NORDDESIGN CONFERENCE
REYKJAVIK, ICELAND, 12-14 AUGUST 2024

AI-assisted Engineering Data Integration for Small and Medium-Sized Enterprises

Thomas Eickhoff, Arnaud Ngamakoua Hakoua, Jens C. Göbel

RPTU Kaiserslautern-Landau, Lehrstuhl für Virtuelle Produktentwicklung

Abstract: Engineering workflow management enables the digital representation and management of various collaborative engineering processes. However, additional knowledge in the minds of involved domain experts is often crucial for executing these processes efficiently. Additionally, most small and medium-sized enterprises (SMEs) do not have an adequate IT infrastructure for sufficient information management. This paper presents a generative AI-based approach to gather and integrate heterogenous digital data objects and undocumented expert knowledge into a linked meta model.

Keywords: Product Development, Artificial Intelligence (AI), Design Tools, Information Management

1 Data Integration as Prerequisite for Systems Engineering Processes

With the increasing complexity of (smart) products, product development processes require an increasing amount of integration and teamwork between different engineering disciplines (e.g., mechanics, software, electronics) (Tomiyama et al. 2019). This interdisciplinary collaboration generates several data sources, leading to challenges, particularly in terms of growing effort for their management (Dong und Srivastava 2013). Especially in the domain of information technology (IT), this means that data from the software systems commonly used in these disciplines must also be integrated. engineering disciplines require software systems like Model-Based-Engineering software or 3D modelling software to perform the requirement traceability or the digital construction, whereas mechanical technicians in production use CAD-Modell in CNC router software and generate various data. State-of-the-art development methodologies, e.g., the VDI 2206 V-Model (2206) or System-Driven Product Development (Salehi et al. 2017), can be used to manage this complexity and to link data from different information sources within a value-creation network. These methodologies are crucial in reducing unnecessary steps and accelerating innovation processes in product development (Mollahassani et al. 2023). Similar challenges can be seen in other areas of engineering data management, such as the integration of necessary data in the creation of digital twins (Layer et al. 2023; Wilking et al. 2022). The latest research visions and current research approaches in the field of AI-supported systems modeling and geometry modeling have already shown enormous potential to support these methodologies (Chami et al. 2022; Gerschütz et al. 2023). Especially in small and medium-sized enterprises, limited resources often lead to a generally lower maturity level of engineering data integration. In particular, not all data is managed in an advanced software solution, e.g., a PLM system. In addition, changes to the IT infrastructure are often not implemented immediately as this would disturb day-to-day processes. Even in larger companies, there are often several legacy IT systems that contain important information that is not integrated into a coherent metamodel.

A practical data integration has the potential to improve various design and systems engineering processes. For example, an immediate use case for an integrated data access is an enterprise-wide search. More specifically, an integrated data model that makes it easy to connect related data objects from different systems can vastly improve impact analysis for engineering change management (Ernst 2016), or simplify the preparation of data for industrial data analytics (Deuse et al. 2024). This paper presents an approach towards data integration as well as specific improvements on the basic approach to make it more usable in the context of small and medium-sized enterprises (SMEs), using both AI and conventional methods to simplify the integration process.

2 Engineering Data Integration for SME

Consolidating experiences from several industrial use cases across past research projects (e.g., the publicly funded projects InnoServPro (Aurich et al. 2019), InAsPro (Aurich et al. 2020), and AKKORD (Deuse et al. 2024)), we have identified several barriers to digitization efforts in the context of SMEs. While defining universal requirements for all SMEs is not possible, the following general requirement clusters (RC) can describe the challenges that must be addressed.

Low impact on day-to-day operation: A fitting data integration solution must be easy to implement, as the day-to-day business cannot be put on hold for disruptive changes in the company's IT infrastructure. More specifically, in SMEs, there are often existing IT systems that cannot be easily replaced. These systems have typically been chosen in isolation and are not necessarily compatible. Thus, a data integration *should not require replacing existing IT systems* (RC1).

Required effort: Additionally, there is often no dedicated in-house software development team, so *the development effort should be minimized*. This rules out approaches requiring comprehensive software solutions built around an all-encompassing data model (RC2).

Pragmatism over conceptual purity: In the spirit of finding a pragmatic approach, a potential solution should be *available in a usable form*. Approaches like the Open Services for Lifecycle Collaboration (Amsden und Speicher 2021) can be seen as an ideal solution in theory but often require context-specific extensions or awaiting the completion and subsequent adoption of additional specifications (RC3).

Existing Best Practice Solutions: Finally, a data integration strategy should follow best practice principles from software engineering, e.g., it should satisfy the *single source of truth* principle. This also includes the usage of open standards and following the Linked Data principles, which approaches such as OSLC already do (RC4).

Several existing approaches at managing data from all phases of product development, product lifecycle management, and enterprise resource planning integrate different data sets across several hierarchy levels. Some of these approaches address the integration of heterogeneous engineering data sources (Bergsjö et al. 2006; Gerhard et al. 2023) following different basic strategies for engineering data integration.

Monolithic IT integration

One approach is the integration of all necessary data into a monolithic IT system. The integration process can be supported using various advanced techniques to ease the transition (Stark et al. 2014). This results in a very coherent solution but, by definition, requires the existence of one such system. In an existing IT landscape, it is not always achievable to move everything into one system, especially in scenarios where several different, incompatible systems have been used for a long time.

Redundant Synchronization

Another approach is synchronization via a standard exchange format. In this case, all existing systems can continue to exist. However, duplicating information in several systems brings its own challenges: If different copies of a data object contain conflicting information, determining which information is authoritative is a nontrivial task. Additionally, copying information comes with increased memory consumption. Finally, the exchange format must address all edge cases of the representations found in each source system. Data exchange formats such as those defined within the Application Protocols of STEP have been used in practical solutions (Petzelt et al. 2010). Commercial solutions in this space can work quite well, as can be seen in (Bajaj et al. 2017). However, the authors point out the effort that must go into ensuring consistency across synched objects.

Decentralized Linking

A third approach is the use of decentralized approaches. Here, data objects remain in their respective source systems. Data objects from other source systems can be used via links and querying the other source system on demand. This approach is very promising, but it also requires a common interface that can handle all requirements from all source systems. Existing approaches like OSLC only cover a subset of these requirements, and additional extensions are necessary, leading to an additional development effort and a not wholly compatible solution.

Based on these conditions, a hybrid solution that requires no duplication of big data objects but is also not completely decentralized can be used. Although not originally designed specifically for SMEs, previous research has led to a hybrid approach towards data integration.

Metadata-based data integration

The metadata-based data integration approach takes some key concepts from the previous approaches. The general structure is depicted in Figure 1. Metadata-based integration includes a common integration platform, but instead of storing all integrated data (like in the monolithic approach), only metadata about the existence of a data object or a link between data objects is stored in the integration platform's Metadata Store. The actual data objects remain in their original source system. Like in the decentralized linking approach, data can be queried from the source systems using REST APIs.

Instead of relying on all software vendors to implement compatibility for this approach, each source system is connected to the integration platform via a small connector. This connector has to be developed for each new system individually but only requires the implementation of a small interface to the platform's core component. The communication between the systems and the core works via REST interfaces; all data is exchanged using the JSON format. The core manages a graph with metadata about all objects stored in the source systems and their connections. One central aspect of the solution is the fact that new nodes and connections can be added to the graph to represent connections between data objects across system orders. Data in different systems about the same conceptual object (e.g., part data from a PLM system and corresponding work plans from the ERP system) can be merged into conceptual meta-nodes.

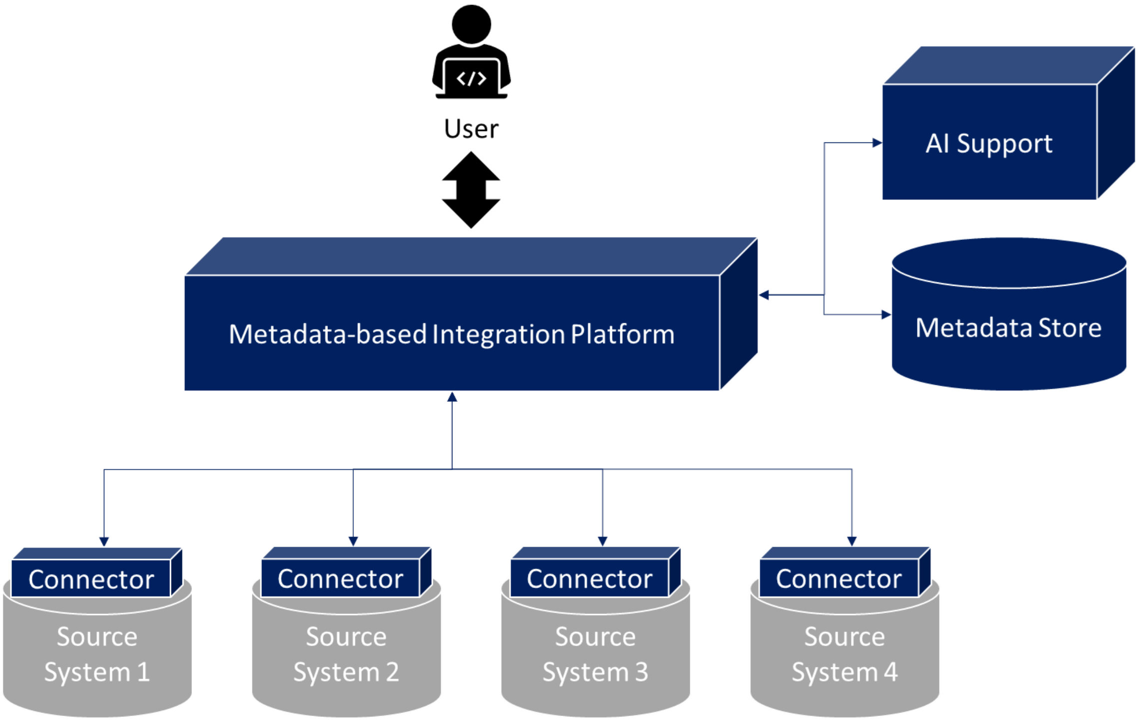

Figure 1 Metadata-based integration architecture

The specific improvements to this approach described in this paper introduce an AI support mainly used to generate connections in the metadata store.

The approach helps to address the requirements of RC1-RC4. In particular, RC1 is satisfied, as the existing source systems can be used as before. The development effort (RC2) can be reduced by choosing a simple model for the connectors and by leveraging the AI support. The use of minimalistic connectors addresses the challenges of RC3, as the approach does not presume that all source systems already speak a common language. Adherence to best practice solutions (RC4) is possible in all described approaches except for synchronization-based approaches, which violate the single-source-of-truth principle by definition.

The Semantic Product/Process Information and Digitized Engineering Repository (SP²IDER) is a software solution for integrating heterogeneous data sources based on creating a lightweight integration solution. The general structure of the platform follows the metadata-based approach described in this section, although it did not include an AI support component. The system has been described in previous publications (Eickhoff et al. 2020). The SP²IDER platform does not require a complex data model. The metadata graph only consists of objects and connections. However, the platform distinguishes between objects and types. Additional nodes and edges can be introduced on both the object and the type level. This makes it possible to import existing ontologies or type hierarchies and link them to the types extracted from the connected data sources. Integrating the types specified by OSLC or STEP profiles and providing additional API functionalities could make the system compatible with those approaches at the cost of additional development effort. Previous descriptions were based on the idea that the mapping of data types across system boundaries can be performed manually (Eiden et al. 2021). The system forms the basis of the software support developed for this paper.

3 SME-adapted Data Integration

Adapting the metadata-based approach for SMEs, in particular, opens up the potential for further improvements. In particular, the effort required in merging data from different source systems can be reduced by using an AI support. Additionally, to improve the approach's practicality for SMEs, measures need to be taken to address the ad-hoc nature of IT infrastructure in SMEs.

To address the first point, it should be noted that the metadata store currently uses no standardized ontology or general data model. This simplifies the development of the platform and new connectors but poses the question of how the heterogeneous data models of the different source systems can be merged into a coherent metadata graph. The existing solution in the SP²IDER platform has been described in (Eiden et al. 2021). However, at that point, the mapping at the data type level always happens manually. In this paper, we will present additional developments toward the automated generation of data mappings on the data type level based on generative AI.

For the second point, small and medium-sized enterprises often use ad hoc solutions for storing data in addition to more structured systems like PLM systems. Important data objects can often be found on network drives, ad hoc databases, and sometimes mission-critical Excel spreadsheets. The existing connectors for the SP²IDER platform connect to structured systems. In this paper, we will present an approach towards integrating data from these less structured sources into the metadata graph.

4 Improved Data Model Mapping Using Generative AI

The basic problem of data model mapping is as follows: Different systems A and B each contain several different data types. Each type has a set of attributes. Some of these types may correspond to each other. Corresponding types can be found by comparing the types' names and their respective attributes. However, the structure of data types can be different across systems. The SP²IDER platform only requires that each type is given as a JSON object, with only a few standardized attributes (e.g., the type's name, source system, …). Everything else is provided using the names and structure provided by the system's internal representation. This makes the problem of finding similar types in two different systems difficult to solve by traditional algorithmic approaches.

The recent advances in the field of generative AI offer another approach. Large language models (LLMs) that are capable of analyzing text in several human languages and computer languages. In particular, LLMs have been used to generate JSON code. A naïve solution to the mapping problem is to copy the types' descriptions from the SP²IDER core directly to a publicly available LLM (e.g., ChatGPT) and simply "ask nicely" to find similar types and provide a data mapping in the required format. However, this leads to suboptimal solutions for several reasons.

First, the model might not generate the answer in the required format. Even though many popular LLMs use computer programs in various programming languages in their training data (including both actual JSON data and its inspiration, JavaScript data literals), the models are usually focused on generating text in natural languages and tend to add explanations to any program code they generate.

Second, the generated mapping might not be valid or correct. LLMs tend to confabulate output that looks plausible but is not necessarily based on facts. Specifically, the model can make up types that do not appear in the corresponding source system, or it might map types to each other that do not fit together. Based on these risks, we propose a process with four steps that can help to ensure correctness. The process is shown in Figure 2.

The type's descriptions are fed to an LLM, together with a prompt describing the desired output. By providing some short example output within the prompt, we can guide the model towards the correct format, described as "Few Shot Learning" (Karpathy 2023). Additionally, a grammar file is used to force the model to generate only valid JSON output and to ensure that each mapping entry contains all required data. This "Grammar-Constrained Decoding" approach has been described in (Geng et al. 2023). Together, these two steps ensure that the output has the correct structure.

After the LLM generates a suggested mapping, we use a conventional program to filter out all mapping entries that try to map a type that does not exist in the corresponding source system. This can be achieved by comparing the generated mappings to the data types present in SP²IDER's metadata graph. As proposed in the original description of the mapping engine, the resulting mapping is then checked by a domain expert to filter out any unsuitable mappings.

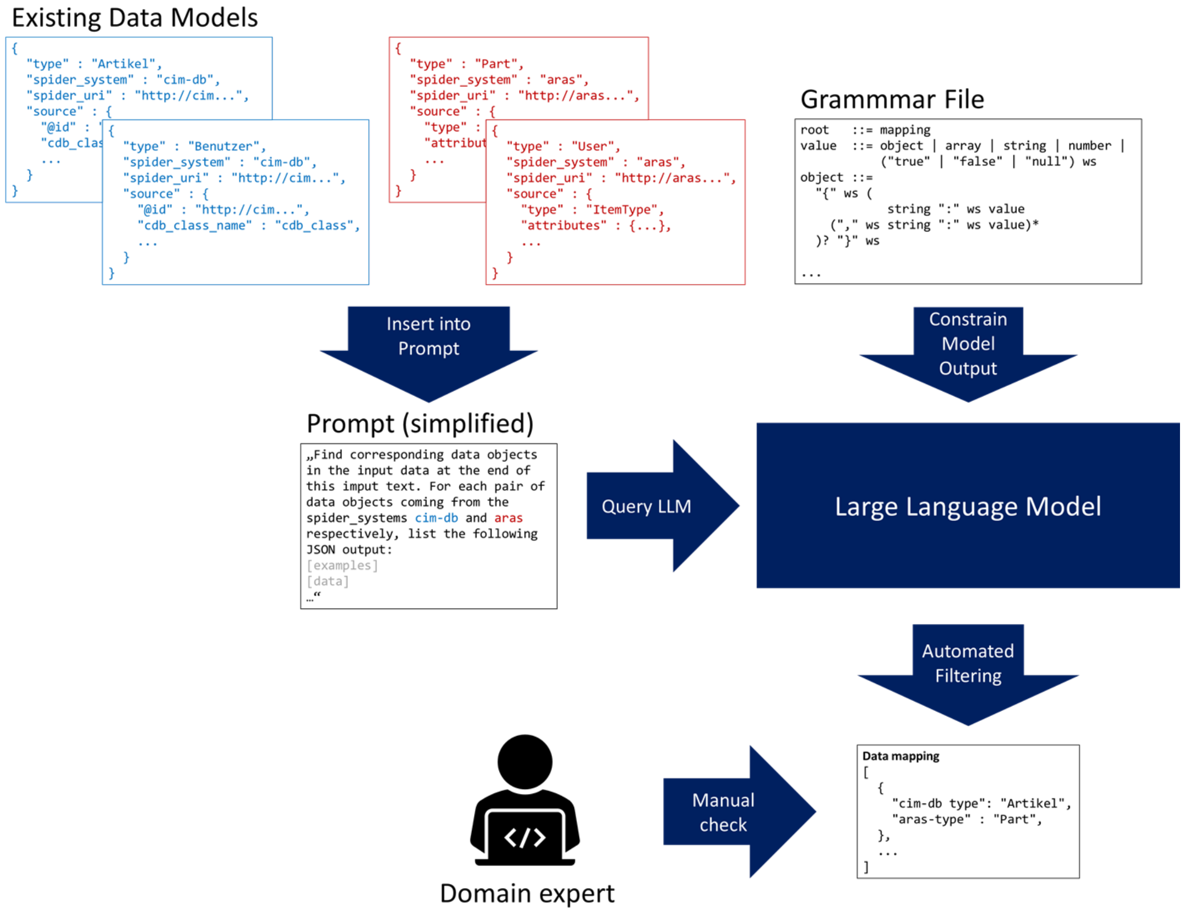

Figure 2 AI-based mapping procedure

5 Connector for Network Drives

Information retrieval from network drives has become challenging due to various individual data storage customs and the vast number of stored files (Zhou und Li 2012). Various departments frequently use different information sources, creating challenges in seamlessly sharing and managing data. We often spend substantial time seeking accurate information, such as definitions and reports, in unstructured drives. This situation becomes more challenging when the data owner has already left the company, where nobody knows where the files are and what they contain.

One approach in this use case is the implementation of a platform component that can consider the files and folders in a network drive as objects and categorize them based on their properties. If the folder in the information's sources contains recursive folders, they will all be listed. The mentioned platform fetches the pertinent object's properties (metadata), identifies, and lists all found metadata to selectively filter them based on the expressions or a group of expressions set by the user. The first step to look for files with relevance is to identify the properties that fit the looking Keyword or a group of keywords with a high probability. The SP²IDER platform saves the found metadata and requires a connection to the data source system and a bilateral communication instance known as a connector (Eyers und Voulgaris 2022). Connectors enable a defined data transfer from network drives to the platform and in the other way. This connector has access to file metadata like file name, extension, creation, or modification date for the file classification. Further information, such as the file owner, can be extracted and used to track the collaboration between colleagues. Files with relevance can be found with this approach and stored in a more structured repository.

A second approach relates to a scenario, such as finding documentation for a device or seeking specific information related to the usage of a device component. Many SMEs use an unstructured knowledge repository or more repositories. They store all kinds of information, such as observations, videos, and communication records, in files and use them to accelerate daily work. It is useful to collect all of them to enable a better understanding of new tasks and potentials (Okoli und Schandl 2009). This approach aims to explore the feasibility of configuring a new connector capable of looking for a keyword or a group of keywords within company drives and generating knowledge. A search algorithm can be used to enhance the efficiency of the research and significantly reduce the search time. A knowledge integration mechanism employs the connector to identify and open accurate files at a high level of abstraction. The connector considers all files as objects, with their content acting as their attributes. Additionally, the search results include some content keywords that can be processed to provide clues for understanding workflows in a company and interpreted as new knowledge source and are enriched with user's expertise, consolidated into new knowledge resources. Through an integration platform, this knowledge can be linked to existing data objects. This approach assists users in creating new expertise from existing information and ensures not only the knowledge transfer between colleagues but also an efficient data exchange.

6 Prototypical Implementation of the SME-adapted Data Integration Approach

The SP²IDER platform is currently being developed as a Python program. The metadata store is based on the existing graph database ArangoDB. The AI-based mapping engine is implemented as a component within the SP²IDER core. As such, it only needs to consume the SP²IDER core's data model and generate mappings accordingly. The mapping procedure itself is also implemented in Python. The LLM used for the mapping generation is run as an external process; the communication works via a REST interface. The particular model we chose was leo-hessianai-13b-chat-bilingual (Plüster 2023), which was chosen for its multilingual capabilities as well as its modest hardware requirements. The model is licensed under the Llama-2 community license, which allows small and medium-sized enterprises to run the model in a commercial context (Touvron et al. 2023).

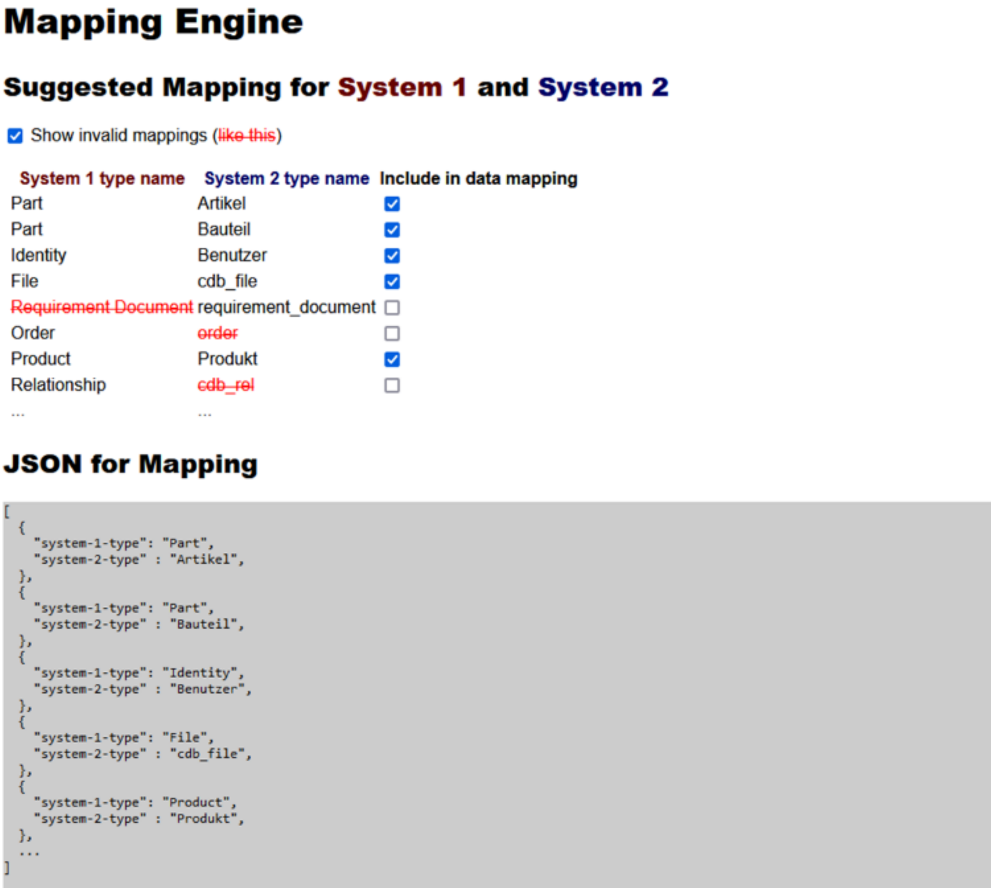

Figure 3 Prototypical Implementation of the AI-supported mapping engine

The network drive connector is implemented as a regular SP²IDER connector. Advanced functionalities like semantic search can also use the LLM component to provide embeddings for textual documents.

Applying the AI-based mapping engine to a newly connected source system (e.g. a network drive) helps to integrate the connected data to the existing data models. Figure 3 shows a preliminary user interface for the final manual check of such a mapping.

The generated mappings show the potential of LLMs in reducing repetitive tasks (corresponding data objects which have the same name in different languages are identified with a high accuracy). On the other hand, the prototype also highlights the necessity of a final check by a domain expert, as the model includes "hallucinated" data types and nonsensical mappings. The quality of the results varies with small modifications to the system prompt or changes to the parameters that control inference on the LLM.

After the mapping is generated, the unified data model can be used to access data from the different sources via the integration platform. The platform provides a visual representation of the metadata graph, including search functionality across all connected data sources. This can be used to improve processes such as engineering change management (Ernst 2016).

7 Future Improvement Potential for the AI Support

Both the AI-based mapping improvements and the network drive connector are currently being developed further. The precision and correctness of the AI-based mappings could further be improved by using a more capable language model or even a model that has been fine-tuned to this particular use case. Additionally, advanced techniques such as Retrieval-Augmented Generation could be used to enhance the model's understanding of the problem space (Lewis et al. 2001). For the network drive connector, advanced semantic search capabilities could be provided using the upcoming multimodal LLMs to provide semantic search capabilities for images and video files. This involves executing a semantic search into the found files. The research results could then be processed or displayed as pop-up windows. This allows the user to read the highlighted information in each pop-up and select the one with the most relevance.

Further processing of the generated knowledge items could also be considered. Found knowledge items could not only be displayed and highlighted. They could also be used from an algorithm that performs a series of analyses and stores the information on a knowledge basis, which is the foundation of a chatbot. The user interacts with the chatbot; he is required to enter his questions in the chatbot's text dialog and should receive more content-specific information.

8 Conclusion

The presented concepts and software prototypes provide a metadata-based approach to data integration. While a completely decentralized approach is a conceptually cleaner model, the introduction of a metadata store allows for a simple management of data links across system boundaries. The presented extensions make the existing platform more capable (by integrating generative AI into the data model mapping process) and more practical (by acknowledging the existing "shadow IT" in companies and providing a method of integrating network drives into a coherent metadata management. Both extensions should not be seen as individual, finished solutions but as building blocks along the way toward more coherent information management.

References

Amsden, Jim; Speicher, S. (2021): OSLC Core Version 3.0. Part 1: Overview. Project Specification Draft. Hg. v. OASIS. Online verfügbar unter http://docs.oasis-open-projects.org/oslc-op/core/v3.0/ps02/oslc-core.pdf.

Aurich, Jan C.; Koch, Walter; Kölsch, Patrick; Herder, Christoph (2019): Entwicklung datenbasierter Produkt-Service Systeme. Berlin, Heidelberg: Springer Berlin Heidelberg.

Aurich, Jan C.; Pier, Marcus; Siedler, Carina; Sinnwell, Chantal (Hg.) (2020): Bedarfsgerechte Digitalisierung von Produktionsunternehmen. Ein modulares Transformationskonzept als praxisorientierter Ansatz. Synnovating GmbH. Erste Auflage. Kaiserslautern: Synnovating.

Bajaj, Manas; Backhaus, Jonathan; Walden, Tim; Waikar, Manoj; Zwemer, Dirk; Schreiber, Chris et al. (2017): Graph-Based Digital Blueprint for Model Based Engineering of Complex Systems. In: *INCOSE International Symp* 27 (1), S. 151–169. DOI: 10.1002/j.2334-5837.2017.00351.x.

Bergsjö, Dag; Malmqvist, Johan; Ström, Mikael (2006): Architectures for mechatronic product data integration in PLM systems. In: DS 36: Proceedings DESIGN 2006, the 9th International Design Conference, Dubrovnik, Croatia.

Chami, Mohammad; Abdoun, Nabil; Bruel, Jean-Michel (2022): Artificial Intelligence Capabilities for Effective Model-Based Systems Engineering: A Vision Paper. In: *INCOSE International Symp* 32 (1), S. 1160–1174. DOI: 10.1002/iis2.12988.

Deuse, Jochen; Klinkenberg, Ralf; West, Nikolai (2024): Industrielle Datenanalyse. Wiesbaden: Springer Fachmedien Wiesbaden.

2206, 2021: Development of mechatronic and cyber-physical systems.

Dong, X. L.; Srivastava, D. (2013): Big data integration. In: 2013 IEEE 29th International Conference on Data Engineering (ICDE). 2013 29th IEEE International Conference on Data Engineering (ICDE 2013). Brisbane, QLD, 08.04.2013 - 12.04.2013: IEEE, S. 1245–1248.

Eickhoff, Thomas; Eiden, Andreas; Göbel, Jens Christian; Eigner, Martin (2020): A Metadata Repository for Semantic Product Lifecycle Management. In: *Procedia CIRP* 91, S. 249–254. DOI: 10.1016/j.procir.2019.11.006.

Eiden, Andreas; Eickhoff, Thomas; Gries, Jonas; Göbel, Jens C.; Psota, Thomas (2021): Supporting semantic PLM by using a lightweight engineering metadata mapping engine. In: *Procedia CIRP* 100, S. 690–695. DOI: 10.1016/j.procir.2021.05.146.

Ernst, J. (2016): Systemübergreifendes Änderungsmanagement: graphbasierte Identifikation und Visualisierung betroffener Konfigurationselemente aus PLM und ERP: Technische Universität Kaiserslautern (Schriftenreihe VPE).

Eyers, David; Voulgaris, Spyros (Hg.) (2022): Distributed Applications and Interoperable Systems. Cham: Springer International Publishing (13272).

Geng, Saibo; Josifoski, Martin; Peyrard, Maxime; West, Robert (2023): Grammar-Constrained Decoding for Structured NLP Tasks without Finetuning.

Gerhard, Detlef; Salas Cordero, Sophia; Vingerhoeds, Rob; Sullivan, Brendan P.; Rossi, Monica; Brovar, Yana et al. (2023): MBSE-PLM Integration: Initiatives and Future Outlook. In: Frédéric Noël, Felix Nyffenegger, Louis Rivest und Abdelaziz Bouras

(Hg.): Product Lifecycle Management. PLM in Transition Times: The Place of Humans and Transformative Technologies, Bd. 667. Cham: Springer Nature Switzerland (IFIP Advances in Information and Communication Technology), S. 165–175.

Gerschütz, Benjamin; Sauer, Christopher; Kormann, Andreas; Nicklas, Simon J.; Goetz, Stefan; Roppel, Matthias et al. (2023): Digital Engineering Methods in Practical Use during Mechatronic Design Processes. In: *Designs* 7 (4), S. 93. DOI: 10.3390/designs7040093.

Karpathy, Andrej (2023): State of GPT. Microsoft BUILD, 2023. Online verfügbar unter https://karpathy.ai/stateofgpt.pdf.

Layer, Max; Neubert, Sebastian; Tiemann, Lea; Stelzer, Ralph (2023): IDENTIFICATION AND RETRIEVAL OF RELEVANT INFORMATION FOR INSTANTIATING DIGITAL TWINS DURING THE CONSTRUCTION OF PROCESS PLANTS. In: *Proc. Des. Soc.* 3, S. 2175–2184. DOI: 10.1017/pds.2023.218.

Lewis, Patrick; Perez, Ethan; Piktus, Aleksandra; Petroni, Fabio; Karpukhin, Vladimir; Goyal, Naman et al. (2001): Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In: *Advances in neural information processing systems - Proceedings of the first 12 conferences*, S. 9459–9474. Online verfügbar unter https://proceedings.neurips.cc/paper_files/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf.

Mollahassani, Damun; Eickhoff, Thomas; Juresa, Yannick; Göbel, Jens C. (2023): Knowledge Collaboration Approach in Smart Product Innovation Networks. In: *Procedia CIRP* 119, S. 662–668. DOI: 10.1016/j.procir.2023.02.158.

Okoli, Adaora; Schandl, Bernhard (2009): Extraction of contextual metadata from file system interactions.

Petzelt, Dominik; Schallow, Julian; Deuse, Jochen (2010): Data integration in Digital Manufacturing based on application Protocols. In: 2010 3rd International Conference on Computer Science and Information Technology. 2010 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT 2010). Chengdu, China, 09.07.2010 - 11.07.2010: IEEE, S. 475–479.

Plüster, Björn (2023): Leolm: Igniting German-language LLM Research. Hg. v. LAION. Online verfügbar unter https://laion.ai/blog/leo-lm/.

Salehi, Vahid; Burseg, Lukas; Paetzold, Kristin; Chahin, Abdo; Taha, Jihad; Rieger, Thomas (2017): Integration of Systems Engineering Approach in Product-Lifecycle-Management by Means of a Mechatronic System. In: Gauthier Fanmuy, Eric Goubault, Daniel Krob und François Stephan (Hg.): Complex Systems Design & Management. Cham: Springer International Publishing, S. 231–232.

Stark, Rainer; Damerau, Thomas; Hayka, Haygazun; Neumeyer, Sebastian; Woll, Robert (2014): Intelligent Information Technologies to Enable Next Generation PLM. In: Shuichi Fukuda, Alain Bernard, Balan Gurumoorthy und Abdelaziz Bouras (Hg.): Product Lifecycle Management for a Global Market, Bd. 442. Berlin, Heidelberg: Springer Berlin Heidelberg (IFIP Advances in Information and Communication Technology), S. 485–495.

Tomiyama, Tetsuo; Lutters, Eric; Stark, Rainer; Abramovici, Michael (2019): Development capabilities for smart products. In: *CIRP Annals* 68 (2), S. 727–750. DOI: 10.1016/j.cirp.2019.05.010.

Touvron, Hugo; Martin, Louis; Stone, Kevin; Albert, Peter; Almahairi, Amjad; Babaei, Yasmine et al. (2023): Llama 2: Open Foundation and Fine-Tuned Chat Models. Online verfügbar unter http://arxiv.org/pdf/2307.09288v2.

Wilking, F.; Sauer, C.; Schleich, B.; Wartzack, S. (2022): SysML 4 Digital Twins – Utilization of System Models for the Design and Operation of Digital Twins. In: *Proc. Des. Soc.* 2, S. 1815–1824. DOI: 10.1017/pds.2022.184.

Zhou, Kevin Zheng; Li, Caroline Bingxin (2012): How knowledge affects radical innovation: Knowledge base, market knowledge acquisition, and internal knowledge sharing. In: *Strategic Management Journal* 33 (9), S. 1090–1102. DOI: 10.1002/smj.1959.

Contact: Eickhoff, Thomas, RPTU Kaiserslautern-Landau, thomas.eickhoff@mv.rptu.de