Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 142 (2026) 770–775

www.elsevier.com/locate/procedia

36th CIRP Design Conference (CIRP Design 2026)

Chasing Symptoms, Missing Causes: Modelling Parallel Engineering Activities in CPS Development with the Advanced System Triple Approach

Albert Albers[a], Thomas Alexander Voelk[a,*], Jannis Piehl[a,b], Stefan Eric Schwarz[a], Sergio Di Giovannantonio[b], Patrick Grycz[a], Lars Gesmann[a], Tobias Düser[a]

[a]*IPEK – Institute of Product Engineering, Karlsruhe Institute of Technology (KIT), Kaiserstr. 10, Karlsruhe 76131, Germany*
[b]*CERN, Esplanade des Particules 1, 1211 Geneva 23, Switzerland*

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

---

Abstract

In cross-disciplinary engineering, the development of complex systems frequently exhibit unexpected behavior because of failures caused by inconsistency situations where interacting engineering artifacts reflect divergent systems states. Although the phenomenon of inconsistency is acknowledged in literature, there is limited empirical research on its emergence, propagation, and resolution within engineering design processes. This paper presents findings from a case study in the field of autonomous robotics, where we observed a cross-disciplinary development team during synthesis and analysis activities. Our investigation focused on identifying when and how inconsistencies arise and propagate across different engineering domains. Building on these insights, we propose a novel modeling approach that captures inconsistency propagation applying the advanced system triple: the system of objectives, the operation system, and the system of objects. By tracing the causes and propagation of inconsistencies across these three systems, we offer a more detailed approach to model cross-disciplinary collaboration that enhances the understanding of critical transition points in the design process. Our contribution provides a conceptual model and also practical insights for engineering practitioners and researchers.

Keywords: Inconsistency management, Cyber-physical systems, Product development process, Interdisciplinary engineering

---

1. Introduction

Product development is complex in advanced systems like cyber-physical Systems (CPS). CPS integrate computation and physical processes [1] and extend mechatronic systems through the possibilities of the Internet of Things [2], so their development requires collaboration between various disciplines.

Engineering projects mostly happen between so-called “knowledge silos” and can be developed only through cross-disciplinary approaches and tools enabling them [3, 4]. Such approaches require a certain consistency between the related models [5]. The preservation of consistency between different stakeholders in CPS engineering, in turn, needs distinct tools and methods [4]. While there are promising approaches to address the software tool challenges in consistency management, e.g. [3, 4, 6], there is still further research needed to address the challenges of suiting methods and processes to CPS engineering [3]. Systems Engineering promises to resolve such challenges and shows good results implementing it [7], but still faces problems in industrial practice, especially in broad and successful implementation in product development processes [8]. Such processes are already successful modelled in process models able to describe the activities of product development on different levels of granularity [9–13]. As such process models mostly are designed for the purpose of management of product development activities, they are not designed to be able to describe nor explain the dependency between (in-) consistency and engineering activities. In this work, we want to give an in-depth analysis of (in-)consistency

10.1016/j.procir.2026.05.344

and its propagation in a real-world engineering project, thereby motivating a unified understanding of inconsistency in cross-disciplinary engineering.

2. Related work

2.1. Engineering process models from an inconsistency point of view

Engineering of systems is reliant on collaboration of different domain experts that is mainly defined by iteration, co-evolution of problem and solution and the conflict of different objectives from different development stakeholders. [12] These characteristics are immanent threads for the consistency of a project. Yet, prominent standards in engineering processes only partially investigate the topic of consistency.

The VDI 2221 [10] as one of the main standards for product development processes mentions, that in simultaneous engineering the stability of work step results is big enough so the probability of change (and corresponding cost) has to be lower than the cost a possible change would result in. While this is hard to be estimated in a complex development environment, it is also dangerous: Especially in unknown engineering domains, the estimation of change criticality can be far more differing than from domain experts. If a change is needed due to emergence effects (e.g. [14]), no domain-specific evaluation would be possible. While the resulting model for a generic product development process is able to derive possible development processes, it is too abstract to emphasize consistency in such processes or only focusses on successful implementation in companies [15].

The VDI 2206 [9] in its current iteration emphasizes the cross-disciplinary reality of product development of CPS as parallel domain-specific development streams, but does not focus on development iterations, but rather on controllable check points. It is expected that developers collaborate on interface topics (e.g. system architecture, installation space for both electrical and mechanical components). Yet the VDI 2006 does not emphasize the operationalization of such tasks. Also, its late focus on validation (is the right system developed) in comparison to verification (is the system developed the right way – or – are the requirements met?) introduces the problem, that unforeseen emergence effects of the CPS are not manageable beforehand.

In recent years, agile practices are more and more common in the engineering of advanced mechatronic and CPS [16]. The advantage of such agile approaches is the iterative and fast development which enables a quick response to changes. Mostly, those approaches are reflected in framework like SCRUM or DevOps. While the original approach was mainly introduced for software development, hardware development adoptions are more and more common. However, it is still not suitable to react to different domain paradigms, resulting in problems with consistency in implemented systems [17].

All of the given process models have in common that they focus more on the sequence of different activities and the resulting transformation of working results, rather than on the effect of iteration on engineering knowledge generation itself. The Advanced System Triple Approach introduces such a fine-granular micro-process model designed to display the interplay of a system of objectives and a system of objects connected and transformed by an operation system through analysis and synthesis activities [13].

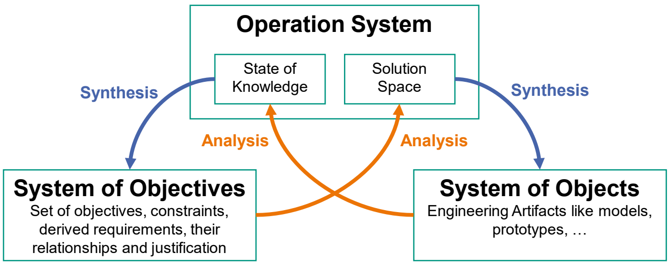

Fig. 1. Advanced System Triple Approach, own illustration based on [13]

This model emphasizes i) human aspects of individual thinking and acting as well as interactions between individuals based on documented objects; ii) knowledge aspects like the classification of knowledge and strategies of knowledge management; and iii) process aspects of co-evolution and iteration in engineering. While this process model emphasizes the underlying mental model of iterative engineering progress as co-evolution of objectives and objects, it fails to distinguish between different domain goals, individual activities, or objects that can be inconsistent with each other.

Although most process models highlight the importance of cross-disciplinary collaboration, they do not model the collaboration activities involved in practice: Either process models only focus on different engineering domains going through parallel engineering activities at the same time (e. g. VDI 2206 [9]), or process models focus on an integrated understanding of activities for the overall development of a new system generation as well as the corresponding validation and production system (e. g. VDI 2221, iPeM [11, 13]). However, neither approach from literature integrates multi-disciplinary collaboration from a practitioner's point of view.

2.2. Managing and Modelling (In-)consistency

Inconsistency is defined primarily by the viewpoints from formal mathematics and informatics. Inconsistency describes a state where two statements on a model or a set of models can't be fulfilled at the same time [18]. Therefore, consistency describes the opposing state. While this understanding is completely logical, it is often difficult to map this understanding onto actual engineering practice [19]. From a continuous systems engineering perspective, consistency means that different models that represent the same system of interest and are modelled from different domain views do not contradict each other [20]. Such inconsistencies often result from changes, which are the result of variation: In the model of SGE – system generation engineering, a new system generation is derived by applying variation operators on elements of a reference system [21]. By applying this model in an interdisciplinary setting, cross-generational consistency preservation becomes a problem. [22]

This state of consistency can be achieved by different approaches. Most approaches (e.g. Model-Based Systems Engineering) choose a single underlying model to ensure consistency. A domain-spanning model is therefore used to reflect domain-specific changes (e.g. [23]). Such approaches fail in two important steps: First, the management of an overarching system model quickly becomes unmanageable by a set of systems engineers. Second, not every semantic overlap of every domain-specific model to a generalized "artificial" single-underlying model can be guaranteed. This manifests itself in manual processes and a lack of automation, issues with tools, and limitations of current processes, communication gaps, cross-domain coordination problems, versioning and traceability issues, human factors increasing the impact of consistency issues and coordination issues in between different process practices reported in inconsistency situations. [24]

In practice, this is translated to a core issue: Single engineers have to ensure consistency while developing their design [19]. However, an engineer does not usually have support that enables them to assess the consistency in their design process in comparison to other sub-project teams in their engineering project.

Völk et al. proposed a framework to retrospectively model inconsistency situations in engineering environments, focusing on the development environment, setup, characteristics, and learnings [25]. While this model has been proven to support engineers in assessing past situations [14], a prospective use was not shown yet.

3. Research Aim, Research Questions and Research Methodology

The aim of this work is to present an approach to model cross-disciplinary collaboration in engineering environments both from a theoretical and practical perspective. To achieve this goal, we want to answer the following questions:

1. How can practitioners analyse inconsistency during ongoing development activities?
2. How are inconsistencies propagated between design activities and engineering domains in cross-disciplinary design?
3. How can the propagation of inconsistency between design activities and engineering domains be modelled based on the Advanced System Triple Approach?

To answer the research questions, we take the introduced inconsistency situation description template [25] as a base for prospective application during an ongoing engineering project. Our main hypothesis is, that the given template is not only applicable for retrospective, but also for prospective documentation.

4. Assessing inconsistency in ongoing engineering projects based on a case study at CERN

The investigated environment is set up at the European Organization for Nuclear Research (CERN) in the robotic maintenance development for its large hadron collider. Due to the hazardous environment in the accelerator tunnel system, maintenance robots are an essential tool for properly operating the research facility. One robotic platform is designed to set fasten Beam Loss Monitors (BLM) to the tunnel floor. BLMs monitor the level of ionizing radiation emitted by the accelerator beam preventing quenching of superconducting magnets and therefore potential catastrophic failure due to overheating. To properly set and fasten BLMs and their support poles, a new tool with a lifting mechanism compatible with the existing robots should be developed.

The robot fleet is remotely operated via Wi-Fi and several cameras on the robot, making the overall fleet a cyber-physical system. To enable the remote control of the newly developed tool, it has to provide accelerometers, endoscopic cameras, and torque sensors to achieve precise operation. Potential future use cases include implemented virtual reality and a complete digital twin for improved operation, as well as autonomous operation.

During the development of the tool, a core team of a mechanical engineer and a robotics engineer collaborated closely: The robotic engineer was responsible for designing electronics (i.e. printed circuit boards, PCB) as well as for communication between PCBs and the developed tool, the robot, and the human operator. The mechanical engineer was responsible for the construction and dimensioning of the mechanical parts, as well as the prototype production and implementation. The authors of this work include both members of the engineering team and external observers documenting the engineering progress and inconsistency occurrence.

Originally, the inconsistencies should be documented in a weekly retrospective: the templates described above were to be used to record the inconsistencies that arose during the week. However, this quickly proved to be quite difficult, as often only progress in development could be reported. An inconsistency, even if rooted in one of those proceeded steps or artefacts, became only imminent if a conflict could be detected. This leads to the first important finding of this work: Inconsistency situations can only be identified retrospectively as they are directly linked to previously not noticed problem. Only if this conflict can be identified, the source of an inconsistency can be resolved.

Nevertheless, the established procedure of regular retrospectives turned out to be useful, especially at the end of the project. Through the rigorous documentation of possible interconnections of development artifacts, the sources of conflict could be quickly identified and therefore resolved, resulting in the following inconsistency situation descriptions.

Due to space limitations, we give only a summary of the inconsistency situations.

4.1. Inconsistency situation 1: Low degree of maturity of the screwdriver system

The maturity of the general design of the screwdriver unit was still low, so constraints and boundary conditions as parts of the system of objectives changed regularly based on different stakeholders' needs. As a result of this constant change, the design of the system's architecture and technical implementation needed to be adapted to the new conditions. Changing involved models from week to week took a lot of developing time and led to inconsistent design states between the developers involved due to constant changes in the system of objectives.

4.2. Inconsistency situation 2: Conflict of decision responsibility in partial system integration

During the design of the housing a conflict in the lead decision making was found. The cylindrical housing should both be stiff enough to fit the use-case but also house electrical components that meet the requirement of an interface change. Therefore, a PCB should be implemented to split up the current for different use cases (electrical motor, IMU and endoscopic camera). The mechanical design engineer and the robotics engineer worked simultaneously on their respective design (housing and PCB). During meetings, both engineers had different designs in mind and the PCB could not be implemented into the housing at first. After it was decided on which design was leading in decision-making, the PCB could be adjusted and fit in the housing.

4.3. Inconsistency Situation 3: Late decision for principle solution

During the design of the lifting mechanism for the BLM-Pole and BLM, several mechanisms were proposed. While one principle solution was based on previous work and therefore was associated with less new development, it did not match the required compliance with the BLM interfaces by design what was not traceable. The alternative solution would require a bigger share of new development but could be adjusted to the introduced interfaces. Due to the unclear boundary conditions, both versions were developed simultaneously with different scopes that were not optimized towards the boundary conditions.

4.4. Inconsistency Situation 4: Late identification of reference system elements

In the beginning of the design of the robot’s end effector, no commitment towards clear systems requirements, architecture or sourcing was made. The choice of whether a commercial tool exchanger or an in-house development connection would be sufficient remained open. During the investigation of alternative solutions, further requirements were adapted, like modularity and universal usage beyond the projects scope. This change in requirements made especially developed single-application concepts redundant, as external sourcing was decided, setting the constraints for further tool interfaces. The mechanism itself was still not decided on.

Later in the development, it was decided that a remote, mechanically actuated tool exchanger would be used as no additional powerlines (electrical or pneumatic) need to be added. A tool exchanger that fits these design constraints was already used in another project in the same section. The development of another solution was therefore redundant from the beginning, as a database of tool exchangers inside the robotics section would have made the research on tool exchangers easier.

4.5. Inconsistency situation 5: Validation environment setup mistakes (FEA simulations)

The inconsistency occurred during the FEA simulations conducted to check if the housing and spacers used would withstand the torque applied by the screwdriver. During the simulation, a singularity (stress diverged to infinity as mesh size decreased) occurred regularly. The reason that this singularity occurred was due to an edge on the screw head of the thread. At a perfect 90-degree angle, the stress diverged to infinity. To resolve this problem, a 0.2mm fillet was added, and the solution converged. The inconsistency stemmed from insufficient knowledge on the transferability of design models into FEA simulations and unfamiliarity with the modelling of a suitable and credible validation environment.

4.6. Inconsistency situation 6: Concurrent data representation and redundancy in agile data management

The inconsistency occurred during the project documentation and realization. Originally, the data was stored on the local computer drive of one of the engineers involved. Because of this decentralized data access, the project's supervisor was not able to access the current state of the project, relying on outdated or accessible data if no direct communication between the engineers involved was possible. Consequently, the data was transferred to the CERN server, so the data was accessible to the supervisor.

At a later stage of the project, a file needed to be accessed that included an intervention manual for the pole setting in the CERN tunnel. As the models of the robotic arm and tool were updated during the development process, the intervention instruction itself needed to be updated. Due to the data transfer and missing linkage, the instruction document was lost. A copy was only found in a decentralized saved message that was sent with the document to the project’s supervisor.

4.7. Inconsistency situation 7: Missing documentation references and conventions

The inconsistency occurred during the creation of technical drawings. During the design process, the quality and usability of derived drawings from the designed elements varied. While a global GD&T (geometric dimensioning and tolerancing) standard is present, the creation of the corresponding models was difficult, as the relevant parameters were not clear to the involved stakeholders, resulting in no increase in quality. A stakeholder intervention resulting in engineering domain-specific guidelines enabled the resolution of inconsistent document states.

5. Modelling inconsistency propagation in the Advanced System triple model for cross-disciplinary engineering

Based on the insights of the engineering case study given in Section 4, we want to give a descriptive model to investigate inconsistency propagation between subsequent activities in engineering design. As introduced in Section 4, a given process should sufficiently cover both joint and individual design processes of different domains and should be able to explain both inconsistency sources and symptoms.

We chose the Advanced System Triple Approach as it is providing an integrated view to model domain-independent engineering activities combined with enough granularity to address the detailed situations introduced in Section 4. Therefore, the core principles introduced by Albers et al. [13] are preserved: Engineering progress is shown as the co-evolutionary refinement of a system of objectives and a system of objects through analysis and synthesis steps through the operation system (see Fig. 1). While this is per se an infinite process, for a detailed analysis a starting point has to be set. As regularly an engineering process is started with a goal that should be satisfied, the system of objectives is set as our default starting point.

We claim that each participating domain – in the sense of (sub-)discipline – or engineering team has its own sub-system of objects and its own operation system, which form a subset of the overarching system but are created by the domain-specific epistemic foundations, interpretations, and standards. However, this split does not apply because the system of objectives itself should be an overarching, unified understanding and is already constructed by the set of all entities in a product engineering organization. That results into two major differences between the original Advances System Triple Approach and our variation illustrated in Fig. 2:

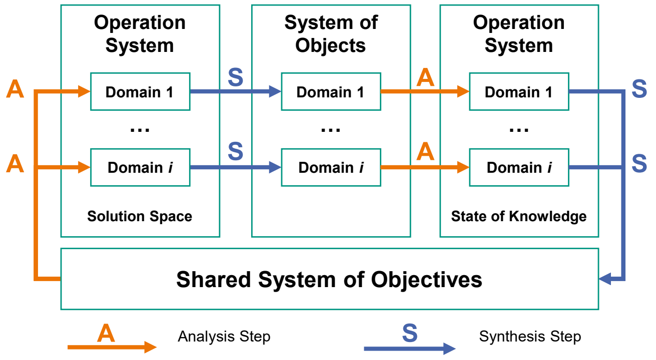

Fig. 2. Domain-specific adjustment of a complete iteration in the Advanced System Triple Approach, own illustration

1. The analysis step from the system of objectives to each of the subsystems of the operation system splits into *i* different streams for *i* different domains.
2. All streams reunite into a continuous system of objectives.

This enables the discussion of three different possible inconsistencies in cross-disciplinary engineering processes:

1. Inconsistency through contradiction in engineering artifacts
2. Inconsistency through contradiction in domain-specific partial operation systems
3. Inconsistency through contradiction in analysis and synthesis activities

Each of the given inconsistency situations described in Section 4 was analyzed regarding the occurrence of an inconsistency source (root cause located in a given element of Fig. 2), inconsistency symptoms (measurable conflicts between artefacts not discovered yet), and inconsistency detection (discovered conflicts between artefacts). The results are given in Table 1.

Table 1. Inconsistency occurrence in inconsistency situations modelled with the adjusted Advanced Systems Triple

Model element	Inconsistency source	Inconsistency symptom	Inconsistency detection
System of objectives	2	0	0
System of Objects	2	7	1
Operation system – (solution space)	6	1	0
Operation system – (state of knowledge)	0	1	7

6. Discussion of theoretical implications and practical indications of the proposed model

Based on the adjustments given, different conclusions for practice can be derived from literature and the given case study:

Inconsistency symptoms can only be found in the system of objects, because it contains the only explicit part of engineering in the shape of artifacts (see also [14]). These symptoms are indicators of inconsistency and are usually connected to an engineering problem. Resolving this inconsistency requires the identification of the root cause of the symptom.

On the other hand, *inconsistency reasons* are mostly found in "translation" mistakes between analysis and synthesis in the operation system. Epistemic barriers are major hindrances in cross-disciplinary collaboration [26]. Such barriers can affect both the analysis and synthesis steps and are deeply rooted in engineering cognition. Also, MBSE can fall victim to this, as modelling activities involve a huge epistemic knowledge and are mostly not accessible to engineering practitioners. This explains why MBSE implementation has proven to be of great use in industry but regularly shows flaws in implementations on a bigger scale. [8] Therefore, consistency management must address semantic overlaps between different sub-systems of objects both in engineering tools and methods, involving a translational effort by a unified understanding, e.g. through the application of SE methods. [8, 24]

Improvement of consistency can be addressed through different measures, but covering inconsistency only where it is discovered is solely treating symptoms. Therefore, conscious inconsistency management should change the way inconsistency is treated: Engineering process managers should stop chasing symptoms that will occur all over again; they should *search* for symptoms and *chase* for root-causes identified in objects, operations, and their transformation activities. Our task as design support researchers must be the synthesis of suitable methods and tools.

7. Conclusion and Outlook

Our introduction of a cross-domain centric adjustment to the Advanced System Triple enables us to discuss root-causes of inconsistency in the development of complex systems such as CPS. Modelling such root causes is important, because it enables future research to investigate cause-and-effect chains in cross-domain collaboration. With this understanding, the design of new approaches for consistency management and (semi-automated) preservation can be achieved. Yet, we can directly explain inconsistency only at a detailed micro-activity level. To further address consistency management on a development project scale, it will be necessary to integrate small cause-and-effect chains into a unified understanding of consistency propagation in engineering environments.

Furthermore, the challenges presented here are not only relevant for product development projects addressing a single product generation but are also related to cross-generational product development. Changes made in an early development stage of a single product generation can directly affect artifacts of a more mature product generation. That can lead to problems in integration and market deployment. Also, the artifacts generated in the product development process must also be kept consistent with the development of a validation system and the corresponding production system of a product (and its generations). Such an integrated understanding of (in-) consistency is still missing and proves the requirement for further work to enable integrated consistency management in engineering of advanced systems such as CPS.

Acknowledgements

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – SFB 1608 – 501798263. This paper has been edited by our textician, Daniel Shea.

References

[1] Lee, E.A., 2010. CPS Foundations, in *Proceedings of the 47th Design Automation Conference*, ACM, New York, NY, USA, p. 737.

[2] Graessler, I., Hentze, J., 2020. The new V-Model of VDI 2206 and its validation *68*, p. 312.

[3] Cicchetti, A., 2016. Consistency and Uncertainty in the Development of Cyber-Physical Systems, in *MPM4CPS: Multi-paradigm modelling for cyber-physical systems*, p. 56.

[4] Reussner, R., Schaefer, I., Beckert, B., Koziolek, A. *et al.*, 2023. Consistency in the View-Based Development of Cyber-Physical Systems (Convide), in *2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C)*, IEEE, p. 83.

[5] Qamar, A., Wikander, J., During, C., 2012. Overcoming Current Mechatronic Design Challenges: A Discussion, in *Mechatronic 2012*.

[6] Klare, H., Kramer, M.E., Langhammer, M., Werle, D. *et al.*, 2021. Enabling consistency in view-based system development — The Vitruvius approach *171*, p. 110815.

[7] Dumitrescu, R., Albers, A., Riedel, O., Stark, R., Gausemeier, J., 2021. *Engineering in Germany - The status quo in business and science, a contribution to Advanced Systems Engineering*, Paderborn.

[8] Albers, A., Koziolek, A., Voelk, T.A., Grote, E.-M. *et al.*, 2025, in press. How to "Measure" a Mirage: Assessing Inconsistency in Mechatronic and Cyber-Physical Development Projects, in *2025 IEEE International Symposium on Systems Engineering (ISSE)*, IEEE, p. 1.

[9] Verein Deutscher Ingenieure e. V. Development of mechatronic and cyber-physical systems, Berlin. Beuth Verlag GmbH *03.100.40, 31.220.01, 39.020*, 2021(VDI/VDE 2206).

[10] Verein Deutscher Ingenieure e. V. Design of technical products and systems: Model of product design, Berlin. Beuth Verlag GmbH *03.100.40*, 2019(VDI 2221 Part 1).

[11] Albers, A., Reiss, N., Bursac, N., Richter, T., 2016. IPeM-Integrated Product Engineering Model in Context of Product Generation Engineering.

[12] Pahl, G., Beitz, W., Feldhusen, J., Grote, K.-H., 2007. Product Development Process, in *Engineering Design*, Springer London, London, p. 125.

[13] Albers, A., Ebel, B., Lohmeyer, Q., 2012. Systems of objectives in complex product development.

[14] Albers, A., Völk, T.A., Pfaff, F., Nowak, K., 2025. *Inconsistency Situations in Engineering of Cyber-Physical Systems*, Zenodo.

[15] Verein Deutscher Ingenieure e. V. Design of technical products and systems: Configuration of individual product design processes, Berlin. Beuth Verlag GmbH *03.100.40*, 2019(VDI 2221 Blatt 2).

[16] Schmidt, T.S., Weiss, S., Paetzold, K., 2018. *Agile Development of Physical Products: An Empirical Study about Motivations, Potentials and Applicability*. Universitätsbibliothek der Universität der Bundeswehr München, Neubiberg.

[17] Weiss, S., Paetzold-Byhain, K., Michalides, M., Pendzik, M., Scharold, F., Stoiber, L., 2023. *Agile Entwicklung physischer Produkte 2023*. Technische Universität Dresden.

[18] Spanoudakis, G., Zisman, A., 2001. Inconsistency Management in Software Engineering: Survey and Open Research Issues, in *Handbook of Software Engineering and Knowledge Engineering*, World Scientific, p. 329.

[19] Albers, A., Koziolek, A., Völk, T.A., Klippert, M. *et al.*, 2024. Identification of Inconsistencies in Agile CPS Engineering with Formula Student, in *Proceedings of The XXXV ISPIM Innovation Conference*, p. 1.

[20] Albers, A., Lohmeyer, Q., 2012. Advanced Systems Engineering - Towards a Model-Based and Human-Centered Methodology, in *9th International Symposium on Tools and Methods of Competitive Engineering TMCE*, p. 407.

[21] Albers, A., Rapp, S., 2022. Model of SGE: System Generation Engineering as Basis for Structured Planning and Management of Development, in *Design Methodology for Future Products*, Springer International Publishing, Cham, p. 27.

[22] Albers, A., Schaefer, I., Gesmann, L., Ochs, P. *et al.*, 2024. Consistency in Cross-Generational Engineering of Cyber-Physical Systems, in *DS 133: Proceedings of the 35th Symposium Design for X (DFX2024)*, The Design Society, p. 85.

[23] Gausemeier, J., Schäfer, W., Greenyer, J., Kahl, S. *et al.*, 2009. Management of Cross-Domain Model Consistency during the Development of Advanced Mechatronic Systems, in *DS 58-6: Proceedings of ICED 09, the 17th International Conference on Engineering Design, Vol. 6, Design Methods and Tools (pt. 2), Palo Alto, CA, USA, 24.-27.08. 2009*, p. 1.

[24] Voelk, T.A., Nowak, K., Pfaff, F., Dehghani, R. *et al.*, 2025, in press. Consistently Inconsistent: AI-Analyzed Patterns in Inconsistency-Situations of Cyber-Physical Systems Development, in *2025 IEEE International Symposium on Systems Engineering (ISSE)*, IEEE.

[25] Völk, T.A., Dehghani, R., Klippert, M., Pfaff, F. *et al.*, 2025. Structuring Inconsistency Situations in Engineering of Cyber-Physical Systems - a Description Template Proposal *136*, p. 49.

[26] Voelk, T.A., Gesmann, L., Götze, S., Feichtinger, K. *et al.*, 2025. Misunderstand me correctly - comprehensibility in interdisciplinary collaboration *5*, p. 2441.