Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 136 (2025) 49–54

www.elsevier.com/locate/procedia

35th CIRP Design 2025

Structuring Inconsistency Situations in Engineering of Cyber-Physical Systems – a Description Template Proposal

Thomas Alexander Völk [a,*], Razieh Dehghani[b], Monika Klippert [a], Felix Pfaff [a], Robert Stolpmann[a], Konstantin Nowak[a], Anne Koziolek[b], Albert Albers[a]

[a] IPEK – Institute of Product Engineering, Karlsruhe Institute of Technology (KIT), Kaiserstr. 10, 76131 Karlsruhe, Germany
[b] KASTEL - Institute of Information Security and Dependability, Karlsruhe Institute of Technology (KIT), Am Fasanengarten 5, 76131 Karlsruhe, Germany

* Corresponding author. Tel.: +49 721 608-47253. *E-mail address:* thomas.voelk@kit.edu

Abstract

In today's interdisciplinary world, mechatronic systems are evolving by taking advantage of software services. Cyber-Physical Systems (CPS) are an example of such a progressive development. Since practitioners from various domains are involved in developing CPS, it is important to maintain consistency between the variety of artifacts produced. To understand the avoidance, discovery, analysis, and handling mechanisms, this paper proposes a template oriented to practitioners' demands to collect data about inconsistency situations. By using this template, a data set with inconsistency situations based on industry practitioners' workshops is provided. Researchers and practitioners can use this data to better understand and manage inconsistencies.

Peer-review under responsibility of the scientific committee of the 35th CIRP Design 2025

Keywords: Cyber-Physical Systems; Inconsistency Situations, Inconsistency in CPS-Engineering, Description of Engineering Situations

1. Introduction

Cyber-Physical Systems (CPS) integrate computation and physical processes [1]. They extend mechatronic systems through the possibilities of the Internet of Things. [2] In this way, CPS are a key enabling technology that holds great innovation potential for future applications. [3]

The successful development of CPS depends on the collaboration of experts from many domains working on different views on the System-in-Development (SiD). Engineers from mechanical and electrical engineering as well as from computer science, all view the same elements in a CPS, but from the differing standpoints of their domains and separate tasks. Those stakeholders work on various models that are usually not interoperable [4]. It is a challenge inherent to CPS engineering, because (a) the technical and socio-technical systems interrelate dynamically, and so (b) degrees of systems complexity increase dramatically. Engineers regularly encounter inconsistent models as well as the representation of such models in views [5]. When an actual inconsistency occurs in the context of a specific engineering situation, we speak of an inconsistency situation. Inconsistency situations per se remain largely unresearched [5]. In consequence, engineers cannot properly handle the inconsistency situations encountered regularly in development and production.

A first step toward addressing the broader problem of inconsistency in CPS engineering (including the steps from design to the production of CPS) is to improve our understanding of the situations in which inconsistency arises [6]. Previous work has shown that the description of inconsistency in engineering situations is not trivial, neither for researchers nor for practitioners [5]. If the description of inconsistency situations is already difficult, it will be even more difficult to fully understand an inconsistency situation. Previous attempts at solving this issue have resulted in a first template as a support for practitioners. Still, this template is not sufficient enough to allow for a precise and reproducible description of

Peer-review under responsibility of the scientific committee of the 35th CIRP Design 2025
10.1016/j.procir.2025.08.011

inconsistency situations. Therefore, further support is needed so that engineering practitioners in CPS engineering are enabled to understand inconsistency situations. This understanding is the prerequisite for actually managing inconsistency situations.

In this paper, we use a template to aid practitioners of CPS engineering in improving their *understanding of* and *communication about* inconsistency situations. The goal of this paper is to provide practitioners with a way of describing inconsistency situations that is both structured and formalized. To that end, we ask the following research questions:

- RQ1: How can established situational description supports be enhanced to provide support for inconsistency situation description?
- RQ2: What challenges do practitioners have when working on the designed support for inconsistency situation description?
- RQ3: What benefits can industry practitioners get from analyzing inconsistency situations from their engineering experience with the designed support for inconsistency situation description?

The paper is organized as follows. In Section 2, we discuss the approaches already available for situational description to propose our template as an enhanced aid in the description of inconsistency situations (RQ1). In Section 3, we conduct a preliminary study of our template by testing it out with students of CPS engineering. Next, our tested template is further trialed in a workshop in the industry environment (RQ2 and RQ3). In Section 4, we conclude and also provide an outlook on future work.

2. Systematic Description of Inconsistency Situations

2.1. State of Research for Situational Description Approaches

CPS engineering in general, and the understanding of inconsistency in CPS engineering in specific, are both yet emerging research [7]. Consequently, inconsistency situations have yet to be described in clear and formalized ways.

Current approaches describe engineering processes and environments only on a general level. For example, the VDI 2221 Part 2 [8] proposes a set of 30 contextual factors to describe engineering situations. Clark and O'Connor [9] provide 48 so-called environmental factors that describe software engineering environments. These situational or environmental factors were introduced to tailor a general engineering process to a specific engineering environment. Thus, the state of research lacks the level of detail needed to describe specific engineering situations and is therefore insufficient to describe inconsistency situations. Those approaches, prove to be inadequate to the task of systematically describing an inconsistency situation.

However, there are other description approaches like Clement [10] or Gausemeier et al. [11]. They describe the modeled SiD and could potentially be used to explain the occurrence of inconsistency in them. However, the description overlooks the engineering situation itself entirely.

The only approach that describes inconsistency situations from a practitioner's view is proposed by Albers et al. [5]. This approach involves engineering practitioners using a structured template. The template consists of three main parts: (1) a description of the inconsistency situation, (2) effects of inconsistency on the project, and (3) possible solution strategies to restore consistency. Thus, the template enables the collection of inconsistency situations from practitioners. However, it still lacks a level of formalization for other factor-based situational description approaches.

In this paper, our proposal is to further develop the template proposed by Albers et al. [5] to enhance it with a more formalized, factor-based description of inconsistency situations.

2.2. Enhancement of established situational description approaches for inconsistency situation description

Albers et al. [9] recommend how their proposed approach could be enhanced:

- Clarify the meaning of *inconsistency*, because the concept is only vaguely defined in interdisciplinary engineering.
- Formalize the description of inconsistency situations, being careful to include the exact type of data a practitioner will need to find the aptest description of the situation.

The clusters of factors associated with aspects of the engineering environment provided by VDI 2221 Part 2 [8] and Clarke and O'Connor [9] are chosen as a good starting point.

To find suitable situational factors, we used both a top-down and a bottom-up approach. From the top down, we collected possible factors describing an inconsistency situation through a literature review of the domains of software engineering, mechanical engineering, and systems engineering (Section 2.1 above). We merged and clustered our list, and then we added further possible factors that were absent. In total, we found 20 factors, which we clustered into the following four clusters (Appendix A):

- *Environment* (6 factors): Factors that have no direct effect on the inconsistency, but describe the context in which it occurred
- *Setup* (7 factors): Factors that describe what caused the inconsistency and describe what engineering tools, people, and resources were involved.
- *Characteristics* (4 factors): Factors that describe specific features of the inconsistency. They are necessary for comparison.
- *Aftermath and Learnings* (3 factors): Factors that describe what happened after the inconsistency, how the inconsistency was resolved, and how the inconsistency could be avoided in the future.

The primary challenge with our 20 factors for the description of inconsistency situations is making the factors accessible and understandable to practitioners. Based on the findings of Albers et al. [5] we decided that by simply listing factors we would not be able to enable practitioners to clearly define the inconsistency situations encountered in their real-world contexts. To overcome this challenge, we used a description technique inspired by User Stories that provide a user-friendly template to collect stakeholders' requirements [12].

We used the structure of the User Story to reformulate our 20 descriptive factors as specific building blocks. The purpose of these building blocks is to aid practitioners in giving more specific answers to questions in our template. Essentially, the building blocks bring additional information to the engineering situation as well as the inconsistency setting. This information includes (a) a static context in which a descriptive factor is analyzed, (b) a variable factor that describes what exact metric, and (c) a type of answer that is expected. An example is shown in Table 1.

Table 1: Structure of the User Story inspired building blocks based on an example from one of the 20 resulting factors

Context	Variable Factor	Type
The inconsistency occurred during the development of	SYSTEM IN DEVELOPMENT	free

In our template, the building blocks are implemented in three ways refering to types of factors introduced:

1) As *guiding questions*: These questions form the backbone of our template because they help to collect the basic information about the inconsistency situation. These types of questions can be used for the factor types "free" and "selection", the main difference lies in the answering mode: Free-type factors are answered by providing information (An example is shown in Fig. 1.), selection-type factors are answered by deleting not necessary information from a complete set of data.
2) As *open illustration boxes*: These boxes provide space and several open-ended questions. The purpose is to allow free choice in how a practitioner visualizes the interrelationship between different parts of the inconsistency situation. An example is shown in Fig. 3. The can be used for the factor type "free".
3) As *scales*: The scales help to handle questions where the answer cannot be discrete but must occur along a continuum. An example is shown in Fig. 4. They can be used for the factor type "continuous"

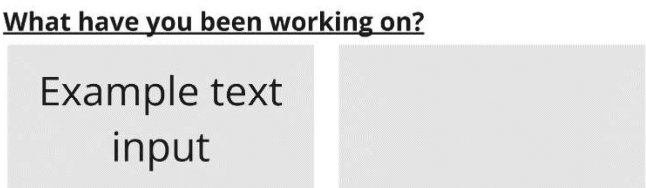

Fig. 1. Implementation of guiding questions of type "free" in a Miro-based template [13]

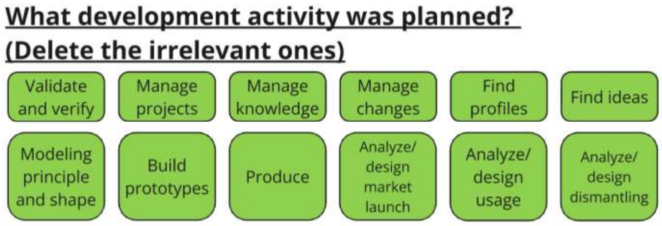

Fig. 2. Implementation of guiding questions of type "selection" in a Miro-based template [13]

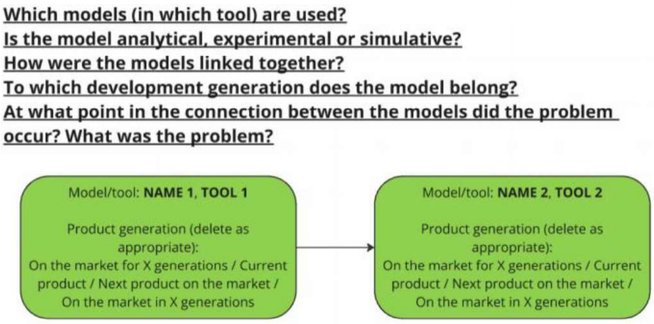

Fig. 3. Implementation of open illustration boxes in a Miro-based template [13]

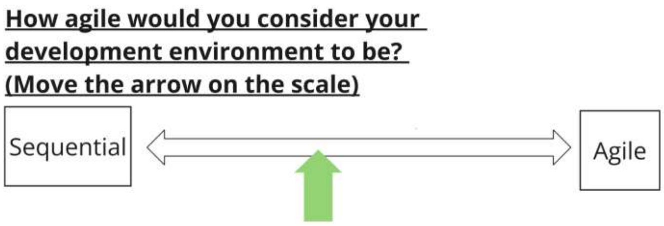

Fig. 4. Implementation of scales in a Miro-based template [13]

The resulting elements of our template are shown in Appendix A at the end of this paper. The implementation of our template in the collaboration tool Miro [13] is described at the end of Section 3.

3. Application of the Re-Designed Inconsistency Situation Template

3.1. Pre-study for testing the created template

As the template changed a lot from the first generation [5], we need to re-evaluate the usability. We tested the template in a workshop with six practitioners from a project during a part-time master's program. The eight students involved already have industrial business and engineering practice with different domain backgrounds from mechanical, electrical, and software engineering. Their task was to develop a CPS as a project during two semesters: The 6th product generation of an autonomous robot whose main use case is to deliver cold beverages from the student beverages sale to the course's main lecture room. The workshop was conducted after the final presentation of their project.

The workshop was oriented on the described procedure by Albers et al. [5]. We started the workshop with an introduction to the topic, a real practitioner example of an inconsistency in a brake system of a formula student racecar, where the reconfiguration of a variable in the source code led to a change in the braking principle and therefore to the failure of the system. This explanation is related to the original workshop from Albers et al. [5]. Besides that, we introduced them to the tasks they were asked to perform. We also stuck to the Think-Pair-Share structure [14] that was used in the initial workshop. After the introduction, the participants had 25 minutes to fill out one or more templates followed by an exchange on the inconsistency situations with a partner and a discussion of their findings in the plenum.

During the workshop, we encountered challenges that led to a change in the mode of the workshop: The time it took the participants to complete one template varied a lot. In general, 25 minutes was not enough to fill out more than one template. They therefore asked for more time, so the pair phase had to be skipped to ensure a discussion of the inconsistency situation. This did not affect the quality of knowledge gained from the workshop, so we decided to skip this phase for future iterations of the workshop to lower the time needed. In the share phase, the time for each explanation of the situations collected was the main factor that made the workshop exhausting in the last 15 minutes. We suggest cutting down the size of the workshop group to a maximum of 4 people and excluding the pair phase of the workshop.

The workshop showed that in general, the template was usable to describe (and later analyze) inconsistency situations from practice without the need for much post-processing (see Fig. 5, the figure is not intended to be readable, but rather should give an impression of the way the participants used the template). The students valued the learnings they got from participating in the workshop. Especially students from electrical and software engineering had no problem describing experienced inconsistencies between their views. Mechanical engineering students were more focused on communication inconsistencies and tool version inconsistencies. To better guide them towards inconsistency situations between views of the system, they wished for a better-introducing example. Also, the workshop was held in person and with physical printed templates. The participants wished for a digital version of the template rather than printed templates.

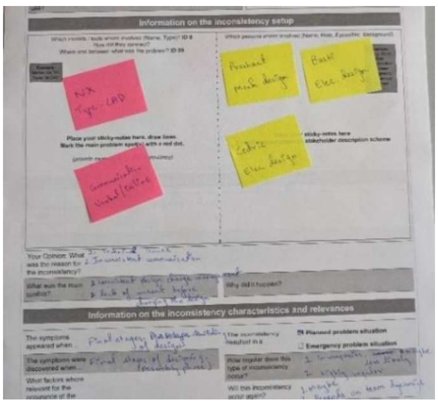

Fig. 5. Example of one of the participant's results from the pre-study in a printed-out version of the proposed template

3.2. Practitioner study applying the adjusted template and workshop concept

Overall, the reception to the proposed support was good and few potentials for improvement were identified. Most of the changes were found in the overall workshop design and template implementation rather than in the template structure.

The resulting iteration of the overall support was used in industry practitioner workshops. To examine whether the re-designed inconsistency situation description approach meets the needs of practitioners in companies, we conducted initial industry participants workshops with six experts (see Table 1).

Table 2: Overview of workshop participants working with the template

Workshop participant	Working field/ Domain	Work experience
Participant 1	Mechanical Engineering	5 years
Participant 2	Mechanical Engineering	7 years
Participant 3	Mechanical Engineering	Over 10 years
Participant 4	Software Engineering	12 years
Participant 5	Software Engineering	10 years
Participant 6	Mechanical Engineering	Over 20 years

The results give a first qualitative value to determine if our proposed template is operatable in company settings.

The workshops were conducted online in German and took between 60 (for single-partner interviews) to 120 minutes (for two or more participants). The workshops featured a presentation to introduce the subject matter and workshop tasks. Miro [13] was used as a collaborative whiteboard tool. The re-designed template described in section 2 was provided to the participants on Miro to standardize the process and maintain consistency in the results. The workshop methodology was intentionally structured to enhance participant engagement, achieve the desired outcomes, and minimize rework by directly documenting results (see section 6 “Discussion” from [5]).

3.2.1. Results regarding the template and workshop concept

First, the main outcomes and findings of the workshop regarding the template and methodology are briefly summarized: The workshop design as well as the re-designed template supported the participants to better understand and describe inconsistency situations. Before and after the workshop, the participants were asked to answer the following questions with the help of scales introduced in section 2.2 (see Fig. 4):

- *Q1:* How clear are the inconsistency situations in your development practice to you?
- *Q2:* Please assess yourself: How well are you able to describe inconsistency situations [before/after] the workshop?

The comparison of the participants' assessment before and after the workshop is shown in Fig. 6 and Fig. 7. Participants' assessment of Q2 (Ability to describe the After the workshop was conducted with numbers in place of the arrows used in the individual workshops (to increase the readability):

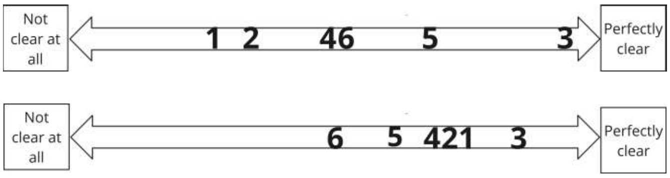

Fig. 6. Participants' assessment of Q1 (Clarity of the inconsistency situation) before (top) and after (bottom) the workshop was conducted

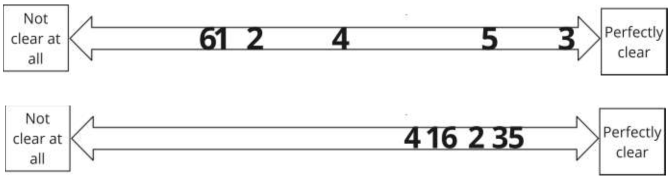

Fig. 7. Participants' assessment of Q2 (Ability to describe the inconsistency situation) before (top) and after (bottom) the workshop was conducted

In the participants' evaluation, we can see that the understanding of the inconsistency situation was increased by most of the participants, especially from mechanical engineering. Also, most of the participants increased their ability to describe inconsistent situations.

All except for one of the participants, who stated, that their understanding of the inconsistency situations was lowered, stated, that the workshop in general was useful for them. That leads us to the interpretation, that the self-assessment before the workshop was too self-confident.

In general, we found the template to be useful for practitioners who are inexperienced in describing inconsistency situations. In our limited sample, this was mostly the case for participants with a mechanical engineering background. To interpret this meta-data, a bigger sample is needed. Nevertheless, also experienced practitioners mentioned the need for such descriptions to communicate inconsistencies with inexperienced co-workers.

Also, the participants provided suggestions for improvement: They mainly included annotations for formatting and the explicit visualization chosen in Miro, but also other points for improvement:

- The wording of some of the building blocks is too close to academia. We address this by putting in additional explanations to the questions.
- The introducing example was too short, the practitioners expressed the need for a more complex situation described. We address this by taking more industry-relevant examples, i.e. the situations described in other workshops.

We conclude that our description support and template support practitioners to effectively communicate on CPS engineering and occurring inconsistency situations. In addition, we achieved successful results: three inconsistency situations have been described using the proposed template.

3.2.2. Results regarding inconsistency situations

As the focus of this paper is not directly on the inconsistency situations described in the workshops but rather on the applicability of the proposed template, we will not get into detail on the concrete findings of those workshops but want to share a brief overview.

The resulting inconsistency situation descriptions are openly accessible on Zenodo with the following DOI: 10.5281/zenodo.13968934 [15]

4. Discussion and Outlook

This paper proposes a way to enable practitioners to describe and analyze inconsistency situations in the engineering of CPS. We propose a template describing inconsistency through the related environment, setup, characteristics, aftermath, and learnings. The template was used in initial workshops in different practical settings. The application of the template resulted in a better understanding and communication of inconsistency situations and provided a first indication of benefit for practitioners. Challenges in implementing the template in workshop settings were shown and addressed. We conclude that the given template is a useful tool for practitioners from different domains to systematically address inconsistency situations in CPS engineering from their experience.

As we only have a very limited number of participants, we need more investigation on the broad application of the template in different professional engineering environments. This is of course a possible threat to validity that we are concerned about. Besides the benefit we could give to describe inconsistency situations, our understanding of inconsistency situations still needs further research. With the first samples given from student projects and the first industry cases, we got the first insights into engineering practice with inconsistency. Still, the given dataset is not sufficient enough to give in-depth insights into inconsistency situations occurring in the industry.

Acknowledgments

This research is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – SFB 1608 – 501798263. Thanks to our textician.

Appendix A. Descriptive Factors of an Inconsistency Situation

Table 3: Descriptive factors regarding the environment of an inconsistency situation

Context	Variable Factor	Type
The inconsistency occurred during the development of	SYSTEM IN DEVELOPMENT	free
The inconsistency occurred in the project during	DEVELOPMENT STATE	free
The inconsistency occurred in the project during	DEVELOPMENT ACTIVITY	selection
On which level did the inconsistency occur?	ASE-LEVEL	continuous
How agile would you describe your environment?	LEVEL OF AGILITY	continuous
Which process models do you use to describe your development?	PROCESS MODEL	free

Table 4: Descriptive factors regarding the setup of an inconsistency situation

Context	Variable Factor	Type
Which models (in which tool) are used?	INVOLVED MODELS	free
How did the models connect?	CONNECTIVITY OF MODELS	free
To which development generation does the model belong? Which variation share was found?	SYSTEM GENERATION	free
At what point of the connection between the models did the problem occur? What was the problem?	INCONSISTENCY	free
What stakeholders were involved?	INVOLVED STAKEHOLDERS	free
What background does the stakeholder have?	DOMAIN BACKGROUND	free
The inconsistency occurred because of	REASON	free

Table 5: Descriptive factors regarding the characteristics of an inconsistency Situation

Context	Variable Factor	Type
When was the inconsistency discovered?	INCONSISTENCY DISCOVERY	free
What symptoms were measurable?	SYMPTOMS	free
What kind of problem resulted from the inconsistency?	PROBLEM SITUATION,	selection/ free
How certain is the reoccurrence of the inconsistency? At what time?	INCONSISTENCY REOCCURANCE	free

Table 6: Descriptive factors regarding the consequences and learnings of an inconsistency situation

Context	Variable Factor	Type
To resolve the inconsistency	STEPS TO FIX INCONSISTENCY	free
Would you have been able to detect the inconsistency earlier? If yes, how?	ABILITY TO DISCOVER INCONSISTENCY	free
What support (through processes, methods, and/ or tools) would you have liked?	(UN-)AVAILABLE SUPPORT	free

References

[1] Lee EA. CPS Foundations. In: Sapatnekar S, editor. CPS Foundations; 2010. New York, NY, USA: ACM; 737–42.

[2] Graessler I, Hentze J. The new V-Model of VDI 2206 and its validation. at - Automatisierungstechnik 2020; 68(5): 312–24 [https://doi.org/10.1515/auto-2020-0015]

[3] acatech - Deutsche Akademie der Technikwissenschaften, editor. Cyber-Physical Systems: Innovationsmotor für Mobilität, Gesundheit, Energie und Produktion. Berlin, Heidelberg: Springer; 2011.

[4] Feichtinger K, Meixner K, Rinker F, *et al.* Industry Voices on Software Engineering Challenges in Cyber-Physical Production Systems Engineering. In: Industry Voices on Software Engineering Challenges in Cyber-Physical Production Systems Engineering; 2022. IEEE; 1–8.

[5] Albers A, Koziolek A, Völk TA, *et al.* Identification of Inconsistencies in Agile CPS Engineering with Formula Student. In: Bitran I, Conn S, Mitsis A, Ritala P, Torkkeli M, Trabelsi M, editors. Identification of Inconsistencies in Agile CPS Engineering with Formula Student; 2024; 1–15.

[6] Albers A, Reiß N, Bursac N, Breitschuh J. 15 Years of SPALTEN Problem Solving Methodology in Product Development. In: Boks C, editor. Proceedings of NordDesign 2016: August 10-12, 2016, Trondheim, Norway. Bristol, United Kingdom: The Design Society 2016.

[7] Giaimo F, Andrade H, Berger C. Continuous experimentation and the cyber–physical systems challenge: An overview of the literature and the industrial perspective. Journal of Systems and Software 2020; 170: 110781.

[8] Verein Deutscher Ingenieure e. V. Entwicklung technischer Produkte und Systeme: Gestaltung individueller Produktentwicklungsprozesse. Berlin: Beuth Verlag GmbH; 2019 11.2019.

[9] Clarke P, O'Connor R. The situational factors that affect the software development process: Towards a comprehensive reference framework. Information and software technology, 54(5), 2012 2012: 433–47.

[10] Clement SJ, McKee DW, Romano R, Xu J, Lopez JM, Battersby D. The Internet of Simulation: Enabling agile model based systems engineering for cyber-physical systems. In: The Internet of Simulation: Enabling agile model based systems engineering for cyber-physical systems; 2017. IEEE; 1–6.

[11] Gausemeier J, Schäfer W, Greenyer J, Kahl S, Pook S, Rieke J. Management of cross-domain model consistency during the development of advanced mechatronic systems. In: Design has never been this cool: ICED 09, the 17th International Conference on Engineering Design; 24 - 27 August 2009, Stanford University, Stanford, Califonia, USA; proceedings volume. Glasgow: Design Society 2009.

[12] Dalpiaz F, Brinkkemper S. Agile Requirements Engineering with User Stories. In: Agile Requirements Engineering with User Stories; 2018. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers; 506–7.

[13] About Miro | Meet the team | Our mission; 2024 [cited 2024 September 3] Available from: URL: https://miro.com/about/.

[14] Macke G, Hanke U, Viehmann-Schweizer P, Raether W. Kompetenzorientierte Hochschuldidaktik: Lehren-vortragen-prüfen-beraten 2016.

[15] Albers A, Völk TA, Pfaff F, Nowak K. Inconsistency Situations in Engineering of Cyber-Physical Systems. Zenodo; 2024.