Ansatz zur domänenübergreifenden Rückverfolgbarkeit in Cyber-Physischen Systemen

Approach to cross-domain traceability in Cyber-Physical-Systems

Francesco Pio Urbano[1], Maximilian Beck[2], Katharina Bause[1], Arne Bischofberger[1], Eric Sax[2], Albert Albers[1]

[1]Karlsruher Institut für Technologie, Institut für Produktentwicklung, Karlsruhe
Francesco.urbano@kit.edu
[2]Karlsruher Institut für Technologie, Institut für Technik der Informationsverarbeitung, Karlsruhe,
Maximilian.beck@kit.edu

Abstract: Die domänenübergreifende Rückverfolgbarkeit stellt eine zentrale Herausforderung in der Entwicklung von Cyber-Physischen Systemen (CPS) dar, insbesondere angesichts der steigenden Vernetzung und dem Einsatz moderner Technologien. In diesem Beitrag wird ein Ansatz zur Integration mechanischer, elektrischer und softwarebasierter Modelle zur Unterstützung der Rückverfolgbarkeit dargestellt. Hierzu wird der Contact and Channel Approach (C&C²) genutzt, um semantische Verknüpfungen zwischen mechanischen und elektrischen Modellen herzustellen und Rückverfolgbarkeitsberichte zu generieren. Eine Fallstudie zeigt die erfolgreiche Anwendung auf ein elektromechanisches Bremssystem. Die Ergebnisse zeigen Potential des Ansatzes für die Verbesserung der Konsistenz, Qualität und Wartbarkeit von CPS. Herausforderungen liegen jedoch in der Automatisierung der Modellverknüpfung, der Skalierbarkeit für große Systeme und der Visualisierung der Berichte. Zukünftige Methodenentwicklungen zielen darauf ab, diese Einschränkungen durch KI-gestützte Ansätze und optimierte Schnittstellen zu überwinden, um eine effiziente und praxisnahe Implementierung zu ermöglichen.

Keywords:
Rückverfolgbarkeit, Cyber-Physische Systeme, C&C²-Ansatz, Domänenübergreifende Modellierung, Automatisierung

Abstract: Cross-domain traceability represents a key challenge in the development of cyber-physical systems (CPS), particularly given the increasing interconnections and use of modern technologies. This paper presents an approach for integrating mechanical, electrical and software-based models to support traceability. The Contact and Channel Approach (C&C²) establishes semantic connections between models and generates traceability reports. A case study shows the successful application to an electromechanical brake system. The results show the potential of the approach for improving the consistency, quality and maintainability of CPS However, challenges remain in automating model linking, scaling for large systems, and visualizing reports. Future method developments aim to address these limitations through AI-driven approaches and optimized interfaces to enable efficient and practical implementation.

Keywords:
Traceability, Cyber-Physical Systems, C&C² Approach, Cross-Domain Modeling, Automation

1 Introduction

Modern cyber-physical systems (CPS) such as vehicles present engineers with new challenges. With the integration of advanced technologies such as autonomous driving and car-to-x communication, the complexity of such systems is increasing significantly. This leads to increased demands on the subsystems involved, including the powertrain as well as chassis and body. Modern vehicles can have up to 150 interconnected electronic control units (ECUs), and an estimated 10,000 individual parts, including a large number of mechanical components (Vetter and Sax 2021; Friedrich and Müller 2024). One example of a subsystem, including all domains, is the braking system, which fulfills new requirements in the context of hybrid or electric vehicles, including recuperation. While conventional hydraulic brake systems do not meet these requirements, hybrid brake systems combine classic brake hydraulics and drivetrain parts, such as motors or transmissions (Venkatesh 2024). These are controlled by several ECUs and corresponding software. The interplay of mechanical, electrical, and software domains makes CPS, such as hybrid braking systems, particularly challenging. While the Model-based systems engineering (MBSE) paradigm aims at an integrated system representation, development tools remain largely domain-specific and isolated, limiting interoperability and traceability across domains (Neureiter and Binder 2022; Wu et al. 2024). This makes it challenging to trace requirements throughout the product development process and impairs maintainability and product quality.

To overcome these challenges, new methods and processes are required that enable the various domains to be linked seamlessly. Tools and methods are only used sustainably in practice if they contribute directly to the product or are prescribed by legal requirements. Above all, the automation of such methods is crucial to guarantee the traceability of requirements and ensure the quality of the developed products in the long term.

In this paper, we contribute to the goal of complete and automatic traceability by proposing methods for connecting mechanical and electrical models, providing a concrete domain model for this purpose and proposing a tool that can be used to analyze and edit the created models. In the end, we will apply the developed methodology in a simple case study, discuss the findings and results, and indicate open questions and future research opportunities.

2 State of the Art

This chapter presents the state of the art in cross-domain traceability in CPS. First, key challenges in model integration across mechanical, electrical and software-based domains are described. Methodological approaches that support traceability in model-based development processes are then systematized. The focus here is on interoperable standards, semantic model links and approaches to tool integration. Finally, a critical classification of existing research gaps is provided.

CPS such as electromechanical braking systems consist of highly interconnected mechanical, electrical and software-based components. These systems require the synchronization of data across multiple domains to ensure efficient functionality and safety (Li et al. 2024). However, development tools used for these domains are often domain-specific and based on proprietary data structures, limiting interoperability and preventing seamless traceability. For example, computer-aided design (CAD) models do not contain any semantic information about physical relationships (Chaparala et al. 2013), while E/E architecture models (e.g. in PREEvision) map functional and logical levels.

A central challenge is the lack of interoperability of these tools, which prevents consistent traceability across all domains (Wu et al. 2024; Gu et al. 2024). Moreover, the manual effort required to maintain and synchronize models, as well as inconsistent use of terminology across domains, hampers sustainable systems engineering. A concrete example of this complexity is the integration of mechanical components with embedded software and ECUs in hybrid braking systems, where each element is often developed in isolation.

MBSE pursues the goal of systematically mapping complex systems in models and developing them consistently over the entire product life cycle. Languages such as SysML are used for this purpose. Various standards have been developed in recent years to improve cross-domain model linking. The System Structure and Parameterization (SSP) standard enables the structured coupling of simulation models and thus creates a common description hierarchy for models of different tools (Blockwitz et al. 2012). In addition, the Functional Mock-up Interface (FMI) allows the co-simulation of heterogeneous models, whereby the focus of this standard is on behavioral simulation. A more comprehensive architecture-based approach is taken by the Unified Architecture Framework (UAF), which offers a uniform modeling structure to consistently link technical, operational and safety-relevant aspects (Abhaya 2021). Nevertheless, Neureiter and Binder (2022) show that many MBSE tools are still strongly tailored to specific domains. Integration across tool and method boundaries remains a key challenge in industrial practice. A promising way to improve traceability is the use of semantic model links. Here, ontologies are used to formally define terms, relationships and rules and to mediate between models (Wu et al. 2024; Grüninger 1995). For example, terms such as "torque" or "force" can be described uniformly across mechanical and electrical models. Klare et. al (2021) build on this idea of a uniform system model and propose a framework that allows to couple different models of different domains using a specialized language to express semantic dependencies. However, they do not provide domain-specific models or concrete concepts on how to map models that lack semantics.

Domain-Specific Languages (DSLs) and tailored query languages provide a more technical means for linking model elements. While they allow fine-grained specification of cross-model relationships (Guo et al. 2020), they are often complex and require deep domain and tool knowledge. Standardization efforts for DSL development aim to make such approaches more scalable and maintainable (Gupta et al. 2021).

Beyond modeling languages and semantics, various tools support traceability across domains. Platforms such as Smartfacts enable consistent traceability management over multiple tools to fulfill compliance and safety requirements (Ebert and Ray 2021). However, these tools are often closed-source and offer limited extensibility. Meanwhile, co-simulation platforms like FMI support runtime interaction of heterogeneous models but are not focused on structural or semantic traceability (Schweiger et al. 2019). Current research explores the use of AI to detect semantic overlaps between models and automate traceability link creation (Wu et al. 2024), while digital twin approaches provide runtime representations of systems for real-time traceability (Tao et al. 2019).

An additional architectural method for linking models is the Contact and Channel Approach (C&C²) introduced by Albers and Wintergerst (2014). This approach models physical interactions between system elements using Working Surface Pairs (WSP), Channel and Support Structures (CSS), and Connectors (C). Extensions have been developed to integrate this method into SysML and simulation environments such as Simulink (Zingel et al. 2012).

Despite this breadth of methods and tools, several research gaps remain. First, the creation and maintenance of cross-domain links is often performed manually, leading to high effort and error potential. Second, most approaches are constrained to specific tools or platforms, which limits interoperability in heterogeneous development environments. Third, only a few methods address all three key domains, mechanical, electrical, and software, with equal depth. Lastly, the visualization and accessibility of traceability information is frequently inadequate, limiting usability and decision support.

Scalability also remains a challenge. Large CPS such as modern vehicles can involve up to 80 million model artifacts (Eigner et al. 2014), requiring efficient, automated traceability mechanisms. Furthermore, software-based models and validation pipelines are often poorly integrated into traceability solutions. In light of these challenges, this paper addresses the following research question. How can cross-domain traceability in CPSs be automated and semantically enriched across mechanical, electrical, and software domains to improve interoperability, maintainability, and scalability in model-based system development?

To answer this question, we propose a method based on the C&C² approach. The method extends C&C² by introducing a mapping language and tool support for cross-domain model linking and traceability analysis. The following section outlines the methodology, including the underlying representation of models, the structure of semantic mappings, and the generation of traceability reports.

3 Methodology

To address the limitations identified in the state of the art, this paper proposes a method for cross-domain traceability based on the C&C² approach. The method enables semantic model enrichment and structured linking between mechanical and electrical domains. This section describes the methodology in three parts: (1) a semantic mapping concept, (2) an extension of CAD semantics using C&C², and (3) the generation of traceability reports.

3.1 Describing Inter-Domain Connections

Information must be represented in some form in both models to connect different models of different domains. For example, let's look at the momentum introduced onto an axis by an electric motor on the electrical side. A current induces a magnetic field and, therefore, a force on the magnets on our axis. On the mechanical side, for example, during simulations, we take the maximum force as a static input to our simulation. Electrical and mechanical models contain the information force or maximum force. We call this information, which is present in multiple models, the semantic overlap of these models. The problem is that these semantic overlaps are often indirect, so there is often no way to formulate a direct mapping by making a simple value assignment as we did in the electrical motor example since the connection is either unknown or some transformation is needed. To solve this, other constraints or assumptions must be introduced, such as physical laws or expert knowledge, to derive values from the different models that can be directly assigned. A similar problem arises when models of specific components are missing, for example, assuming that there is no exact model of the brake system hydraulics as it is still under development. Still, due to expert knowledge, a signal sent by the brake controller can be mapped directly to the brake cylinder using a lookup table for the brake pressure. Ideally, during development, architects define a high abstraction model of the system in the beginning stages, which is then refined by each department as soon as they create and refine their models. Parallel to the model refinement, the mappings can be concretized and optimized.
To create couplings between models, a language that allows the formal description of semantic overlaps is needed to evaluate these mappings and automatically generate traceability reports. Klare et al. (2021) propose a language to connect different software metamodels, such as PCM, UML, Java, or SysML. This language can also describe mappings between models of other domains, such as mechanical or electrical, or even connections between models of different domains. Therefore, we define a simplified version of this language that is suitable for the most common use cases:

$$\{\mathrm{ID\_Element\_1}\} <-> \{\mathrm{ID\_Element\_2}\}\text{: Optional } [\{\mathrm{Instructions}\}] \tag{1}$$

The first part names the two elements that need to be connected. In the optional instructions space, all parameters of the model elements can be accessed, and essential mathematical functions can be used. The instruction's part is optional to allow the possibility to define relations that may not be quantified or specified due to missing knowledge or other reasons but will still be included in the traceability report. Before execution, this simple language is compiled into the mapping language Klare et. al (2021) proposed. This also allows, if needed, the usage of conditional and other statements for some complex relations. When applied to the example from the beginning, we can define that relationship in the simplified mapping language using the following syntax:

$$\mathrm{motor1} <-> \mathrm{cad\_axis\_1}: \mathrm{motor1.max\_torque} = \mathrm{cad\_axis\_1.max\_torque} \tag{2}$$

Models like CAD lack semantic information. This is due to how CAD models are stored; they comprise many geometric triangles with no semantic information, making it hard to create mappings. To map information between CAD models and other models from different domains, an intermediate layer is needed to store additional information. The next part will describe how this layer could be built using the C&C² approach.

3.2 Modeling CAD Semantics using the C&C²-Approach

We used the C&C² approach (Albers and Wintergerst, 2014) to add semantics to CAD models. The approach works with three key elements that can be used to describe the embodiment and function of technical systems: The Working Surface Pair (WSP); The Channel and Support Structure (CSS), and the Connector (C):

- The *Working Surface Pair* represents surface interactions between two surfaces—solid bodies or generalized interfaces of liquids, gases, or fields—that come into contact. These interactions facilitate the exchange of energy, material, or information. A WSP can comprise two CSS bound together or one CSS and one Connector if the CSS reaches the system boundary.
- *Channel and Support Structures* are volumetric elements that establish connections between two Working Surface Pairs. They provide pathways through which matter, energy, or information can flow, encompassing solid bodies, liquids, gases, or field-permeated spaces. The CSS corresponds to the parts and geometries used in the CAD model.
- *Connectors* account for the system's external effects and bridge it with its environment. Positioned outside the primary design area, Connectors include a representative Working Surface and serve as parameterized models of the system's surrounding context. They exist within the observation boundary but remain outside the design scope.

Figure 1 presents the proposed Unified Modeling Language (UML) Profile for the C&C² approach, which facilitates its implementation within UML and SysML environments. The profile defines CSSs and Connectors as stereotypes extending the class type, enabling their instantiation. The WSP is modeled as an extension of the association type, as it connects two CSPs or one CSP and one Connector, consistent with the principles of the C&C² methodology.

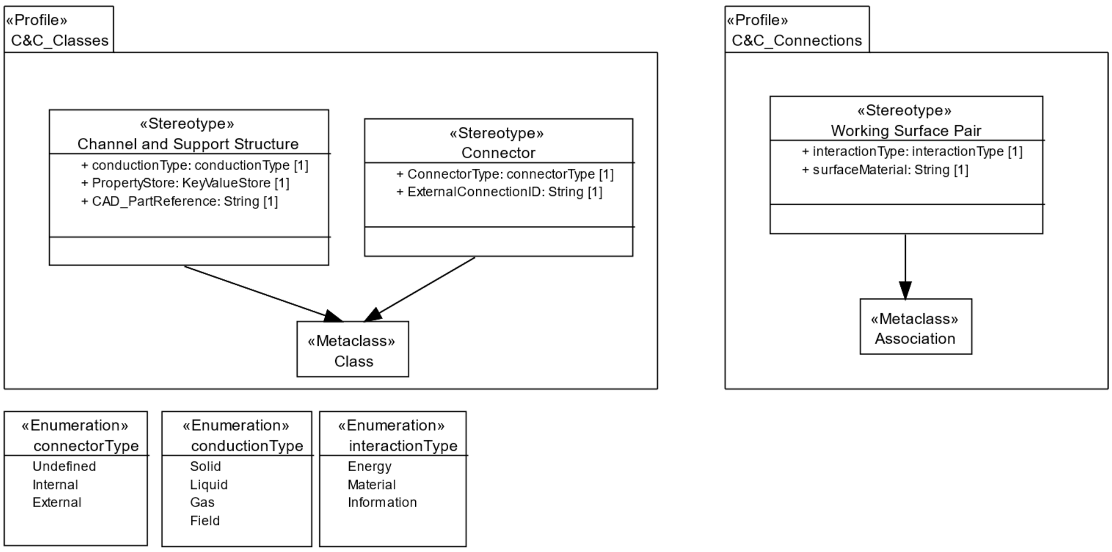

Figure 1: Proposed UML-Profile for the C&C² Model

The Channels and Support Structures can be directly mapped to CAD models using the CAD Part-Reference property. However, the CSPs include dedicated mapping storage for custom properties that are not mapped through the CAD Part-Reference property—such as material properties, mechanical constraints, or other relevant design data. This ensures that all necessary parameters can be defined,

stored, and accessed within the UML profile, providing flexibility for various design requirements. Connectors are categorized into two types: internal Connectors, which indicate boundaries within the design space, and external Connectors, which link the design to models in other domains, such as electrical or thermal systems. The external connection ID references these external mappings created in the mappings language introduced in the previous chapter.
Figure 2 shows our editor for the C&C^2 Model Profile, which we implemented for easy creation and modification of C&C^2 models. Based on a CAD model (top right) of a brake disk connected to an axis, a tree-based element viewer (top left), and the C&C^2 Model (bottom), the engineer can see the semantic information and, later, the couplings in this all-in-one view. In the C&C^2 Viewer, Channels and Support Structures are represented by the grey boxes, Connectors by the orange square, and WSPs by the dashed lines. To reduce complexity, the C&C^2 view supports a hierarchical representation. This means that a system can be divided into subsystems. In a subsystem view, the internal connector makes it possible to interact with other subsystems. A subsystem is also represented as a CCS structure on a top level. In this example, the external connector attached to the axis can be associated with the control signals sent to the Electronic Speed Controller (ESC) of our electric motor. This association is valid because the motor — connected directly to the axis — generates torque in response to specific control signals, thus creating a semantic relationship between the signal values and the resulting torque on the axis. Figure 4 displays a schematic overview of the resulting dependency tree, including this mapping. Model elements are displayed as circles; they correspond to the CCSs and WSPs in Figure 2.

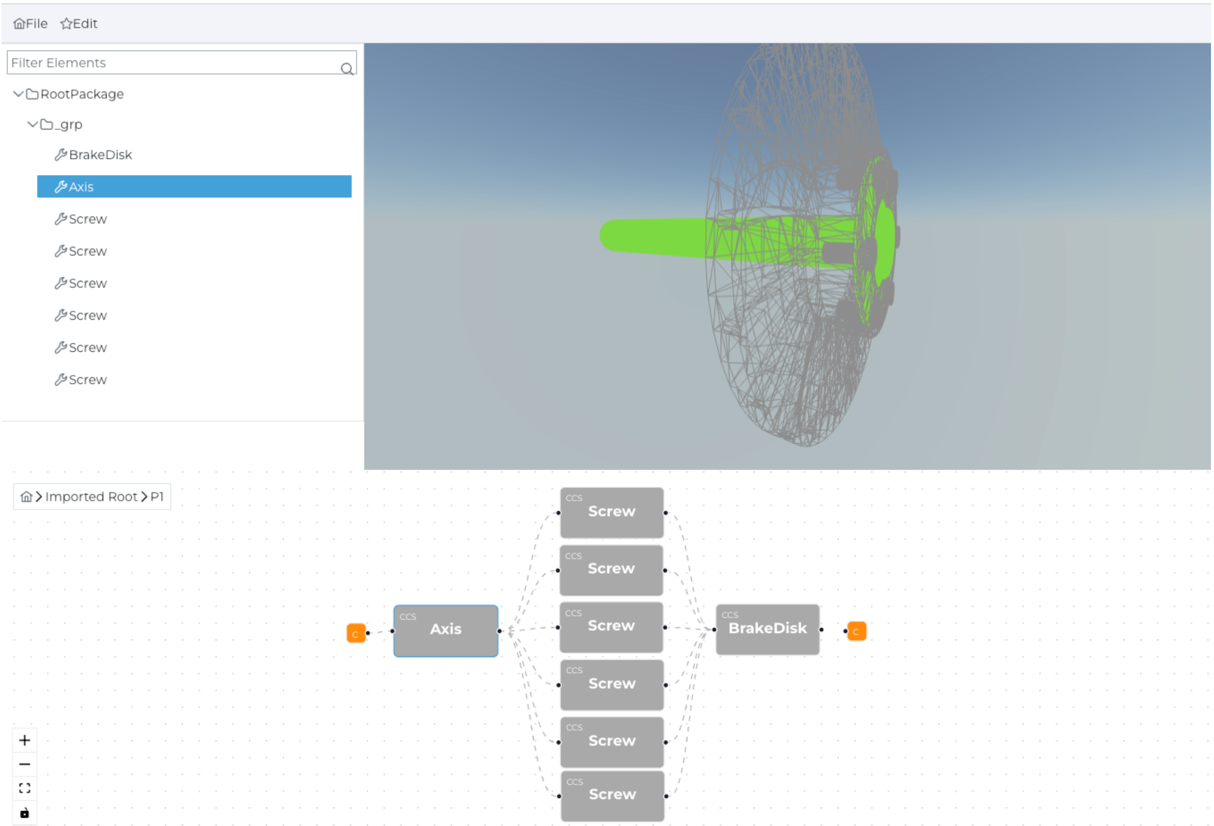

Figure 2: Proposed C&C^2 Editor for the UML-Profile

Additionally, models from the E/E domain can be seen. The boxes indicate model groups containing interconnected model elements, but they are not displayed here for simplicity. Using our brake example, we can see the axis connected to the brake disk using screws. The axis is then linked using an external connector with the ESC Speed Signal, controlling directly the speed of the motor driving the axis, which is included in our E/E Model. We can now build our dependency tree as these signals are further connected to requirements, hardware, and software model elements. The dependency tree now connects the mechanical model and the communication model, which is then linked to other models in the electrical domain and models such as software or requirements. When all model elements become interconnected, the resulting dependency trees may grow to include every artifact in the system. Such expansive dependency structures complicate dependency analysis, making it challenging to pinpoint where relevant changes or potential impacts occur. Therefore, it's crucial to apply targeted filtering mechanisms or analytical strategies to effectively focus on significant dependencies and efficiently identify critical points of change or influence during the generation of traceability reports.

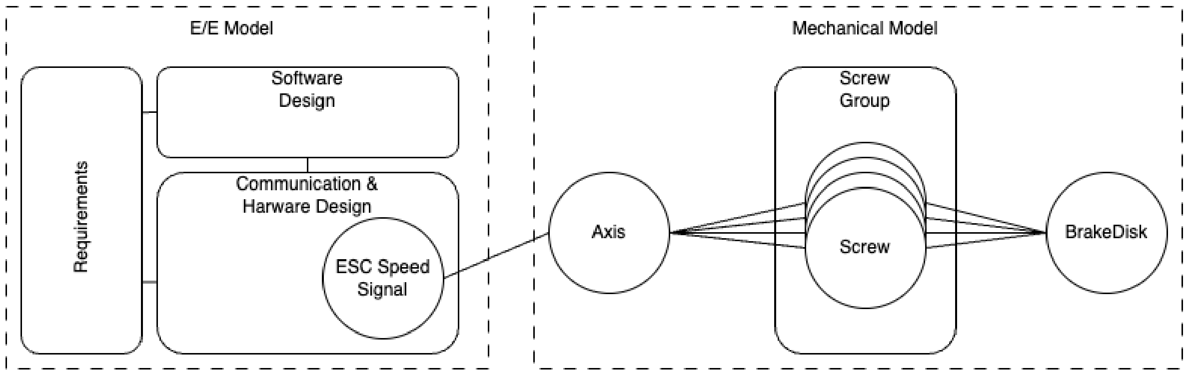

Figure 3: Schematic Overview of the Dependency Tree

3.3 Generation of Traceability Reports

The traceability reports can now be generated based on the mappings created in the previous step. Reports, in the form of tables that include all related object IDs and names, are generated on two occasions: First, when a model element is selected, and the engineer wants to perform a traceability analysis. In this case, the requested element is set as the source of the report. Secondly, if a change occurs, all change-affected model elements are set as sources, and the analysis is performed. These traceability reports are generated using a dependency tree built on all available information. Since in modern CPS, ideally, all model elements are interconnected, this would lead to a dependency tree including every model element. Therefore, filter mechanisms can be applied during the generation of the report. We propose the following filter mechanisms to reduce the amount of information displayed:

Filter by Distance: This filter setting defines the maximum number of hops included from a source element. For example, if the distance is set to five, the report contains only model elements reachable from any source in five steps if set to five. This setting assumes that a change introduced to a system only propagates to a limited number of model elements, decreasing in impact the further away from the initial impact source. When applying a distance filter set to three steps, starting from the brake disk in our example, the report would include the axis, the screws connecting the axis to the brake disk, and the ESC motor speed control signal, but not distant elements such as software modules controlling the ESC.

Filter by Deviation: This filter set uses the instructions provided during the mapping definition. It can only be used during impact-based traceability reports. According to the definition of semantic dependency, it propagates a change through the tree and calculates the difference between the old and new values. If the deviation is more significant than a specific threshold value, such as 10 %, it shows the tree element. The problem with this filter is that a quantifiable formal description of the semantic overlap must be available to apply it. For example: If the brake disk thickness changes by more than 10% compared to its previous design, this filter would highlight CAD elements directly affected by that significant geometric alteration, such as mounting screw lengths or structural load analyses, but ignore minor changes below the defined threshold.

Combined Filter: This filter is used if formal instructions are absent for every element, making deviation filtering impossible. In this setting, a combination of deviation and distance is used: For each model coupling with a formal description available, the deviation filter is used. If no description is set, the distance filter is used. Transferred to our brake example: In scenarios where some couplings like the torque-to-speed relationship are formally quantified, a significant deviation (e.g., ESC control signal changing torque by over 10%) would trigger inclusion. If no formal rule is set, such as signal to software dependencies, the system defaults to including only elements within a certain number of hops, based on the last element identified by the deviation filter.
Filter by Type: This filter setting is used when an analysis focusing on specific types of model elements is needed. For example, when tracing requirements, a path through the dependency tree is searched from the source element to the closest n model elements of a particular type to determine the impact of a change. This filter setting can also be combined with the other filter proposed above. An example would be to focus only on requirements impacted by a mechanical change, setting the filter to Requirements and CCS. This filter would trace from the brake disk through connected elements until the nearest requirements in the dependency tree are reached, displaying how the brake disk design modifications relate to system requirements.

4 Discussion

To evaluate the applicability of the proposed traceability method in a realistic development context, a case study was conducted using an electromechanical brake system of an electric vehicle. The objective of this case study was not to assess the behavior of the braking system itself, but to demonstrate the feasibility and limitations of the C&C²-based approach for semantically linking domain-specific models and generating traceability information. The case study thus serves as a practical validation of the method's core components: the semantic model enrichment, the mapping concept, and the reporting mechanism.
The scope of the case study was restricted to a single rear wheel of the vehicle. For the mechanical domain, a detailed CAD model of the brake assembly was created using PTC Creo. The electronic and software-related architecture was modeled in PREEvision and comprised functional, logical, hardware, and communication layers. These two models served as input to the modeling environment developed for this study, which supports the C&C² modeling profile and integrates the mapping concept introduced in the previous section.
The modeling process followed the five-step workflow outlined in Section 3.2. In the first step, the CAD model was imported into the tool, and all CSSs were automatically extracted. Nested assemblies were also recognized and hierarchically organized as subsystems. In the second step, internal Connectors and WSPs were manually defined by the engineer, capturing relevant physical interactions such as the contact between the brake pad and the disc. Subsequently, external Connectors were identified to establish links between the mechanical model and elements of the E/E model, such as motor speed signals, sensor positions, and brake control valves. Finally, semantic mappings were defined for each relevant external Connector using the lightweight mapping syntax introduced in Section 3.1. This step allowed the generation of traceability links across domains.
While the initial steps of importing and parsing the CAD model were fully automated, the manual definition of WSPs, external Connectors, and mappings required substantial effort. The modeling of the C&C² structure and semantic mappings for this single-wheel scenario took approximately ten hours, highlighting the need for automation in future developments.
Once the mappings were complete, traceability reports were generated in tabular format. These reports included both internal relationships, within either the mechanical or E/E models, and external links across the two domains. To manage the potentially large number of artifacts, filtering mechanisms as described in Section 3.3 were applied. In particular, the combination of "Filter by Type" and "Filter by Deviation" proved to be the most effective for tracing requirement-related changes.

These filters significantly reduced the information density in the reports while preserving relevance and semantic trace depth.
Overall, the case study demonstrated that the proposed method is applicable to complex, real-world CPS models and supports structured traceability across domains. At the same time, the case revealed weaknesses, especially in terms of the manual modeling effort and usability. These aspects are critically discussed in the following chapter. The case study confirms the applicability of the proposed method for enabling cross-domain traceability between mechanical, software and electrical models. By combining semantic enrichment based on the C&C²-approach with a structured mapping mechanism, the method facilitates the formalization of relationships between domain-specific model elements. This structured approach supports the generation of traceability information that is otherwise not available in conventional CAD or E/E modeling tools. A major strength of the method lies in the automated extraction and structuring of CSS from CAD models. This feature significantly reduced the manual modeling effort in the initial phases. However, the manual creation of WSPs, external Connectors, and semantic mappings remains a time-intensive task. In the case study, this accounted for more than 70% of the total modeling effort. These observations highlight a key limitation of the current implementation and point to the need for further automation, particularly in the identification of potential mapping candidates and the suggestion of semantic relationships. Traceability reports generated through the method proved to be a valuable tool for analyzing interdependencies across domains. The built-in filter mechanisms, especially the combination of “Filter by Type” and “Filter by Deviation,” were effective in reducing the volume of trace data and focusing the analysis on relevant artifacts. Nevertheless, the current output format, static Excel spreadsheets, limits the practical use of the reports, especially in collaborative engineering environments. An interactive visualization environment would significantly enhance the usability and interpretability of the results. Scalability of the method was not fully evaluated in this case study, as the scope was limited to a single wheel subsystem. While the method is designed with modularity and hierarchical modeling in mind, further validation in larger system contexts is necessary to confirm its performance under more complex conditions. Additionally, the current version of the mapping language covers basic use cases but would benefit from enhanced expressiveness to handle more advanced, conditional or nonlinear mappings.
In summary, the case study provides initial evidence that the method is capable of supporting traceability tasks in real-world CPS development contexts. The structure, formalism, and tooling presented form a consistent methodical framework for cross-domain model integration. However, further work is needed to address automation, scalability, and integration into engineering toolchains. These next steps will be essential to establish the method as a robust solution for traceability in complex CPS.

5 Future Work and Conclusion

Cross-domain traceability in CPS can enhance product development. In particular, the systematic integration of different domains and the ability to track changes throughout the development process create a solid foundation for high-quality and maintainable products.

In future work, the focus should be on automating the mappings to reduce the high manual effort. The use of AI technologies to identify semantic overlaps could be crucial here.

Currently, visualization of traceability reports is difficult to read or analyze. The reports are presented in static formats such as Excel tables, which leads to a flood of information for large models. To increase user-friendliness, interactive and integrated solutions would be necessary. An intuitive interface that allows engineers to explore dependencies and dynamically filter data based on specific criteria is needed. Such tools could support decision-making processes by providing real-time insights into the potential impact of changes. Data security challenges and cross-divisional data exchange, particularly between suppliers and manufacturers, must also be addressed.

Since we only tested the methodology using the brake HiL case study, we did not gather empirical data on performance or scalability. Studies proving its effectiveness in industrial applications are needed. It needs to be validated whether integrating the method into existing CI/CD pipelines could also improve its efficiency by providing constant feedback if changes occur.

In summary, the method presented represents a step towards efficient product development in CPS. Through targeted further development and the solution of the limitations shown, automatic traceability could become an essential tool for engineers facing the increasing demands of modern complex systems.

Acknowledgments

The work in this paper is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – SFB 1608 – 501798263

References

Journal article:

Abhaya 2021 — Abhaya, Lalitha: UAF (Unified Architecture Framework) Based MBSE (UBM) Method to build a System of Systems Model. In: INCOSE International Symposium 31 (2021), Nr. 1, S. 227–241

Ebert and Ray 2021 — C. Ebert ; R. Ray: Test-Driven Requirements Engineering. In: IEEE Software 38 (2021), Nr. 01, S. 16–24. URL http://doi.ieeecomputersociety.org/10.1109/MS.2020.3029811

Chaparala and Hartman 2013 — Chaparala, Ramya T. ; Hartman, Nathan W. ; Springer, John: Examining CAD Interoperability through the Use of Ontologies. In: Computer-Aided Design and Applications 10 (2013), Nr. 1, S. 83–96

Gu et al. 2024 — Gu, Pengfei ; Zhang, Yuteng ; Chen, Zhen ; Zhao, Chun ; Xie, Kunyu ; Wu, Zhuoyi ; Zhang, Lin: X-RMTV: An Integrated Approach for Requirement Modeling, Traceability Management, and Verification in MBSE. In: Systems 12 (2024), Nr. 10, S. 443

Klare et al. 2021 — Klare, Heiko ; Kramer, Max E. ; Langhammer, Michael ; Werle, Dominik ; Burger, Erik ; Reussner, Ralf: Enabling consistency in view-based system development — The Vitruvius approach. In: Journal of Systems and Software

171 (2021), S. 110815. URL https://www.sciencedirect.com/science/article/pii/S0164121220302144

Li et al. 2024 — Li, Wenfei ; Wang, Ming ; Huang, Chao ; Li, Boyuan: A Survey of Hybrid Braking System Control Methods. In: World Electric Vehicle Journal 15 (2024), Nr. 8

Neureiter and Binder 2022 — Neureiter, Christian ; Binder, Christoph: A Domain-Specific, Model Based Systems Engineering Approach for Cyber-Physical Systems. In: Systems 10 (2022), Nr. 2, S. 42

Schweiger et al. 2019 — Schweiger, G. ; Gomes, C. ; Engel, G. ; Hafner, I. ; Schoeggl, J. ; Posch, A. ; Nouidui, T.: An empirical survey on co-simulation: Promising standards, challenges and research needs. In: Simulation Modelling Practice and Theory 95 (2019), S. 148–163. URL https://www.sciencedirect.com/science/article/pii/S1569190X1930053X

Venkatesh et al. 2024 — Venkatesh, Appalabathula ; Kumar, S. Phani ; Kiran, S. S. ; Gurucharan, K.: Cyber-physical systems for hybrid braking control techniques in hybrid electric vehicles. In: International Journal of Information Technology (2024)

Wu et al. 2024 — Wu, Shouxuan ; Wang, Guoxin ; Lu, Jinzhi ; Hu, Zhenchao ; Yan, Yan ; Kiritsis, Dimitris: Design ontology for cognitive thread supporting traceability management in model-based systems engineering. In: Journal of Industrial Information Integration 40 (2024), S. 100619. URL https://www.sciencedirect.com/science/article/pii/S2452414X24000633

Zingel et al. 2021 — Zingel, Christian ; Albers, Albert ; Matthiesen, Sven ; Maletz, Michael: Experiences and Advancements from One Year of Explorative Application of an Integrated Model-Based Development Technique Using C&C²-A in SysML. In: International Journal of Computer Science 34-39 (2012), S. 165–181

Contribution to a collective work (e.g. contribution to a conference volume):

Albers and Wintergerst 2014 — ALBERS, Albert ; WINTERGERST, Eike: The Contact and Channel Approach (C&C²-A): relating a system's physical structure to its functionality. In: CHAKRABARTI, A.; BLESSING, L. (Hrsg.): An Anthology of Theories and Models of Design: Philosophy, Approaches and Empirical Explorations. London : Springer, 2014, S. 151–171

Blockwitz et al. 2012 — Blockwitz, Torsten ; Otter, Martin ; Akesson, Johan ; Arnold, Martin ; Clauss, Christoph ; Elmqvist, Hilding ; Friedrich, Markus ; Junghanns, Andreas ; Mauss, Jakob ; Neumerkel, Dietmar ; Olsson, Hans ; Viel, Antoine: Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models. In: Proceedings of the 9th International MODELICA Conference, September 3-5, 2012, Munich, Germany : Linköping University Electronic Press, 2012 (Linköping Electronic Conference Proceedings), S. 173–184

Grüninger and Fox 1995 — Grüninger, Michael ; Fox, Mark: Methodology for the Design and Evaluation of Ontologies (1995)

Gupta et al. 2021 — R. Gupta ; S. Kranz ; N. Regnat ; B. Rumpe ; A. Wortmann: Towards a Systematic Engineering of Industrial Domain-Specific Languages. In: 2021 IEEE/ACM 8th International Workshop on Software Engineering Research and Industrial Practice (SER&IP), 2021, S. 49–56

Guo et al. 2020 — Guo, Jiangmin ; Lu, Jinzhi ; Ding, Jie ; Wang, Guoxin: A Model Query Language for Domain-Specific Models. In: 2020 5th International Conference on

	Mechanical, Control and Computer Engineering (ICMCCE), 2020, S. 1207–1213
Vetter and Sax 2021	Vetter, Andreas ; Sax, Eric: Hierarchical Versioning to Increase Compatibility in Signal-Oriented Vehicle Networks, Bd. 182. In: Selvaraj, Henry; Chmaj, Grzegorz; Zydek, Dawid (Hrsg.): Proceedings of the 27th International Conference on Systems Engineering, ICSEng 2020. 1st ed. 2021. Cham : Springer International Publishing; Imprint Springer, 2021 (Lecture Notes in Networks and Systems, 182), S. 435–444

Book:

Eigner et al. 2014	Eigner, Martin ; Roubanov, Daniil ; Zafirov, Radoslav: Modellbasierte virtuelle Produktentwicklung. Berlin, Heidelberg : Springer Vieweg, 2014
Friedrich and Müller 2024	Friedrich, Horst E. ; Müller, Gerd: Werkstoffe und Bauweisen in der Fahrzeugtechnik. Berlin, Heidelberg : Springer Vieweg, 2024 (ATZ/MTZ-Fachbuch)
Tao et al. 2019	Tao, Fei (Hrsg.); Zhang, Meng (Hrsg.); Nee, A.Y.C. (Hrsg.): Digital Twin Driven Smart Manufacturing : Academic Press, 2019