Proceedings of the Design Society, Volume 6: DESIGN 2026
https://doi.org/10.1017/pds.2026.10645

Semantic fusion of SysML elements for model integration utilizing knowledge graphs

Roman Stephan[1]✉, Zhiwei Zhou[2], Simon Gottschalk[2] and David Inkermann[1]

[1] *Clausthal University of Technology, Germany,*
[2] *L3S Research Center, Leibniz Universität Hannover, Germany*

✉ stephan@imw.tu-clausthal.de

ABSTRACT: While top-down System Engineering supports the definition of system boundaries and interfaces, practical implementation often proceeds in a bottom-up manner, resulting in the need for model integration of SysML models. This is currently hindered by inconsistent naming, different abstraction levels, and unclear interfaces. In this work, a novel approach to integration is proposed, utilizing RDF knowledge graphs and LLM-driven entity alignment with similarity thresholds to perform semantic fusion. A practical use case shows correct consolidation based on cosine similarity thresholds.

KEYWORDS: model-based systems engineering (MBSE), knowledge graph, model integration, knowledge management

1. Introduction

Systems Engineering (SE) is an interdisciplinary approach aimed at supporting the successful development of complex products by considering all life cycle phases of the system of interest (SoI) (Walden, 2023). Model-Based Systems Engineering (MBSE) represents a structured application to support systems engineering activities and to enable a structured, consistent representation of system elements and their relationships using formal models (Friedenthal et al., 2015). Due to its potential to enhance system understanding and communication among stakeholders, MBSE has gained increasing attention in both industry and academia (Berschik et al., 2023). However, despite these advantages, implementing MBSE in practice is associated with significant challenges that often limit the effective utilization.

1.1. Model integration challenges in the context of MBSE

When applying MBSE, development decisions and results are stored in a central, coherent system model. Product characteristics are concretized following a top-down approach, involving the breakdown of the system for early definition of system boundaries and interfaces in a top-down manner (Walden, 2023). This aims at defining interfaces down to the lowest system level before the internal behaviour and structure are determined. In practice, this approach is rarely feasible, as the relevant knowledge needed for a detailed interface definition is often distributed across different development departments and is mostly based on experience from previous projects. The definition of the overall vehicle architecture in the automotive sector, for example, is carried out in parallel in different organizational silos. Development teams focus on individual subsystems, such as braking or steering, and rarely possess an integrated view of the overall architecture (Krog et al., 2025). This means that preliminary interfaces are defined first, but detailed interface modelling can only be done during architecture definition at the lower subsystem level, without an overarching system architecture in place beforehand. This results in simultaneous top-down and bottom-up development, which is a suitable approach from both an efficiency and a risk perspective (Inkermann & Ammersdörfer, 2026) but leads to the existence of multiple subsystem models. In the absence of an integrated architecture at the higher level, it is difficult to

https://doi.org/10.1017/pds.2026.10645 Published online by Cambridge University Press

assess whether the defined interfaces are compatible, whether they overlap, or are inconsistent. Consequently, cross-system functionalities, such as recuperation of energy, cannot be reliably evaluated. This results in the need to integrate different subsystems in order to ensure functionality, check compatibility of interfaces, or as an input for subsequent development activities. A further scenario in which model integration is important is cross-project system model reuse (Lange et al., 2018). It has been stated that a central base, also known as a repository, is vital for efficient reuse (Sivaloganathan & Shahin, 1999). Therefore, the creation of the repository is achieved by summarizing, i.e., integrating, different model elements from different system models. For both scenarios, the term system model refers to SysML-based representations that serve as the data basis for integration, while integration encompasses their combination, extension, or merging. Model integration is understood as the systematic consolidation of two or more SysML models, or subsets thereof (e.g., functions or components), into a single coherent SysML model. During the model integration, challenges arise which are illustrated in Figure 1 and consist of three distinct ones:

- Insufficient definition of boundaries and interfaces
- Heterogeneous abstraction levels of modelling
- Inconsistent use of descriptions and model names

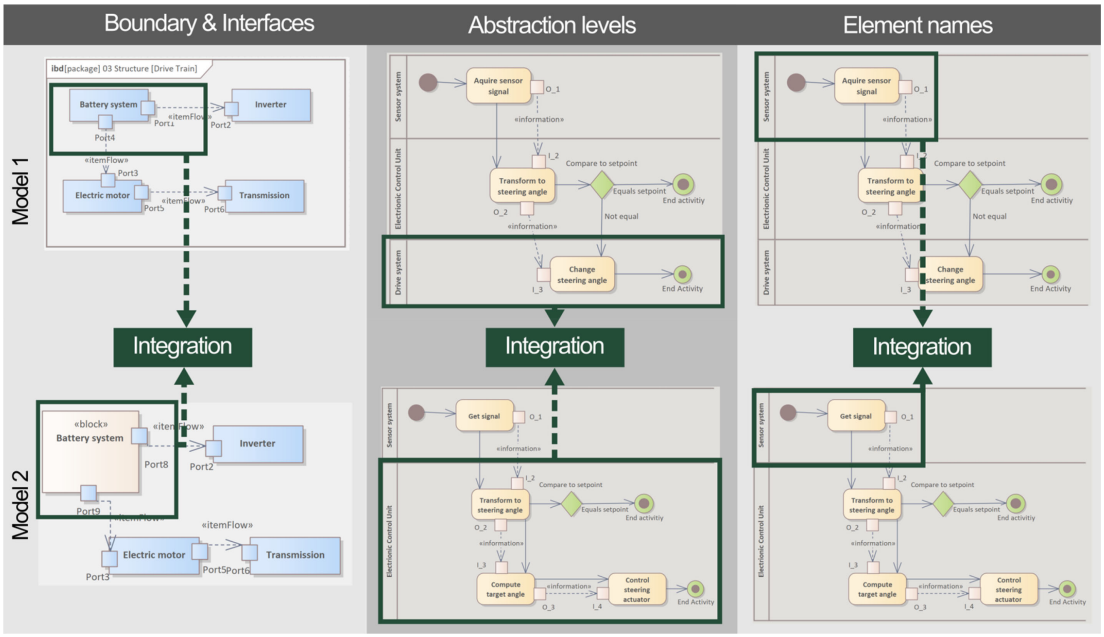


Figure 1. Challenge types in the integration of SysML models

The first challenge results from the insufficient definition of subsystem boundaries and interfaces. This may lead to overlapping structures or redundancies in the overall system architecture (Fosse & Delp, 2013). If interfaces are not defined correctly, a subsystem may be part of multiple other systems, leading to duplication of design. The second challenge concerns heterogeneous levels of modelling detail and abstraction. Subsystems are often developed using different conventions, resulting in incompatible model depths that hinder seamless integration (Trujillo et al., 2020). For example, functions can be defined at different, ambiguous system levels, making uniform integration not possible. The third challenge arises from inconsistent naming of elements. In the absence of uniform naming conventions, identical real-world entities may be represented by different terms or, conversely, the same term may refer to different elements across subsystems. This is particularly problematic when elements appear in multiple diagram types, for example as swim lane components in activity diagrams and as structural elements in internal block diagrams (Schumacher & Inkermann, 2023). These challenges impede the integration of consistent architectural models and thereby limit their effective use for design definition. Overall, it can be concluded that current integration difficulties significantly complicate MBSE implementation, both in bottom-up development environments and in model reuse scenarios. Against this background, the present work explicitly focuses on the systematic support for model integration in MBSE, which serves as the basis for the research objectives and questions outlined in the following section.

https://doi.org/10.1017/pds.2026.10645 Published online by Cambridge University Press

1.2. Research objective and questions

The objective of this work is to reduce semantic ambiguities and inconsistencies arising from heterogeneous naming and representation of SysML models in the context of MBSE. This is achieved by enabling a semantic comparison of system element names, thereby facilitating more efficient and consistent model integration. For this purpose, methods for integration are employed, as they provide a structured basis for applying entity alignment techniques and offer flexibility for extending functionalities such as semantic comparison and element fusion. Based on this objective, the following research questions are addressed:

- **RQ1**: *How can the systematic consolidation of two or more SysML models in the course of model integration be supported?*
- **RQ2:** *How can the integration of semantically divergent SysML elements be enabled?*

Answering these research questions contributes to improving the integration of SysML models for both subsequent development activities and cross-project reuse. In this scientific work, the following procedure was followed. Based on a review of existing literature on model integration in MBSE a research gap was identified. A solution was developed conceptually based on an established procedure, and an implementation was carried out using an industry near example. The remainder of this paper is structured as follows. Chapter 2 presents an overview of existing integration approaches in MBSE, as well as an introduction to knowledge graphs and entity alignment using Large Language Models. Chapter 3 describes the proposed method, including its purpose, procedural steps, and expected benefits. Chapter 4 demonstrates the technical implementation, followed by a summary in Chapter 5. Finally, Chapter 6 discusses the limitations of the work and outlines directions for further research.

2. Model integration and semantic fusion in MBSE

This section establishes the conceptual and methodological foundation for addressing the research objective, reviewing and critically assessing existing approaches to model integration in MBSE. Most existing work on model integration in MBSE focuses on the interoperability between SysML and other modelling tools rather than on the systematic integration of multiple SysML models. A prominent example is the integration of SysML with Simulink models, which is primarily aimed at enabling co-simulation and tool interoperability (Cao et al., 2011; Kapos et al., 2021). In these approaches, model integration is typically understood as a technical interface problem between different modelling languages rather than as a semantic consolidation of heterogeneous SysML structures. Other contributions interpret model integration as an extension of SysML with additional analytical capabilities, such as safety and reliability (S&R) analysis (Hu et al., 2023; Z. Li et al., 2025). While these approaches enhance the expressiveness of SysML, they do not explicitly address the integration of independently developed subsystem models that exhibit divergent naming conventions, modelling depths, and structural semantics. Model integration is also considered within broader MBSE process chains, in which SysML models serve as intermediate artefacts supporting downstream activities (Schumacher et al., 2025). In such cases, integration is embedded in predefined workflows and presupposes a high degree of consistency between models, thereby avoiding the integration challenges illustrated in Figure 1. With regard to model reuse, element bases have been introduced, for example for the creation of functional libraries (Spütz et al., 2023) or architecture databases (Slobin et al., 2024). Figure 2 shows a proposal for a generic reuse process for system models in MBSE which is comprised of five steps in total: (1) provision of the models, (2) determination of reusable elements, (3) insertion into the target system model, (4) adaptation of the elements to the target model, and (5) validation of functionality. Here the creation of functional libraries and architecture databases is included in the first step. However, these approaches often assume either pre-aligned data, i.e., data sets that have already been prepared for reuse, or the construction of a central database independent of existing SysML models. Consequently, they provide limited support for integrating existing heterogeneous SysML models originating for instance from different organizational contexts. A critical limitation of the aforementioned approaches is that they typically operate under conditions of minor or non-existent semantic conflicts between models (Trujillo et al., 2020). As a result, they circumvent the core challenges of heterogeneous model integration, such as inconsistent naming, ambiguous semantics, and structural divergence. Their applicability is therefore restricted when confronted with diverse datasets that reflect realistic industrial conditions. Based on this analysis it can be concluded, that there is a lack of approaches that explicitly address the semantic integration of heterogeneous SysML models exhibiting substantial differences in

https://doi.org/10.1017/pds.2026.10645 Published online by Cambridge University Press

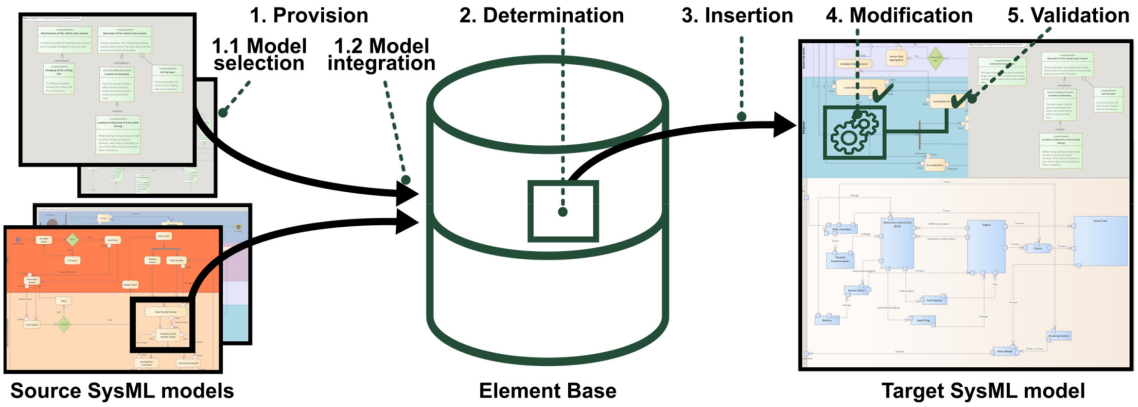


Figure 2. Cross-project reuse process of system models

terminology, structure, and level of abstraction. This gap motivates the exploration of alternative mechanisms capable of supporting robust, scalable, and semantically grounded model integration.

2.1. Entity alignment in knowledge graphs

A knowledge graph (KG) consists of entities and their relations, represented as nodes and edges respectively (Hogan et al., 2022). Information can be stored in various formats; however, this work utilises the Resource Description Framework (RDF)[1] in Turtle format, in which data is represented as triples consisting of subject, predicate, and object (Färber et al., 2017). RDF was selected instead of alternative representations, such as property graphs, due to its formal semantics, reasoning capabilities, and compatibility with established semantic technologies (Besta et al., 2024). Entity alignment (EA) refers to the process of identifying semantically equivalent entities across different KGs that represent the same real-world objects (Cheng et al., 2025; Liu et al., 2022). In the context of this work, EA is applied to identify corresponding SysML elements originating from different subsystem models that may differ in naming, structure, or level of abstraction. This step is essential for enabling reliable integration, as it provides the basis for the subsequent consolidation and fusion of aligned elements into a unified SysML model. Traditional EA methods typically rely on supervised learning approaches, including translation-based embeddings and Graph Neural Networks (GNNs), which require labelled training data (Q. Li et al., 2023). Such requirements limit their applicability in engineering contexts where annotated datasets are rarely available. Recent research therefore explores end-to-end EA frameworks driven by Large Language Models (LLMs) (Z. Li et al., 2025), which enable semantic comparison without extensive manual annotation (Cheng et al., 2025; Jiang et al., 2024). For example, the EasyEA architecture first summarises KG information using LLMs to capture entity semantics, then integrates multi-dimensional features, and finally applies hierarchical candidate selection to identify aligned entity pairs (Cheng et al., 2025). These approaches demonstrate robustness in heterogeneous and large-scale scenarios, making them particularly relevant for SysML model integration across organisational boundaries.

2.2. Semantic fusion of elements

Semantic fusion builds upon the results of entity alignment by merging aligned entities and their associated information into a single coherent representation. Within the scope of this work, semantic fusion represents the process through which semantically corresponding SysML elements are consolidated to form an integrated and unambiguous system model. This step transforms aligned element pairs into structurally and semantically consistent architectural structures suitable for reuse in subsequent development activities. State-of-the-art fusion approaches indicate that combining topological context with logical consistency rules significantly improves the integrity and completeness of merged knowledge representations (Lin et al., 2023). Furthermore, the integration of LLMs with KG fusion techniques enables the enrichment of structured data through context-aware reasoning while preserving semantic coherence (Pan et al., 2024). Advanced multimodal fusion approaches incorporate relational, structural, and textual information simultaneously, with methods demonstrating improved performance through link-aware aggregation mechanisms. In this work, semantic fusion is therefore not

[1] https://www.w3.org/RDF/

https://doi.org/10.1017/pds.2026.10645 Published online by Cambridge University Press

treated as a generic data aggregation step, but as a targeted mechanism for constructing a unified system model from SysML models. Together with entity alignment, it forms the operational backbone of the proposed integration approach.

3. Conceptual approach for semantically supported model integration

In this chapter, the proposed concept for supporting the integration of heterogeneous SysML models through KG-based semantic analysis is introduced. In contrast to existing approaches, the presented concept explicitly targets semantic inconsistencies arising from divergent naming conventions, modelling practices, and structural representations. The proposed approach operationalizes the research objectives by combining entity alignment and semantic fusion into a structured integration concept applicable to realistic MBSE environments. The contribution of this chapter lies in the definition of a reusable, semantically grounded conceptual framework that enables the identification, evaluation, and consolidation of semantically equivalent SysML elements across independently developed models. This framework establishes the methodological foundation for a prototype implementation and subsequent evaluation, directly supporting RQ1 and RQ2 by (i) providing a systematic mechanism for model integration and (ii) enabling differentiated integration strategies based on the type and degree of semantic divergence. An overview of the developed concept is illustrated in Figure 3.

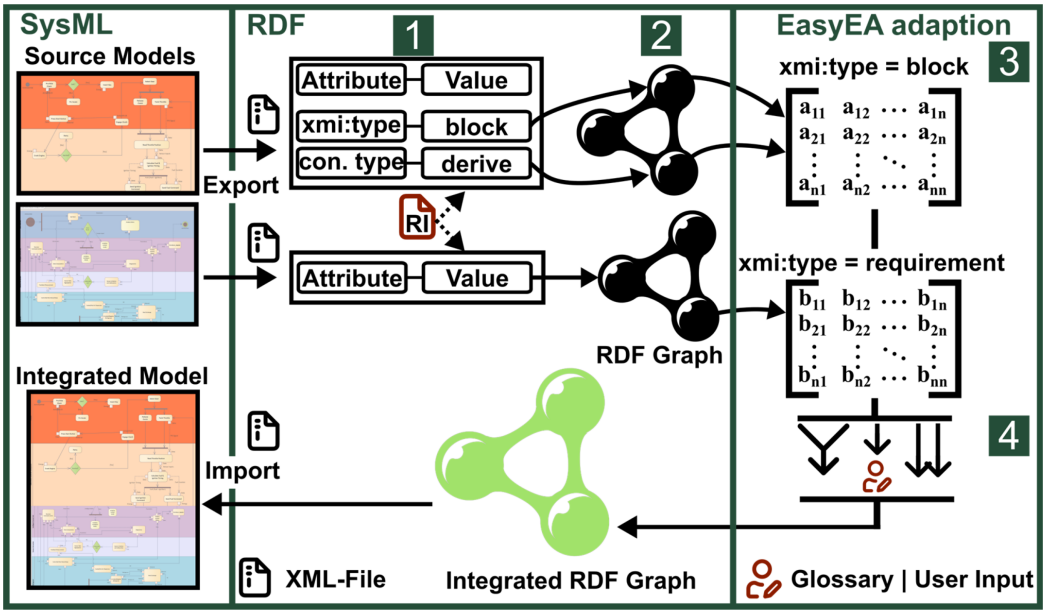


Figure 3. Developed approach for model integration using semantic fusion

The proposed concept is structured into four phases. Which are based on the initial EasyEA framework (Cheng et al., 2025) and extended with a fourth step to allow for KG merging. Once the system model is exported as XML file, the proposed concept automatically carries out the following steps. It takes SysML models as both input and output, ensuring compatibility with existing MBSE toolchains. The first phase consists of the structured extraction of relevant information from SysML models provided in XML format. Relevant information (RI) depicted as a document in red in figure three, such as system elements, relationships, and attributes, is isolated and stored in a separate data structure to ensure flexibility with respect to implementation-specific requirements and future extensions. This abstraction layer decouples the conceptual integration logic from concrete implementation details and supports controlled data transformation. In the second phase, the extracted information is transformed into a KG representation. System elements are modelled as nodes, while relationships are represented as edges. Additional metadata, including element attributes such as xmi:type, is preserved to maintain semantic context and enable type-consistent comparison in subsequent phases. This transformation establishes the semantic data foundation required for KG-based analysis. The third phase focuses on semantic similarity assessment, which operationalises the entity alignment principles discussed in Section 2.2. Each node is evaluated against other nodes of identical element types using an extended entity alignment procedure that incorporates the EasyEA method.

https://doi.org/10.1017/pds.2026.10645 Published online by Cambridge University Press

Beyond structural comparison, semantic relationships between element names are analysed using LLMs, and cosine similarity measures are computed to quantify semantic proximity. The resulting similarity values are stored in NxN similarity matrices, with values ranging from -1 (negative correlation) to +1 (positive correlation). Type-based segregation ensures that only semantically and structurally compatible elements are compared, thereby preventing unintended grouping of dissimilar element categories. The fourth phase introduces a structured decision logic for integration based on the calculated similarity values. Three distinct integration scenarios are distinguished. High similarity values indicate strong semantic equivalence, allowing for the automatic merging of elements. Low similarity values signify semantic divergence, resulting in the preservation of separate entities. Intermediate similarity values indicate ambiguity and trigger context-sensitive decision mechanisms were element names are compared against an existing glossary. This glossary, whether based on company-specific definitions or domain-specific, contains a number of terms that have been clearly defined and supplemented with synonyms. If a corresponding entry exists, standardised terminology defined withing the glossary is applied; otherwise, combined naming strategies are used and user validation is initiated. The final phase involves the transformation of the integrated KG back into SysML-compliant XML format. An XML template is employed to re-instantiate the integrated elements in a form compatible with standard SysML tools. The newly integrated elements are initially embedded as structural components within the system model and may subsequently be visualised within the corresponding diagrams by the user. This ensures practical applicability while maintaining separation between semantic integration and graphical representation. Overall, the proposed concept establishes a structured, semantically grounded integration workflow that bridges the gap between heterogeneous SysML model representations and unified architectural consistency. By systematically combining KG transformation, entity alignment, and semantic fusion, the approach provides a robust foundation for scalable, traceable, and reusable model integration in MBSE environments, thereby addressing the limitations identified in Chapter 2 and advancing the state of the art in SysML model integration.

4. Technical implementation and case study

This chapter details the technical implementation of the proposed concept and demonstrates its applicability by an illustrative case study. The focus is on the concrete toolchain and execution of the approach. The implementation aims to demonstrate the feasibility of the proposed integration logic and to illustrate how semantically heterogeneous SysML models can be consolidated within a KG-based framework.

4.1. Execution of the integration pipeline

The technical implementation employs a modular toolchain designed to ensure transparency, reproducibility, and extensibility. SysML models serve as the primary input and are initially exported from Enterprise Architect as XML files, allowing for systematic parsing and transformation. Declarative mapping of XML structures to semantic representations is achieved using YARRRML, a human-readable mapping language that facilitates maintainability of the transformation rules. The generated mappings are processed using RMLMapper to create the RDF graphs in Turtle format. YARRRML was selected due to its clarity and flexibility in representing complex mapping rules, while RMLMapper was chosen for its compatibility with RDF standards and its capacity to handle heterogeneous data sources. The technical pipeline corresponding to the conceptual phases introduced in Chapter 3 is illustrated in Figure 4, including the used packages and pyhton libaries. In the first phase, SysML XML files are parsed and transformed into RDF graphs via the defined YARRRML mappings. This phase results in a semantically annotated KG representation of each SysML model, capturing entities, relationships, and structural attributes required for subsequent analysis. In the second stage, entity alignment is performed,

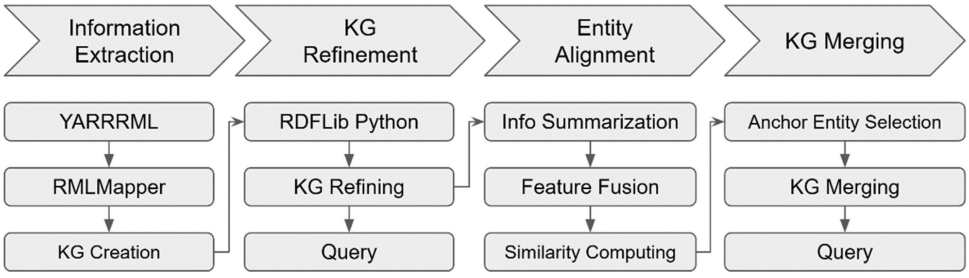


Figure 4. Pipeline for SysML model integration

https://doi.org/10.1017/pds.2026.10645 Published online by Cambridge University Press

further establishing the EasyEA approach. While other approaches for semantic comparison, such as wpath exists (Zhu & Iglesias, 2017), EasyEA was chosen because it enables the application with the support of LLMs even for small data sets.

In this application, ChatGPT-5.1 was employed to enrich node representations by synthesizing contextual information derived from element names, requirements, functions, and relational context. These enriched representations are transformed into vector embeddings, and cosine similarity scores are calculated to quantify semantic proximity between entity pairs. The third stage applies similarity-based decision thresholds to determine integration actions. For this implementation, cosine similarity values were determined based on a separate dataset comprising subsystems of a lawnmower. After manually verifying the correctness of the results, the values were set for the final dataset as follows. A cosine similarity value of 0.87 and above was considered an indication of strong semantic equivalence and allowed for automatic merging. Similarity values between 0.87 and 0.5 were classified as ambiguous and were subject to manual validation by the researcher. Values below 0.5 were interpreted as semantically divergent and were consequently not merged. These thresholds were again tested and reflect a trade-off between integration precision and risk of incorrect fusion. The final stage consists of consolidating aligned entities into a unified knowledge graph. This process eliminates redundant nodes while preserving all unique structural and relational information. The resulting integrated KG serves as the semantic representation of the combined system architecture and forms the basis for transformation back into SysML format.

4.2. Case study description

The implementation was evaluated using two simplified SysML models consisting of a reduced number of elements and connections representing subsystems of a vehicle drive system: a brake system model and a steering system model. Both models contained overlapping functional and structural elements, including requirements and sensor-related components, but exhibited divergent naming conventions and structural representations. The brake system model included subsystems such as a "Sensor system" and a torque sensor, as well as a brake wear sensor. The steering system model contained a subsystem labelled *Sensors* comprising a torque sensor and a steering wheel sensor. The objective of the case study was to assess whether the implementation could correctly identify and semantically align equivalent elements despite inconsistent naming and structural variation.

4.3. Integration results and interpretation

The components in the integrated KG are shown in Figure 5. The implementation successfully identified semantically equivalent elements across both subsystem models and consolidated them into unified nodes where appropriate. The requirement *recuperation* was detected in both models despite differences in naming and descriptive context and was merged into a single requirement node. Similarly, the elements *sensor system* and *sensors* were aligned and consolidated, reflecting their functional equivalence at the subsystem level. At the component level, the elements *torque sensor* and *torque sensors* were also correctly identified as semantically equivalent and merged. In contrast, elements such as the *brake wear sensor* and the *steering wheel sensor*, which represented distinct functional roles, were retained as separate nodes, preserving necessary system differentiation.

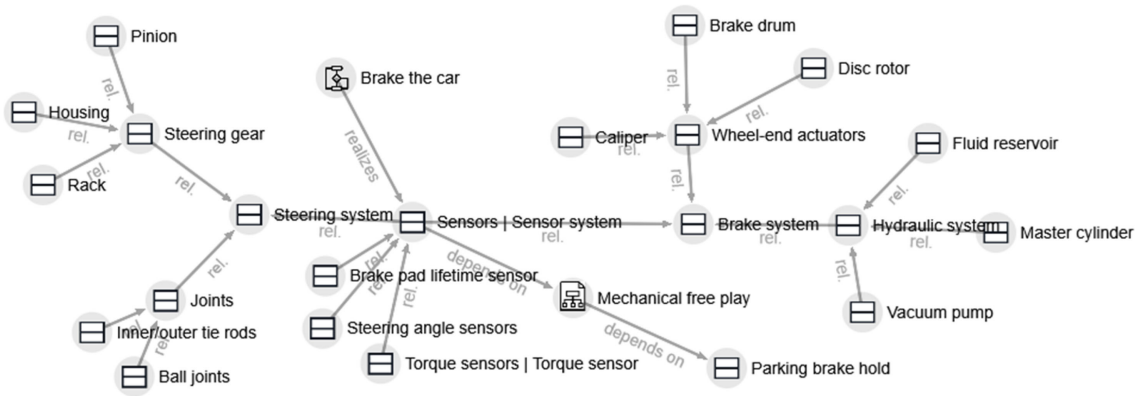


Figure 5. Integrated knowledge graph after semantic fusion

https://doi.org/10.1017/pds.2026.10645 Published online by Cambridge University Press

These results demonstrate the ability to distinguish between semantically equivalent and divergent elements, thereby supporting both automated integration and controlled differentiation. While certain decisions required manual validation in ambiguous cases, the overall pipeline illustrates the viability of the proposed approach for supporting RQ1 by enabling the systematic integration of multiple SysML models and RQ2 by providing differentiated integration strategies based on semantic similarity. Nevertheless, the case study also reveals limitations. The reliance on manually validated thresholds and the dependency on modelling quality indicate the need for further refinement and scale testing.

5. Conclusions, limitations, and future research

This work set out to support the integration of heterogeneous SysML models in the context of MBSE by introducing a semantically grounded approach based on KG, entity alignment, and semantic fusion. The primary objective was to facilitate the consolidation of independently developed SysML models and to provide structured alternatives for action based on detected differences between model elements. With regard to *RQ1*, this research demonstrated that SysML model integration can be systematically enabled through a KG-based workflow that transforms model elements into semantically enriched representations and applies entity alignment mechanisms to identify equivalent elements across models. The implementation showed that semantically corresponding elements, despite heterogeneous naming and modelling practices, can be reliably identified and consolidated into a unified architectural representation. Considering *RQ2,* the proposed approach introduced a differentiated decision logic based on similarity thresholds. This enabled the distinction between automatic fusion, manual validation, and preservation of separate elements, thereby offering structured integration strategies tailored to the degree of semantic divergence. The central scientific contribution of this work lies in the development and demonstration of a reusable conceptual and technical approach that bridges the gap identified in the state of the art between syntactic integration mechanisms and the need for semantically robust model integration in realistic MBSE environments. By integrating KG technologies and LLM-based entity alignment, this work advances current practices and provides a foundation for scalable and traceable SysML model integration.

Despite demonstrating feasibility, several limitations constrain the current contribution. Methodologically, the approach focuses primarily on semantic comparison and does not yet address structural consistency checking, behavioural compatibility, or interface integrity, which remain critical aspects of comprehensive system integration. Reliance on empirically determined cosine similarity thresholds introduces uncertainty about their optimality and transferability to other datasets or industrial contexts. From a technical perspective, the implementation is limited by the extent of information that can be extracted from XML-based SysML representations, thereby restricting coverage of the full SysML language spectrum. Additionally, the dataset used in the case study reflects a simplified subsystem scenario and does not capture the complexity, hierarchical depth, and variability typical of large-scale industrial models. As a result, conclusions regarding scalability and generalisability must be interpreted with caution. Furthermore, the integration process partially depends on manual validation in ambiguous cases, which limits automation and introduces subjectivity. Compared to existing approaches, the presented method provides greater semantic flexibility but lacks the automated consistency checks and large-scale validation seen in more mature integration frameworks.

Future research should focus on formalising the proposed approach as a fully defined method, including a systematic specification of requirements, functions, and evaluation criteria. A key direction involves the extension of the concept to incorporate structural and behavioural consistency analysis, such as interface compatibility and dependency verification, thereby addressing the full spectrum of MBSE integration challenges highlighted in Chapter 1. Moreover, future work should aim at automating threshold calibration through adaptive or learning-based mechanisms to reduce reliance on empirically defined parameters. Quantitative evaluation metrics, such as precision, recall, and alignment accuracy, should be introduced to enable objective assessment and benchmarking against existing integration approaches. Additional research opportunities include the integration of domain ontologies for improved semantic grounding, the exploration of fully automated fusion strategies using LLM-driven reasoning, and the validation of the approach within real industrial MBSE workflows. These developments will enhance both the practical applicability and the scientific maturity of semantically supported SysML model integration.

Acknowledgement

Funded by zukunft.niedersachsen, the joint science funding program of the Lower Saxony Ministry of Science and Culture and the Volkswagen Foundation, and supported by the Center for Digital Innovations (ZDIN).

https://doi.org/10.1017/pds.2026.10645 Published online by Cambridge University Press

References

Berschik, M. C., Schumacher, T., Laukotka, F. N., Krause, D., & Inkermann, D. (2023). MBSE WITHIN THE ENGINEERING DESIGN COMMUNITY – AN EXPLORATORY STUDY. *Proceedings of the Design Society*, 3, 2595–2604. https://doi.org/10.1017/pds.2023.260

Besta, M., Gerstenberger, R., Peter, E., Fischer, M., Podstawski, M., Barthels, C., Alonso, G., & Hoefler, T. (2024). Demystifying Graph Databases: Analysis and Taxonomy of Data Organization, System Designs, and Graph Queries. *ACM Computing Surveys*, 56(2), 1–40. https://doi.org/10.1145/3604932

Cao, Y., Liu, Y., & Paredis, C. J. J. (2011). System-level model integration of design and simulation for mechatronic systems based on SysML. *Mechatronics*, 21(6), 1063–1075. https://doi.org/10.1016/j.mechatronics.2011.05.003

Cheng, J., Lu, C., Yang, L., Chen, G., & Zhang, F. (2025). EasyEA: Large Language Model is All You Need in Entity Alignment Between Knowledge Graphs. In W. Che, J. Nabende, E. Shutova, & M. T. Pilehvar (Eds.), *Findings of the Association for Computational Linguistics: ACL 2025* (pp. 20981–20995). https://doi.org/10.18653/v1/2025.findings-acl.1080

Färber, M., Bartscherer, F., Menne, C., & Rettinger, A. (2017). Linked data quality of DBpedia, Freebase, OpenCyc, Wikidata, and YAGO. *Semantic Web*, 9(1), 77–129. https://doi.org/10.3233/SW-170275

Fosse, E., & Delp, C. L. (2013). Systems engineering interfaces: A model based approach. 2013 IEEE Aerospace Conference, 1–8. https://doi.org/10.1109/AERO.2013.6497322

Friedenthal, S., Moore, A., & Steiner, R. (2015). *A Practical Guide to SysML*. Elsevier. https://linkinghub.elsevier.com/retrieve/pii/C20130144571

Hogan, A., Blomqvist, E., Cochez, M., D'amato, C., Melo, G. D., Gutierrez, C., Kirrane, S., Gayo, J. E. L., Navigli, R., Neumaier, S., Ngomo, A.-C. N., Polleres, A., Rashid, S. M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., & Zimmermann, A. (2022). Knowledge Graphs. *ACM Computing Surveys*, 54(4), 1–37. https://doi.org/10.1145/3447772

Hu, Y., Peng, Q., Ni, Q., Wu, X., & Ye, D. (2023). Event-based safety and reliability analysis integration in model-based space mission design. *Reliability Engineering & System Safety*, 229, 108866. https://doi.org/10.1016/j.ress.2022.108866

Inkermann, D., & Ammersdörfer, T. (2026). Risk-Aware System Planning in Offer Phases based on Integrated Variation Analysis and Risk Assessment. Procedia CIRP, Accepted for Publication.

Jiang, X., Shen, Y., Shi, Z., Xu, C., Li, W., Li, Z., Guo, J., Shen, H., & Wang, Y. (2024). Unlocking the Power of Large Language Models for Entity Alignment (arXiv:2402.15048). arXiv. https://doi.org/10.48550/arXiv.2402.15048

Kapos, G.-D., Tsadimas, A., Kotronis, C., Dalakas, V., Nikolaidou, M., & Anagnostopoulos, D. (2021). A Declarative Approach for Transforming SysML Models to Executable Simulation Models. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 51(6), 3330–3345. https://doi.org/10.1109/TSMC.2019.2922153

Krog, J., Akbas, C., Nolte, B., & Vietor, T. (2025). Development of a Functional and Logical Reference System Architecture in Automotive Engineering. *Systems*, 13(3), 141. https://doi.org/10.3390/systems13030141

Lange, C., Grundmann, J. T., Kretzenbacher, M., & Fischer, P. M. (2018). Systematic reuse and platforming: Application examples for enhancing reuse with model-based systems engineering methods in space systems development. *Concurrent Engineering*, 26(1), 77–92. https://doi.org/10.1177/1063293X17736358

Li, Q., Guo, S., Luo, Y., Ji, C., Wang, L., Sheng, J., & Li, J. (2023). Attribute-Consistent Knowledge Graph Representation Learning for Multi-Modal Entity Alignment. Proceedings of the ACM Web Conference 2023, 2499–2508. https://doi.org/10.1145/3543507.3583328

Li, Z., Husung, S., & Wang, H. (2025). LLM-Assisted Semantic Alignment and Integration in Collaborative Model-Based Systems Engineering Using SysML v2. 2025 IEEE International Symposium on Systems Engineering (ISSE), 1–8. https://doi.org/10.1109/ISSE65546.2025.11369983

Lin, Q., Mao, R., Liu, J., Xu, F., & Cambria, E. (2023). Fusing topology contexts and logical rules in language models for knowledge graph completion. *Information Fusion*, 90, 253–264. https://doi.org/10.1016/j.inffus.2022.09.020

Liu, X., Hong, H., Wang, X., Chen, Z., Kharlamov, E., Dong, Y., & Tang, J. (2022). SelfKG: Self-Supervised Entity Alignment in Knowledge Graphs. Proceedings of the ACM Web Conference 2022, 860–870. https://doi.org/10.1145/3485447.3511945

Pan, S., Luo, L., Wang, Y., Chen, C., Wang, J., & Wu, X. (2024). Unifying Large Language Models and Knowledge Graphs: A Roadmap. *IEEE Transactions on Knowledge and Data Engineering*, 36(7), 3580–3599. https://doi.org/10.1109/TKDE.2024.3352100

Schumacher, T., & Inkermann, D. (2023). Model inconsistencies and solution approaches to maintain consistency in model-based systems engineering. emphThe Eighteenth International Conference on Systems.

Schumacher, T., Müller, C.-K., & Inkermann, D. (2025). SysML process chains in MBSE: Systematic literature review and future research directions. *Digital Engineering*, 4, 100037. https://doi.org/10.1016/j.dte.2025.100037

https://doi.org/10.1017/pds.2026.10645 Published online by Cambridge University Press

Sivaloganathan, S., & Shahin, T. M. M. (1999). Design reuse: An overview. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 213(7), 641–654. https://doi.org/10.1243/0954405991517092

Slobin, S., Yoon, Z., & Fugger, S. (2024). Applying MBSE to Optimize Satellite and Payload Interfaces in Early Mission Phases. *Systems*, 12(8), 310. https://doi.org/10.3390/systems12080310

Spütz, K., Jacobs, G., Konrad, C., Wieja, F., & Stürmer, S. (2023). Knowledge-supported Development of Function Models for Mechanical Products Using Intelligent Function Libraries. *Procedia CIRP*, 119, 558–563. https://doi.org/10.1016/j.procir.2023.02.149

Trujillo, A., De Weck, O. L., & Madni, A. M. (2020, November 16). An MBSE Approach Supporting Technical Inheritance and Design Reuse Decisions. ASCEND 2020. https://doi.org/10.2514/6.2020-4167

Walden, D. D. (Ed.). (2023). *Systems engineering handbook: A guide for system life cycle processes and activities (Fifth edition)*. Wiley.

Zhu, G., & Iglesias, C. A. (2017). Computing Semantic Similarity of Concepts in Knowledge Graphs. *IEEE Transactions on Knowledge and Data Engineering*, 29(1), 72–85. https://doi.org/10.1109/TKDE.2016.2610428

https://doi.org/10.1017/pds.2026.10645 Published online by Cambridge University Press