSs like autonomous cars and an integrated intelligent Home Assistant can exchange data to make decisions. A physical model of a smart home environment including sensors, microcontrollers and other technical devices in the OMiLAB physical laboratory will serve as the execution environment for the evaluation experiments. Using different scenarios in this setting, the validity of the models created using the proposed modeling language is evaluated. The formal framework helps to identify and avoid inconsistencies and conflicts which could arise during the decision making of the CPSs.

Fig. 3. Simple schematic model of a Smart Home Environment.

Car Parking. A process could be induced due to the need of car parking. Once the car drives in, the smart environment senses if anything could stop this process. Examples could be some obstacles on the street. A process is initialized to sense these obstacles and the obtained data is matched with the data from the autonomous car. This matching is needed to minimize any risk during the parking. Once both the smart environment and the autonomous car agree that the parking is safe, the car drives into the garage.

Playground. Another scenario could be in a playground where kids play. Due to security concerns, it would be needed to check if the nearby swimming pool is closed and if the barbecue is turned off. There could be further threats like strangers passing by. To ensure that the process can be executed without any issues, the sensing capabilities of the smart environment are used. So the constant stream of data obtained by the sensors must by processed to detect any anomalies which would disrupt the processes. So the stream of data must be constantly evaluated to ensure that all conditions are met for a smooth execution of the processes.

These scenarios are chosen to show that an arbitrary use-case like car parking or a playground scenario requires certain capabilities from the CPS which in this case is the Smart Home Environment. It is necessary to evaluate those capabilities to ensure that the execution of tasks by the CPS is possible. As the project envisions to develop a formal language which enables such evaluations, experiments in a laboratory should indicate the practicability of the approach. Fig. 3 shows the schematic of a smart home environment incorporating sensing devices.

5.4 Unique Contribution

The unique contribution of this research project is to use formal mathematical tools to define a modeling language for Cyber-physical systems which enables to evaluate the functionalities of the CPSs in specific environments. Category theory offers a broad array of useful abstractions to study and model composable systems. In this project those mathematical tools are applied to study, model and evaluate CPSs.

In a first step, a literature review was conducted to address the research questions RQ1 and RQ2. Following issues were identified as crucial to evaluate the functionality of CPSs:

- Topology of the CPS: Information related to the number of components, their connections and relations enable to define a high level network structure of the CPS.
- Reliability of Sensors and Actuators: CPSs rely on various sensors to sense the physical world and use actuators to perform actions. The challenges are the possibilities of malfunctions due to various reasons but also issues related to the correctness of the sensed data and execution inaccuracies.
- Heterogeneity of Communication: Components of CPSs can communicate using more than one communication channel but also using various communication protocols. Pair-wise communication between components can be directed or undirected. All these parameters have an effect on the used network model to describe the system.
- Dependency of applications: Parts of a CPS can be dependent on other parts but parts can also be independent and autonomous. A good understanding the hierarchy and relations of the parts of a CPS is obligatory.
- Safety and Security: Privacy and security related issues must be addressed and are dependent on the aforementioned issues.

As explained in section 5.1, using graphs is not always sufficient to grasp the complexity of the network as it does not include the notion of composition and equivalence of different paths in a network. Categories allow the use of diagrams to reason about the path equivalences in a network.

In the upcoming second step a modeling language for CPSs based on category theory will be developed given the requirements mentioned in this paper. In a third step, the conceptual modeling method based on the developed modeling language will be implemented using the ADOxx metamodeling platform to produce a software prototype. Based on empirical experiments using the prototypical implementation the developed tool and the modeling language will be refined iteratively. Such a modeling language subsequently would then allow the integration of CPSs models with the conceptual models of business processes and enterprises.

Acknowledgement. This doctoral project is supervised by o. Univ.-Prof. Prof.h.c. Dr. Dimitris Karagiannis, head of the research group Knowledge Engineering and the Open Models Initiative Laboratory (OMiLAB) at the Faculty of Computer Science, University of Vienna.

References

1. Rajkumar, R., Lee, I., Sha, L., & Stankovic, J. Cyber-physical systems: The next computing revolution. In Proceedings of the 47th Design Automation Conference, DAC '10, pp. 731-736. ACM, Anaheim, California (2010).
2. Derler, P., Lee, E. A. and Sangiovanni Vincentelli, A., Modeling Cyber–Physical Systems. In Proceedings of the IEEE, vol. 100, no. 1, pp. 13-28. IEEE, (2012).
3. Esterle, L. & Grosu, R. Elektrotech. Inftech. 133: 299. (2016)
4. Sztipanovits, J. et al., Toward a Science of Cyber–Physical System Integration. In Proceedings of the IEEE, vol. 100, no. 1, pp. 29-44, IEEE, (2012).
5. Lee, E. A., Fundamental Limits of Cyber-Physical Systems Modeling. ACM Transactions on Cyber-Physical Systems, vol. 1, issue 1, art. 3. ACM, New York (2017)
6. Lee, E. A. The Past, Present and Future of Cyber-Physical Systems: A Focus on Models. Sensors, 15, 4837-4869. MDPI, (2015).
7. Henzinger, T.: The Theory of Hybrid Automata. In Proceedings of the 11th Annual IEE Symposium on Logic in Computer Science, pp. 278-292. IEEE, New Brunswick (1996).
8. Alur, R. et al., The algorithmic analysis of hybrid systems. Theoretical Computer Science, 138, 1, 3-34. (1995).
9. Alur, R., Dill, D. L., A Theory of Times Automata. Theoretical Computer Science, 126, pp. 183-235. (1994).
10. Lee, E. A., Cyber Physical Systems: Design Challenges. In 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC). IEEE, Orlando (2008).
11. Tan, Y. et al., Spatio-Temporal Event Model for Cyber-Physical Systems. In 29th IEEE International Conference on Distributed Computing Systems Workshops. IEEE, Montreal (2009).
12. Lin, J., Sedigh, S., Miller, A. Modeling Cyber-Physical Systems with Semantic Agents. In 34th Annual Computer Software and Applications Conference Workshops. IEEE, Seoul (2010).
13. Thalheim, B., Conceptual Model Notions - A Matter of Controversy: Conceptual Modelling and its Lacunas, Special Issue on Conceptual Modelling in Honour of Heinrich C. Mayr. EMISA, pp. 9–27, (2018).

14. Karagiannis, D., Mayr, C. H. & Mylopoulos, J., Domain-Specific Conceptual Modeling: Concepts, Methods and Tools, 1st ed. Springer Publishing Company, (2016).
15. Buchmann, R. A., Karagiannis, D. & Walch, M., How can diagrammatic conceptual modelling support knowledge management? In Proceedings of the 25th European Conference on Information Systems (ECIS 2017), pp. 1568-1583, Guimarães, Portugal, (2017)
16. Walch, M., Operating cyber-physical systems with microservices: the s*IoT conceptual modelling approach. In Proceedings of the 2018 7th International Congress on Advanced Applied Informatics (AAI 2018), p. 787-792, AAI, Japan (2018).
17. Walch, M., Karagiannis, D., How to connect design thinking and cyber-physical systems: the s*IoT conceptual modelling approach, Manuscript accepted for publishing, Hawaii International Conference on System Sciences (HICSS), (2019).
18. Hevner, A., Chatterjee, S.: Design research in information systems: Theory and Practice, 1st ed. Springer Science and Business Media, (2010).
19. The OMiLAB Community, "OMiLAB" http://www.omilab.org/psm/about., last accessed 2018/10/7
20. Götzinger, D., Miron, E.-T. et. al, OMiLAB: An Open Collaborative Environment for Modeling Method Engineering, pp. 55-76. Cham: Springer International Publishing, (2016).
21. Meseguer, J. et al., Petri Nets Are Monoids, Information and Computation 88, 105-155 (1990).
22. Leinster, T., Basic Category Theory, Cambridge Studies in Advanced Mathematics, Vol. 143, Cambridge University Press, Cambridge, (2014).
23. Adamek, J. et al, Abstract and Concrete Categories, http://katmat.math.uni-bremen.de/acc/acc.pdf, Open access file, last accessed 2018/10/7
24. Barr, M., Wells, C., Category Theory for Computing Science, Prentice-Hall, Inc. (1990).
25. Healy, M.J., Caudell, T. P., Ontologies and worlds in category theory: implications for neural systems, Axiomathes, 16:165–214. (2006).
26. Karagiannis, D., and Kühn, H., Metamodelling Platforms, pp. 182-182. Berlin, Heidelberg: Springer Berlin Heidelberg, (2002).
27. Karagiannis, D., "Agile modeling method engineering," in Proceedings of the 19th Panhellenic Conference on Informatics, PCI '15, pp. 5-10, ACM, New York (2015).
28. Karagiannis, D., Buchmann, R. A., Burzynski, P., Reimer, U. and Walch, M., Fundamental Conceptual Modeling Languages in OMiLAB, pp. 3-30. Cham: Springer International Publishing, (2016).
29. Efendioglu, N., Woitsch, R. & Utz, W., A toolbox supporting agile modelling method engineering: Adoxx.org modelling method conceptualization environment. In Proceedings of PoEM'16, pp. 317–325, Skövde (2016).
30. Saunders, M. L., Categories for the Working Mathematician, 2nd ed., Springer Science and Business Media, (1978).