INTERNATIONAL DESIGN CONFERENCE – DESIGN 2024
https://doi.org/10.1017/pds.2024.265

AI-based analysis and linking of technical and organisational data using graph models as a basis for decision-making in systems engineering

Sebastian Katzung [1], Hüseyin Cinkaya [1], Umut Volkan Kizgin [2,✉], Alexander Savinov [3], Julian Baschin [2] and Thomas Vietor [2]

[1] ENBACE GmbH, Germany, [2] Technische Universität Braunschweig, Germany, [3] Siemens Mobility GmbH, Germany

✉ u.kizgin@tu-braunschweig.de

Abstract

The increased complexity of development projects surpass the capabilities of existing methods. While Model Based Systems Engineering pursues technically holistic approaches to realize complex products, aspects of organization as well as risk management, are still considered separately. The identification and management of risks are crucial in order to take suitable measures to minimize adverse effects on the project or the organization. To counter this, a new graph-based method and tool using AI, tailored to the needs of complex development projects and organizations is introduced here.

Keywords: risk management, model-based systems engineering (MBSE), graph theory, artificial intelligence (AI), product lifecycle management (PLM)

1. Introduction

1.1. Motivation

Modern product development processes and products are confronted with unprecedented complexity. The demands for innovation, efficiency and quality are leading to the creation of highly complex systems that are designed to work across divisions and departments. Therefore, today's products must be understood as a comprehensive system that is developed from various technical areas such as mechanics, electronics and software. (Huth and Vietor, 2020). This change is creating a heterogeneous data landscape in which information from different sources and formats converge. However, this highlights a crucial problem: the inconsistent linking and transfer of data between these areas leads to a lack of traceability and transparency.

In order to manage complexity, ensure traceability and identify risks in system and software development, various systems engineering (SE) and model-based systems engineering (MBSE) approaches and tools are already in use. However, although these approaches have been established in various areas for many years, users still face considerable challenges. In systems engineering, isolated approaches have proven to be suboptimal, as the dependencies between technical processes and organizational (project and enterprise) processes have not been sufficiently taken into account. This leads to a lack of traceability of information from the entire product life cycle and makes it difficult to identify risks. This means that risk identification and risk management in highly complex projects are often limited by methods that are not sufficiently designed to recognize and understand the complex interrelationships between the various project areas. There is therefore a need for a new holistic integrative process model / project management approach in the sense of systems engineering to manage

https://doi.org/10.1017/pds.2024.265 Published online by Cambridge University Press

complexity ensure traceability along the product life cycle and facilitate risk management for agile system and software development.
In terms of model-based systems engineering, models are used to ensure traceability and increase the effectiveness of development processes. However, existing MBSE approaches and tools primarily cover the technical process as defined by ISO 15288. Often, the necessary linkage for a holistic view of enterprise and project processes within the context of MBSE is missing. This results in a fragmented perspective on complex systems and a lack of integration of the various dimensions crucial for efficient product development and design. The successful implementation of MBSE lies in the integration of data and models throughout the product's entire lifecycle. Furthermore, the implementation and use of existing MBSE tools often requires a steep learning curve and in-depth knowledge of complex modeling languages such as SysML. This transition also requires a cultural change as traditional document-based approaches are replaced by model-based methods, which can lead to resistance and acceptance issues within the organization. In addition, training employees in the use of specific modeling tools and techniques is time-consuming and costly.

1.2. Aim of the paper

In this context, it is crucial for product development teams to be able to address not only the technical aspects but also the organizational and process-related challenges to ensure effective and efficient development of highly complex systems. Graph-based/model-based systems engineering with the use of artificial intelligence can serve as a fundamental building block for success in this dynamic environment. In this paper, which resulted from the publicly funded project RePASE (Reflexive Prozessentwicklung und -adoption im Advanced Systems Engineering), we present a new integrative process model and a graph-based method for software and systems engineering developed in response to the SE and MBSE challenges mentioned above. Our approach aims to manage the complexity of products and processes and ensure efficient communication among stakeholders in product development by harmonizing the heterogeneous data landscape, establishing a foundation for data path analysis, preventing inconsistent data transmissions, and enhancing traceability, through artifical intelligence-based analysis and linking. Moreover, the approach provides an intuitive application to overcome the initial hurdle on the path to MBSE and increase acceptance without requiring elaborate modeling languages.

2. State of the art

2.1. Systems engineering and its model based approaches

Over the years, various approaches have been developed to address the increasing complexity of systems. Systems Engineering has established itself as an interdisciplinary approach in industries such as defense and aerospace for designing, developing, and managing complex systems throughout their entire lifecycle. It is a systematic and structured methodology aimed at understanding and implementing the requirements, functions, and performance of a system within a holistic context. SE applies to a wide range of systems, including technical systems as well as non-technical systems like organizations and processes. (INCOSE Technical Operations, 2007)
The SE processes are standardized in ISO 15288. This norm deals with the objective, assessment, and improvement of processes. It provides a structured framework that covers the entire lifecycle of a system. Within this framework, processes are defined to support the development and management of systems. These processes include technical processes like requirement definition, system design, integration, verification, validation, and system commissioning. Besides technical processes, the standard encompasses project processes, agreement processes, and enterprise processes (ISO, 2023).
The current "state of the art" for SE is heavily influenced by model-based approaches. These approaches utilize models as central elements for defining, designing, and managing complex systems. Models offer an efficient way to visualize, organize, and communicate information. In this context, MBSE is an interdisciplinary approach based on the use of models to describe, analyze, and optimize the entire

https://doi.org/10.1017/pds.2024.265 Published online by Cambridge University Press

lifecycle of a system. It allows for a holistic view and integration of all relevant aspects, from initial requirements to validation and commissioning. (INCOSE Technical Operations, 2007)

In the context of MBSE, various modeling languages and tools are used to handle the complexity of systems. These include, for example, the Unified Modeling Language (UML) and the Systems Modeling Language (SysML). These languages provide standardized symbols and concepts that contribute to creating coherent and consistent models. Besides the modeling language and tool, the choice of a suitable modeling approach and its consistency with other modeling components are of great importance for the successful application of model-based Systems Engineering (Delligatti, 2017). A detailed overview of MBSE approaches such as OOSEM, SYSMOD and OPM can be found in (Estefan, 2008).

2.2. Process models in system and software engineering

In today's engineering applications, there is a strong interplay among mechanics, electrical engineering and software engineering. Originating from software engineering, the V-model outlines a systematic approach to such a development. In accordance with this goal, VDI/VDE 2206 has been specially adapted for the field of mechatronic and cyber-physical systems in order to do justice to the special features of this interdisciplinary field. The V-model organizes system development into phases, starting with requirements analysis and system design on the left side and mirroring testing and validation phases on the right. It illustrates the relationship between each development phase and its corresponding testing phase, ensuring that the system meets the specified requirements. (VDI, 2021) As already mentioned, the principles of system and software engineering are standardised in ISO 15288, which serves as an extension of ISO 12207, which specifically provides a framework for processes in the life cycle of software (ISO, 2017). Software project management and development methodologies come in a diverse array, each offering its own approach and benefits. For instance, the Waterfall model follows a sequential design process, the Spiral model emphasizes iteration and risk management, while methodologies like Scrum and DevOps focus on agility and collaboration throughout the development lifecycle. These models and methodologies offer different frameworks to cater to the distinct needs and objectives of various projects. (Haraty and Hu, 2018)

Today, continuous software development is used to quickly deliver high-quality software by regularly integrating changes, reducing risks and promoting adaptability to market requirements. It aims to improve collaboration, reduce time to market and increase customer satisfaction through iterative, frequent updates and improvements. It is a methodology that emphasizes an ongoing and iterative approach to software development, where changes and updates are regularly integrated and delivered. It involves a constant cycle of planning, development, testing, deployment, and monitoring, often in small, incremental steps. This methodology aims to improve the software development process by reducing the time between writing new code and deploying it to production, enabling faster response to change and more frequent updates based on user feedback. Continuous integration, continuous delivery (CI/CD), and agile development methodologies are often part of continuous software development, enabling teams to deliver high-quality software more frequently. (Bosch, 2014)

2.3. Application of graph theory and AI in systems engineering

The literature features initial proposals for integrating graph theory with systems engineering. Peterson (2015) advocates a symbiotic relationship between graph theory and MBSE to enhance the understanding of complex systems and expedite the innovation process. He underlines the capability of graph theory to provide dynamic visualizations and asserts its relevance in conceptualizing, analyzing, and integrating intricate systems. Peterson (2015) emphasizes derivatives of graph theory, such as Network Analysis and Design Structure Matrix (DSM), for their pivotal role in MBSE applications, enabling the modeling and comprehension of interactions among system entities.

Further reinforcing this viewpoint, Bajaj et al. (2020) and Filomonov et al. (2016) echo the significance of organizing product or system information into network structures, represented as mathematical graphs that center on entities and their relationships. Filomonov et al. (2020) furthers this perspective by developing a graph-based modeling approach, aiming to simplify the inherent complexities in Model-Based Systems Engineering. The graphical representations introduced by these models provide a more

https://doi.org/10.1017/pds.2024.265 Published online by Cambridge University Press

effective means of visualizing and managing complex systems, particularly beneficial for those not well-versed in specific formal languages or conventional text-based description methods.

Collectively, these findings underline the vital role of graph theory in system engineering, particularly in facilitating dynamic visualizations and enabling a comprehensive understanding of intricate system architectures. The integration of graph theory into the domain of Model-Based Systems Engineering not only enriches visualization and understanding but also significantly accelerates the innovation process for development teams. For a deeper exploration of various approaches and aspects of graph theory's application in product development within the context of systems engineering, additional insights are available in (Chucholowski and Lindemann, 2015), (Harrison, 2016) and (Peterson, 2016).

In addition to proposals for the integration of graph theory, the integration of artificial intelligence into MBSE has gained significant attention in recent research. Charfi et al. (2023) investigate how AI can enhance MBSE tools. They propose that integrating AI techniques such as Machine Learning and Natural Language Processing (NLP) into MBSE tools is essential for effectively fulfilling stakeholder requirements. Additionally, knowledge representation using ontologies and semantic web technologies is discussed to enable AI for Systems Engineering (Hagedorn et al., 2020). Furthermore, the adoption of AI for MBSE involves researching data extraction and data classification topics from PDF files to transfer it to system models elements (Abdoun, 2022). The combined use of MBSE and AI is emphasized to support the handling of the increasing complexity and the requirements for sustainability of systems in the early phase of product development (Schneider et al., 2022).

3. Contribution of the paper

The development of highly complex products necessitates a comprehensive approach that encompasses both technical and organisational aspects. As mentioned before, isolated SE-perspectives often yield suboptimal results as they fail to account for the interdependencies among various system components and organisational processes. The technical aspects involve the use of MBSE to improve product specification, analysis, verification and validation through comprehensive models. These models help to better understand and control the behaviour of complex systems. On the other hand, organisational aspects relate to project management and the structure required for effective project implementation. Aspects such as team structure, communication channels and decision-making processes should be organised in such a way that they support the technical requirements and enable smooth implementation. Team collaboration plays a central role and fosters a culture of knowledge sharing that accelerates development cycles and increases product quality. A holistic view takes into account the interactions between technical systems and organisational framework conditions in order to identify risks at an early stage, improve decision-making processes and promote agility in product development. Within this context, (Project Management Institute, 2017) emphasizes the significance of an integrated project management approach that combines technical and organizational elements for the success of complex projects.

In addition to these aspects, risk management is particularly important in the development of highly complex products, as it aims to recognise potential problems and take countermeasures to avoid delays and quality defects. However, existing methods are often unable to capture the complex interactions between different project areas. Traditional approaches focus on linear cause-and-effect relationships. Nevertheless, in a complex project environment, risks are often multidimensional and can span different areas, meaning that a risk in one area can affect another area in unpredictable ways. For example, a delay in the development of a software module could not only affect the software release schedule, but also impact hardware testing, marketing strategies and even legal compliance issues. This kind of cross-domain causality and correlation is often overlooked by traditional risk management methods that tend to work in silos. The literature acknowledges these challenges, as highlighted by Sargut and McGrath (2017), who emphasize that complexity often leads to "causal ambiguities," where the sources of risks are hard to identify, and the impacts are difficult to predict. To address the identified deficiencies in the development of complex products and meet the demand for a holistic perspective, an innovative approach is needed, one that transcends the boundaries of traditional system development. Therefore, the research work aims to answer these research questions:

https://doi.org/10.1017/pds.2024.265 Published online by Cambridge University Press

RQ1: How can an integrative lifecycle process for system development be conceptualized that both enables risk reduction through its focus on validation and verification at different stages of the development process and ensures agility and flexibility through continuous integration and deployment?
RQ2: How can an AI-based method and tool help to visualize and identify causal relationships and correlations of development information along the system lifecycle for risk management purposes?

4. Concept for holistic linking of design data

4.1. The symbiosis of structure and agility

The V-Model and Continuous Software Development (CSD) represent two ends of the development spectrum (Figure 1) that are increasingly being integrated into a combined approach to address the complexity of modern systems. The V-Model, as a traditional, phase-oriented framework, provides a structure that ensures the quality and safety of complex products through clear validation and verification processes. On the other hand, CSD stands for flexibility and speed, relying on continuous integration, testing, and deployment. In this hybrid approach, the V-Model plays a crucial role by providing a solid foundation for the Continuous Lifecycle and DevOps approaches. By linking the rigorous testing and integration phases of the V-Model with the agile practices of CSD, a balance between rapid market entry and reliable system quality is achieved. In a DevOps-oriented environment where development and operations are closely intertwined, the V-Model ensures that even with a high frequency of updates and new features, consistent quality and system stability are not neglected.

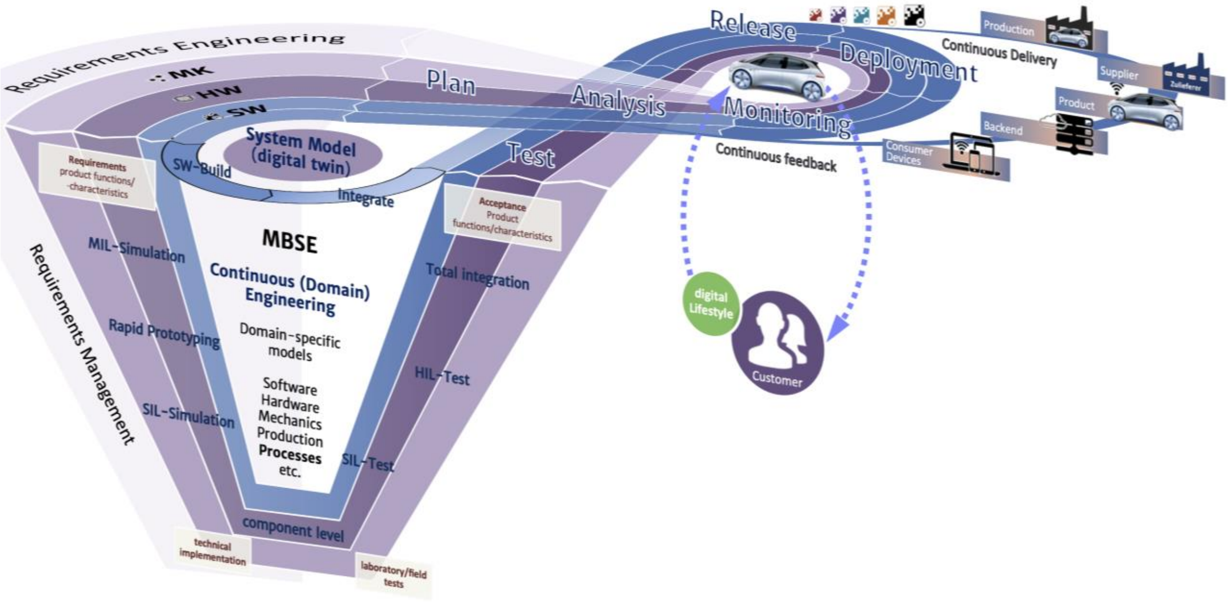


Figure 1. Relationship between the V-Model and Continuous Software Development (CSD)

Graph models serve as bridging tools within this integrated approach to visualize and understand the dynamic interactions between different development domains and gain transcendent insights into the "product-project-organization" system. They enable the capture and analysis of complex dependencies and interactions between the structured V-Model and adaptive Continuous Software Development. This holistic approach ensures that teams can respond to both technical requirements and changing market demands and internal organizational dynamics. By combining the V-Model and CSD, companies can align the necessary structure and agility to remain competitive in the fast-paced world of technology.

4.2. Concept of the integrated approach

In the development of highly complex systems, a profound understanding of the intricate interdependencies between different domains is essential. An innovative approach that promotes this

https://doi.org/10.1017/pds.2024.265 Published online by Cambridge University Press

understanding is the application of integrated graph models, which not only represent the complexity of these relationships but also make them analysable.

In the first step, the identification and definition of core areas, such as development, production, and operation, which constitute the "product-project-organization" system, is carried out. Specific graph models are developed for each of these domains, representing their internal structures and processes and allowing for a detailed examination of domain-specific causality.

The fusion of these domain-specific models into a holistic overall graph enables the understanding and analysis of complex cross-domain correlations and interactions (Forsgren et al., 2018). The use of advanced analysis tools from machine learning and network analysis supports the discovery of hidden patterns and the anticipation of potential risks that could arise from the interaction of individual domains. Such an integrated graph model approach is not static but is dynamically adapted to changing conditions, making it compatible with the agile practices of Continuous Software Development and the seamless processes of DevOps. This approach combines the precise testing and verification processes of the V-Model with the adaptability and speed of continuous development methods to ensure a balance between structure and flexibility (Boehm, 2006; Forsgren et al., 2018).

Through the use of feedback loops, a continuous learning process is established, allowing all stakeholders to feed their insights directly into the development process and contribute to the continuous optimization of the system. In conclusion, this integrated graph model approach realizes the synergy of the strengths of the V-Model with the agility of Continuous Software Development and DevOps, making a significant contribution to the development and management of highly complex systems.

4.3. Structured construction of graph models

The integration and linkage of heterogeneous data sources into a coherent graph present a significant challenge. In today's data-driven world, this ability is crucial for gaining deeper insights and supporting informed decision-making. Graph technologies provide a flexible framework for the integration and analysis of various data types, which is particularly valuable in complex environments such as system development of the "product-project-organization**."** The primary goal is to unify and link heterogeneous data in a coherent graph that allows for deeper data analysis in various contexts.

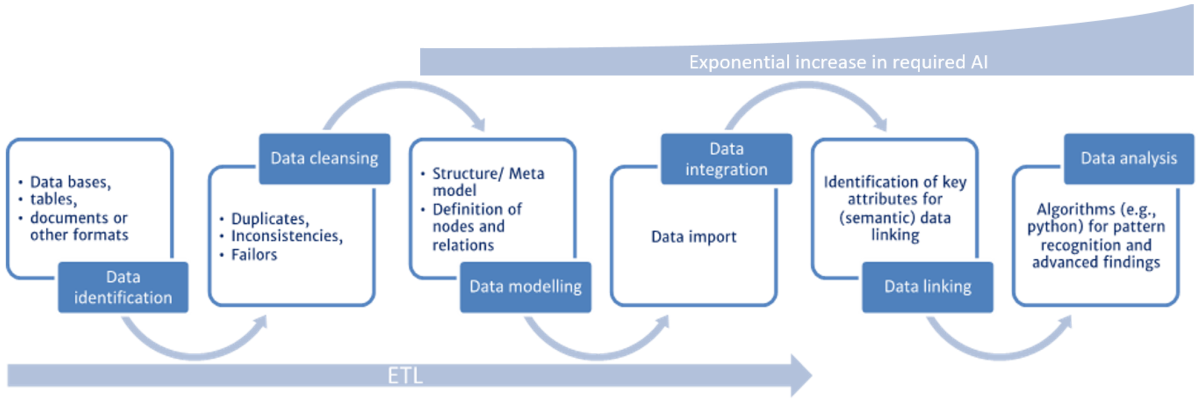


Figure 2. Structured approach to linking heterogeneous data into graph structures

The process begins with data identification (Figure 2), where all relevant data sources are collected. In the next step, data cleansing, duplicates and inconsistencies are removed to ensure data quality. Data modeling follows, including the definition of nodes (entities) and edges (relationships) that reflect the structure and relationships of the data. Data integration involves processes such as ETL (Extract, Transform, Load) or specialized graph database tools to transform the data into the graph. Data linking is done through the identification of common keys or attributes, supported by techniques like data matching and semantic linking. Finally, data analysis using graph analysis algorithms allows the extraction of patterns and insights from the data. It is important that the graph is regularly updated to incorporate new data and reflect changes in existing data.

https://doi.org/10.1017/pds.2024.265 Published online by Cambridge University Press

4.4. Transcendent insights into "product-project-organization"

Gaining transcendent insights into the "product-project-organization" system requires a comprehensive view of interactions and relationships that go beyond the boundaries of conventional analysis. In practice, this means going beyond domain-specific insights and gaining a deep understanding of the dynamic and often hidden influences that shape the entire system.

The application of graph models enables the visualization and analysis of complex relationships between technical processes, organizational structures, and project management methods. By mapping the internal and external links within the system, statements can be made about the impact of changes in one part of the system on other parts. This holistic view helps understand cross-domain causality and correlations that affect the end product, project progress, and organizational performance.

The use of advanced analysis techniques, such as machine learning and network analysis, within the framework of graph models facilitates the identification of patterns and trends that would remain hidden in an isolated view, since a graph-based approach also simplifies and accelerates the process of recognising design patterns (Mayvan and Rasoolzadegan, 2017). For example, it can uncover risk factors that affect multiple domains and enable the development of proactive risk mitigation strategies.

The integration of concepts such as the V-Model and Continuous Software Development into this system further ensures that product development not only meets technical requirements but is also agile enough to respond to changing market demands and internal organizational dynamics. This leads to increased resilience and adaptability of the overall system.

Transcendent insights in this context mean grasping the "big picture" and making decisions based on a deep understanding of all factors. This enables better anticipation of future developments and decision-making based on a solid data foundation, significantly increasing the likelihood of success in complex product development projects. The next section introduces a specific tool for creating graph models, translating theoretical concepts into tangible results.

5. Case study

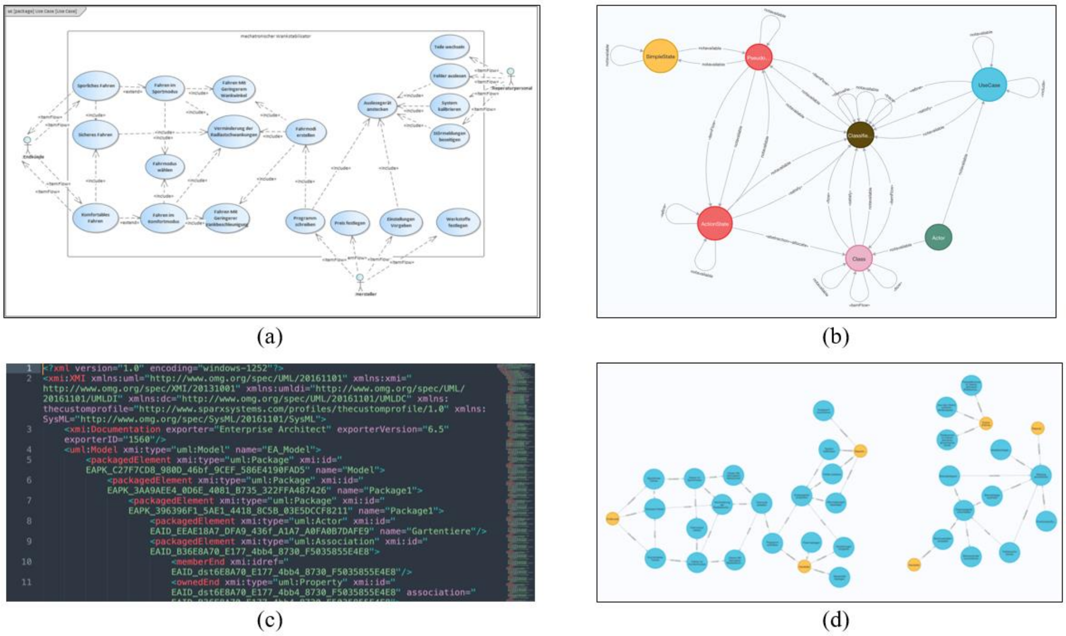
The tool "Grapholi" was developed by the project partner ENBACE GmbH. The tool and the method were tested in a large enterprise that develops automated train control systems. This section will demonstrate how the tool connects different information from different domains in this development context to visualize and analyse the dynamic relationships and dependencies. A unique feature of the tool is its ability to identify relations between requirements and test cases, as well as map and analyze processes and organizational structures. These case studies aim to demonstrate the strengths of the tool and the challenges encountered and provide a comprehensive understanding of its practical potential and limitations in real-world scenarios.

5.1. Empirical testing and application contexts

The tool is implemented as a user-centric web application, allowing interaction with a multi-dimensional analysis tool through standardized devices. Built on the Quasar Framework, it facilitates communication with a Neo4j graph database via a REST API. The architecture is complemented by the integration of Python scripts for advanced analysis techniques, promoting data transfer with other systems, achieving exceptional flexibility, and scalability.

Two use cases were selected for empirical testing. The initial application situation addresses the processing and integration of unknown data from an already familiar system - specifically: system models from the modeling tool Enterprise Architect (EA) (Figure 3). In line with the methodology presented in section 4.3, the iterative process begins with data identification and raw data parsing (Figure 3c). The first steps of the methodology can be summarized here as a data preparation phase. Since we use artificial intelligence with an advanced pattern recognition algorithm, the original modeling language plays an insignificant role. During data integration, an ETL process is used to transfer the data into the graph. Once an ontology for semantic data linking has been adapted and updated as a metamodel, the data is analysed using graph analysis algorithms and patterns and insights are extracted from the data. This opens up new perspectives and insights into data interrelationships.

https://doi.org/10.1017/pds.2024.265 Published online by Cambridge University Press



```
<?xml version="1.0" encoding="windows-1252"?>
<xmi:XMI xmlns:uml="http://www.omg.org/spec/UML/20161101" xmlns:xmi="
http://www.omg.org/spec/XMI/20131001" xmlns:umldi="http://www.omg.org/spec/UML/
20161101/UMLDI" xmlns:dc="http://www.omg.org/spec/UML/20161101/UMLDC" xmlns:
thecustomprofile="http://www.sparxsystems.com/profiles/thecustomprofile/1.0" xmlns:
SysML="http://www.omg.org/spec/SysML/20161101/SysML">
    <xmi:Documentation exporter="Enterprise Architect" exporterVersion="6.5"
    exporterID="1560"/>
    <uml:Model xmi:type="uml:Model" name="EA_Model">
        <packagedElement xmi:type="uml:Package" xmi:id="
        EAPK_C27F7CD8_980D_46bf_9CEF_586E4190FAD5" name="Model">
            <packagedElement xmi:type="uml:Package" xmi:id="
            EAPK_3AA9AEE4_0D6E_4081_B735_322FFA487426" name="Package1">
                <packagedElement xmi:type="uml:Package" xmi:id="
                EAPK_396396F1_5AE1_4418_8C5B_03E5DCCF8211" name="Package1">
                    <packagedElement xmi:type="uml:Actor" xmi:id="
                    EAID_EEAE18A7_DFA9_436f_A1A7_A0FA0B7DAFE9" name="Gartentiere"/>
                    <packagedElement xmi:type="uml:Association" xmi:id="
                    EAID_B36E8A70_E177_4bb4_8730_F5035855E4E8">
                        <memberEnd xmi:idref="
                        EAID_dst6E8A70_E177_4bb4_8730_F5035855E4E8"/>
                        <ownedEnd xmi:type="uml:Property" xmi:id="
                        EAID_dst6E8A70_E177_4bb4_8730_F5035855E4E8" association="
```

Figure 3. (a) EA model; (b) Automatically derived metamodel of the input data; (c) XML excerpt from EA data stream; (d) Automatically reconstructed EA model

The secondary use case deals with the processing of known data structures of unknown origin in various formats. This involves the development information in documents such as the requirements specification, architecture specification and test case specification in pdf format. By analysing the data structures, the development artefacts (requirements, functions, test cases, ...) can be identified from the documents.

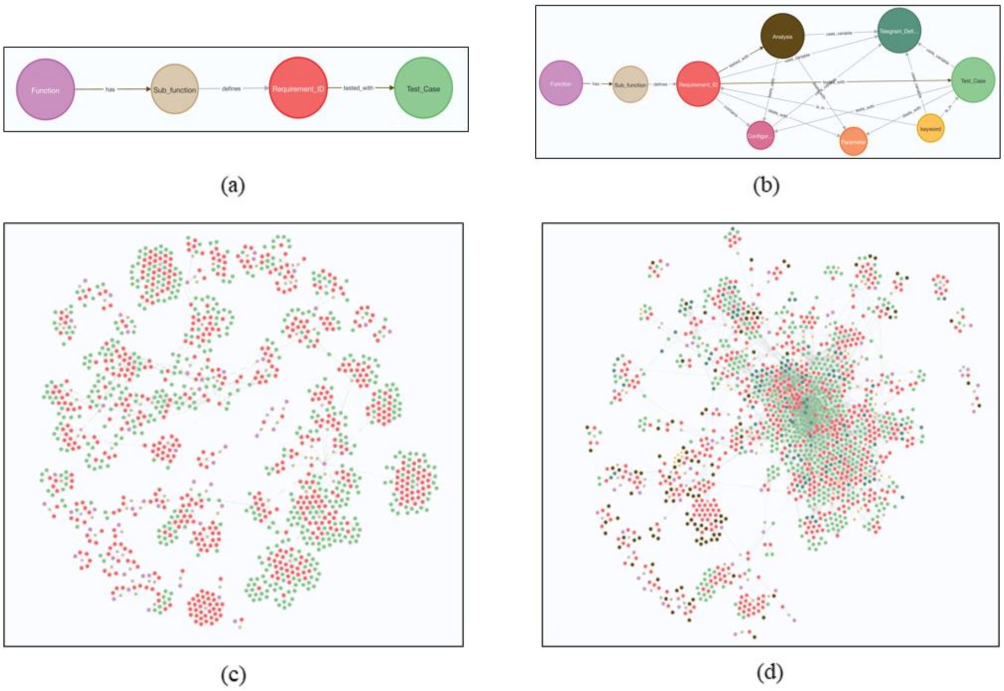


Figure 4. (a) Preliminary metamodel; (b) Expanded metamodel with analysis, configuration, parameter, keywords; (c) Test cases and requirements; (d) Representation of newly added dependencies

The creation of a temporary metamodel and an adaptive parser facilitates the association of the data and the updating of the ontology, which enables in-depth data analysis. In this way, new relationships between artifacts (e.g. test cases and requirements) can be identified through semi-semantic analysis

https://doi.org/10.1017/pds.2024.265 Published online by Cambridge University Press

based on the terms such as keywords, parameters and configurations that appear in the descriptions of these artifacts. Figure 4 illustrate the gradual expansion and refinement of the metamodel and the graph structure. They document the addition of new categories such as analysis, configuration and parameters as well as the integration of keywords. They also visualize the test cases, requirements and the newly identified dependencies.

5.2. Discussion of case studies and achieved insights

The implementation of the tool in real contexts, such as data analysis from Enterprise Architect, demonstrates the high performance of the tool in handling and transforming complex data structures into a comprehensive ontology. Furthermore, the tool reveals its efficiency in synthesizing partial metamodels and enhances understanding of previously hidden data relationships. Figure 5 highlights the added value of semantic analysis by revealing unrecognized connections between different systems and enabling configuration-specific requirement analyses.

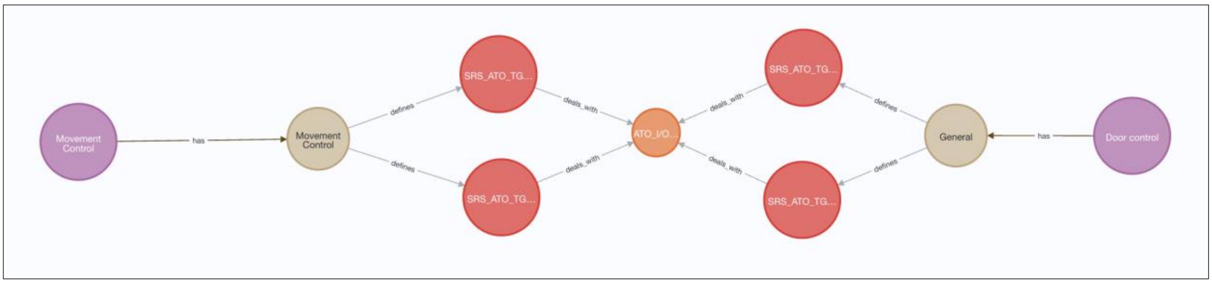


Figure 5. Linkage via a parameter

The synergetic linking of the use cases reveals the strengths of the methodology, in particular the efficient synthesis and accessibility of complex data structures. The integration of a user-oriented interface, which includes drag-and-drop functionality, symbolizes the user-friendliness and supports the seamless integration of unknown data into the analysis platform. The algorithms developed contribute significantly to the recognition of patterns and identities in the data, which promotes a detailed understanding of the system requirements and provides profound insights into the design and development process. The web application is establishing itself as an advanced tool for handling complex data structures. It not only enables visualization, but also provides essential insights for the design and optimization of process and organizational structures, making it an indispensable tool in the developer's repertoire.

6. Conclusion and outlook

In the current product development landscape, the increasing complexity of systems across different domains has led to challenges in data consistency and traceability. This paper introduces a new integrated holistic process model for the system and software development and a graph-based tool as a response to these challenges. On the one hand, the combination of the V-Model and CSD is advocated, emphasizing the importance of structure and agility in today's dynamic environment. On the other hand, the graph-based approach and tool aims to harmonize heterogeneous data, enhance traceability, and prevent inconsistent data transmissions. It provides an intuitive application to overcome initial hurdles in MBSE adoption without requiring expertise in complex modeling languages.

The tool was tested in a real railway industry system development project. It demonstrated the visualization of complex dependencies and relationships between development artifacts, aspects that are notably challenging to identify within documentation. Furthermore, it facilitated the exchange of information among stakeholders and engineering domains. It should be emphasised that the case study presented is only one possible use case. The approach could also be used for other purposes, e.g. for the identification of modules by grouping structural components according to the number of interfaces using the concept of a design structure matrix by graph theory. In addition, further research could develop the use of AI in MBSE to enable systems to automatically make predictions and adapt to changing requirements and constraints.

https://doi.org/10.1017/pds.2024.265 Published online by Cambridge University Press

Acknowledgement

“This research and development project is funded by the German Federal Ministry of Education and Research (BMBF) within the “Innovations for Tomorrow’s Production, Services, and Work” Program (funding number 02J19B140) and implemented by the Project Management Agency Karlsruhe (PTKA). The author is responsible for the content of this publication.”

References

Abdoun, N. and Chami, M. (2022). Automatic text classification of pdf documents using nlp techniques. Incose International Symposium, 32(1), 1320-1331. https://doi.org/10.1002/iis2.12997

Bajaj, M., Bckhaus, J., Waikar, M., Zwemer, D. Schreiber, C. (2017), Graph-Based Digital Blueprint for Model Based Engineering of Complex Systems, 27th Annual INCOSE International Symposium (IS 2017) Jul 15-20, 2017, Adelaide, Australia, https://doi.org/10.1002/j.2334-5837.2017.00351.x

Boehm, B. (2006), "Some Future Trends and Implications for Systems and Software Engineering Processes, Systems Engineering", Vol. 9, No. 1, Wiley InterScience, https://doi.org/10.1002/sys.20044

Bosch, J. (2014) Continuous Software Engineering: An Introduction. In: Bosch, J. (eds) Continuous Software Engineering. Springer, Cham. https://doi.org/10.1007/978-3-319-11283-1_1

Charfi, A., Kochbati, T. and Mraidha, C. (2023). From research to practice: does ai promote or prevent the use of an mbse tool?, https://doi.org/10.5121/csit.2023.131202

Chucholowski, N. and Lindemann, U. (2015), An Initial Metamodel to Evaluate Potentials for Graphbased Analyses of Product Development Projects, 7th International Dependency and Structure Modeling Conference, Texas, USA

Delligatti, L. (2013), SysML Distilled: A Brief Guide to the Systems Modeling Language. United States: U.S. Corporate and government Sales, 2013.

Estefan, J. A. (2008), Survey of model-based systems engineering (MBSE) methodologies, INCOSE MBSE Initiative.

Filomonov, M., Oraifige, I., Vijay, V. (2020), "A novel graph-based modelling approach for reducing complexity in modelbased systems engineering environment", International Journal of System of Systems Engineering, http://dx.doi.org/10.1504/IJSSE.2020.109142

Filomonov, M., Raju P. and Chapman C. (2016), "Graph-based modelling of systems interaction in model-based systems engineering environment", 7th International Systems & Concurrent Engineering for Space Applications Conference, Madrid, Spain

Forsgren, N., Humble, J. and Kim, G. (2018), Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations, IT Revolution Press.

Hagedorn, T., Bone, M., Kruse, B., Grosse, I., & Blackburn, M. (2020). Knowledge representation with ontologies and semantic web technologies to promote augmented and artificial intelligence in systems engineering. Insight, 23(1), 15-20. https://doi.org/10.1002/inst.12279

Haraty, R. and Hu, G. (2018), "Software process models: a review and analysis", International Journal of Engineering & Technology, 7 (2.28) 325-331, https://doi.org/10.14419/ijet.v7i2.29.13206

Harrison, W. (2016), "The Role of Graph Theory in System of Systems Engineering", https://doi.org/10.1109/ACCESS.2016.2559450

Huth, T. and Vietor, T. (2020), Systems Engineering in der Produktentwicklung: Verständnis, Theorie und Praxis aus ingenieurswissenschaftlicher Sicht. 125–130 (2020), https://doi.org/10.1007/s11612-020-00505-1

INCOSE Technical Operations (2007), Systems Engineering Vision 2020, INCOSE-TP-2004-004-02

ISO (2017), ISO 12207:2017 Systems and software engineering -Software life cycle processes

ISO (2023), ISO 15288:2023, Systems and software engineering - System life cycle processes

Mayvan, B. and Rasoolzagedan A. (2017), "Design pattern detection based on the graph theory", Knowledge-Based Systems 120 (2017) 211–225, http://dx.doi.org/10.1016/j.knosys.2017.01.007

Peterson, T. (2015), "Understanding Systems through Graph Theory and Dynamic Visualization", Ground Vehicle Systems Engineering and Technology Symposium

Peterson, T. (2016), "Exploring System Space with Graph Theory", 26th Annual INCOSE International Symposium, Edinburgh, Scotland, UK, https://doi.org/10.1002/j.2334-5837.2016.00215.x

Project Management Institute (2017). A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Sixth Edition

Sargut, G. and McGrath, R. G. (2011), "Learning to Live with Complexity", Harvard Business Review, 89, 68-76.

Schneider, B., Riedel, O., & Bauer, W. (2022), "Review: model-based systems engineering and artificial intelligence for engineering of sustainable systems", 37-59. https://doi.org/10.30844/wagb_2022_3

VDI (2021), VDI/VDE 2206: Entwicklung mechatronischer und cyber-physischer Systeme, Verein Deutsche Ingenieure, Verband der Elektrotechnik Elektronik Informationstechnik

https://doi.org/10.1017/pds.2024.265 Published online by Cambridge University Press