# Knowledge Extraction from CAD Models: Modern Applications of Design Feature Graphs

Sarah Steininger[1,2]*, Florian Tenzer[1], Johannes Fottner[1]

[1] TUM School of Engineering and Design, Dep. of Mechanical Engineering, Technical University of Munich, Germany
[2] BMW Group, Munich, Germany

** Corresponding Author:*
*Sarah Steininger, M. Sc.*
*TUM School of Engineering and Design*
*Technical University of Munich*
☎ *+49 151 60115250*
✉ *sarah.steininger@tum.de*

## Abstract

Increasing competitive pressure and shorter product life cycles necessitate more efficient development processes. Although CAD models contain valuable design knowledge, particularly design feature information (DFI) such as parameters and modelling logic, this potential has remained largely untapped. However, modern technologies such as knowledge graphs, graph neural networks and natural language processing offer new opportunities for exploiting this information. This paper presents a structured methodology for identifying and evaluating DFI-based applications that can improve product development. Based on a literature review, expert interviews and iterative evaluation, eight applications were developed to address key challenges, ranging from design reuse and model quality analysis to manufacturing integration and requirements traceability. The results demonstrate the potential of DFI to act as a semantic bridge between engineering disciplines, paving the way for future research and the industrial implementation of knowledge-based CAD applications.

## Keywords

*CAD, design feature information, knowledge graphs, product development, product lifecycle management*

 | DOI: 10.35199/dfx2025.11

## 1. Motivation

In today’s rapidly evolving industrial landscape, companies are under high pressure to innovate quickly while maintaining high standards of quality, efficiency, and cost-effectiveness. One of the primary levers for achieving these objectives is the digitalization of product development. Modern product creation processes heavily rely on computer-aided systems such as CAD (Computer-Aided Design), CAE (Computer-Aided Engineering), and CAM (Computer-Aided Manufacturing). These systems produce and manage vast amounts of data during the design and engineering phases [1].

While geometric data (such as B-Rep models) is commonly exchanged and reused, the underlying logic of the design is often lost or ignored. Yet, the intent of the designer is captured during the creation of CAD models using parametric features, constraints, sketches, and feature dependencies. This body of knowledge is referred to as Design Feature Information (DFI). DFI holds the key to understanding how and why a part was designed in a particular way. It enables better support for downstream processes such as simulation, manufacturing, and quality control. However, current data exchange standards and practices focus almost exclusively on geometry, leaving DFI underutilized [2].

At the same time, emerging technologies from the fields of artificial intelligence and data science - particularly knowledge graphs, deep learning, and natural language processing - have shown promise in structuring and interpreting complex technical data. However, integrating artificial intelligence into product development processes is happening at a slow pace and requires substantial effort [3, 4]. Applying these tools to DFI promises the opportunity to enhance effective use of design knowledge across the product lifecycle. This paper is motivated by the need to systematically explore and exploit the value of DFI.

## 2. State of the Art

### 2.1. Product Development Tools and Challenges

Modern product development involves multiple overlapping disciplines and digital systems. The core systems like CAD, CAE, CAM, or CAP generate technical data during various phases of design, simulation, planning, and manufacturing. These CAx tools are mostly supported by Product Data Management (PDM) and Product Lifecycle Management (PLM) systems, which organize and store technical documents, versions, and metadata [5]. Despite the digital maturity of these tools, several systemic challenges remain unresolved. According to [6], model-based product development lacks structured mechanisms for design retrieval and reuse, which often causes engineers to reinvent existing components. Yet, effective reuse strategies can reduce development time by up to 30% [7]. Increasing model complexity further complicates the understanding of internal CAD structures, which negatively affects design performance and increases risk for errors [8]. Additionally, insufficient model quality can lead to significant overhead during reuse or data exchange [9]. The traceability of requirements to design elements is also frequently missing, especially due to heterogeneous and unstructured information formats. Finally, while manufacturability is a critical consideration in design, the transfer of related knowledge into production workflows remains poorly integrated [10].

### 2.2. Knowledge Graphs and Deep Learning

A knowledge graph is a structured representation of information that represents entities and their relationships. According to [11], it acquires and integrates information into an ontology and applies reasoner to derive new knowledge. An ontology provides formalized structures for domain-specific knowledge representation, facilitating shared understanding and interoperability in data-driven systems. In the area of CAD, knowledge graphs also have gained increasing attention, especially in their application within CAD [12].

When combined with AI techniques, knowledge graphs improve applications such as recommendation systems, semantic search or anomaly detection. Deep learning, especially Graph Neural Networks (GNN), further enhances these capabilities by learning from the structure and attributes of graphs. GNNs create embeddings that can be used for clustering, classification, and prediction [13]. More recently, Large Language Models (LLMs) have shown promise in combination with knowledge graphs [14].

### 2.3. Design Feature Information

DFI is not a fixed termin in literature. Related terms are design sequence, model tree, feature tree, feature graph, model graph or feature representation. It can be defined as all design-related digital objects extractable from the CAD modeling process, which capture the process and logic of CAD model creation (see Figure 1).

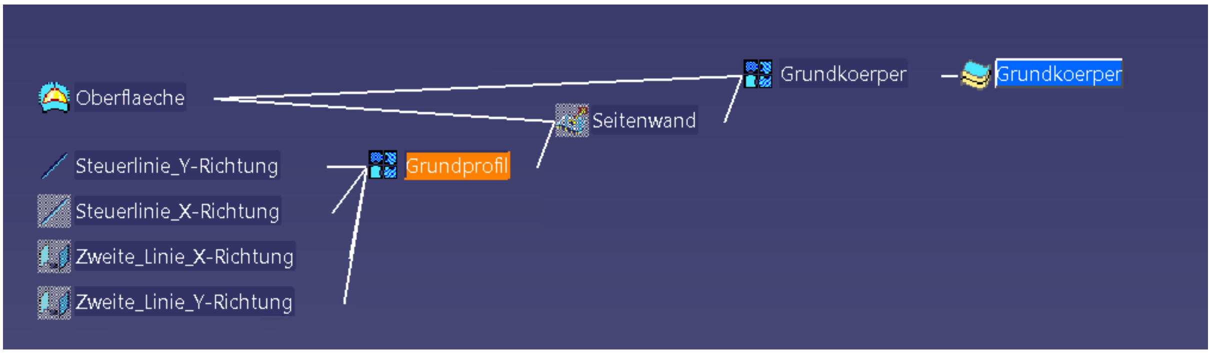


Figure 1: Dependency visualization of a parametric-associative part using parent-child functionality in CATIA V5.

Unlike geometric representations, DFI can be modeled as a graph that expresses not only what the shape is, but also how and why it was created. For analyzing and discussing DFI, the paper focuses on a single CAD model, and thus on a component as a self-contained system. The relationships between assemblies (i.e. the conditions between two components) are not considered. Additionally, the types and relationships of DFIs can vary depending on the design tool used. The design graphs analyzed were created using CATIA V5. Furthermore, the modelling process can be displayed in different formats, such as sequence or graph. For further processing and tool-dependent reasons, the DFIs are displayed as a graph (see Figure 2). A detailed analysis of the graph reveals the nodes and relationships shown in Table 1.

Table 1: Type of nodes, type of dependencies and their description.

| Nodes | Description | Examples |
|---|---|---|
| Metadata | Information about the CAD document or the author, not directly related to the model | PartDoc, Author |
| Design Element | Constructive elements for defining the geometry and shape of a model | Translate, Assemble, Plane, Line, Extrude, Point, Add, Circle, AxisSystem, LinearRepartition |
| Structural Element | Structuring elements for organizing geometry and parameters | GeoSets, ParameterSet |
| Parameters | Values that define the dimensions and properties of the model | RealParam, Length, BoolParam, StrParam, Direction, IntParam, Angle, Thickness |
| Relations | Links or dependencies between parameters and elements | Formula |

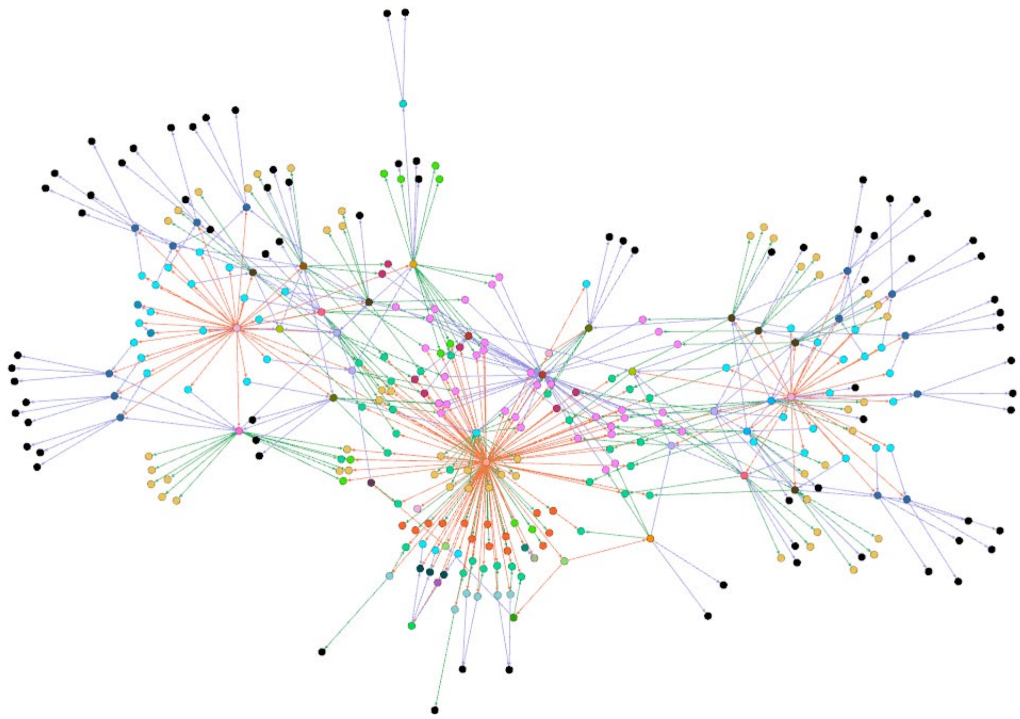

Figure 2: Example dependency graph of a parametric-associative part from a Lego brick.

Approaches for combining knowledge graphs and CAD are varied. A key focus lies in design assistance, such as the extraction of procedural knowledge and support for the designer during modeling through rule-based design recommendation [15] or feature prediction using machine learning models [16]. Other approaches target the retrieval and reuse of design components, employing similarity-based search mechanisms [17] or identifying recurring design patterns for reuse in new designs [18]. Complementary work addresses the structural analysis of CAD models, examining aspects such as feature complexity and dependency [19]. This research thread is promising, but still immature. However, the growing availability of data and advances in AI technologies are accelerating its development and expanding its potential impact.

## 3. Problem Statement and Research Objectives

Although CAD systems store rich design information internally, current practices do not exploit this information to its full potential. Most research and industrial applications still rely on geometric models and keyword-based metadata for search, analysis, and automation. Although the geometrical form is widely available and accessible to learning models [16, 20 bis 22], it does not present knowledge about the process and logic of CAD model creation. Technological advancements in knowledge graphs and deep learning provide enormous application potential to product creation due to new processing and inference techniques [13]. Promising results from these advancements should be transferable to design, planning, manufacturing, and production [10]. Research on this transfer is still in the early stages with a need to evaluate proposed approaches [23]. The potential of DFI combined advances in processing and inference technologies makes addressing global competition-induced efficiency challenges in product creation valuable. The question of how to utilize DFI is therefore stated as a problem. This leads to the following research question:

In what ways can DFI be used to enhance product creation using modern technologies?

Table 2: Solution Requirements and Constraints.

| Symbol | Requirement Description |
|---|---|
| R1 | Embrace technological advancements |
| R2 | Solve painpoint in design and product creation |
| R3 | Provide value to legacy design databases |
| Symbol | Constraint Description |
| C1 | Use of DFI as an essential part |
| C2 | Use of part information only |

In order to answer the research question, the requirements (R) and constraints (C) for the solution set have been defined as refinements of the question or practical limitations (see Table 2). Extracting DFI via APIs has been possible since the advent of feature-based design, with research focusing on specific use cases. Earlier algorithms were limited by the technology available at the time. Emerging technologies such as deep learning and knowledge graphs offer great potential, yet remain underutilised. Solutions should leverage these technologies (R1). As discussed in Section 2.1, applications should either address current challenges or unlock untapped potential (R2). Generative design has an impact on the early stages of product development, making new approaches necessary to add value to design databases and knowledge-driven processes (R3). Extracted DFI is a deterministic source of design intent and forms the basis of application value in this thesis (C1). Due to the large volume of CAx data, only partial information is considered, excluding assemblies (C2).

## 4. Methodology

In order to systematically explore and validate the potential of DFI-based applications, a three-phase methodology is adopted (see Figure 3).

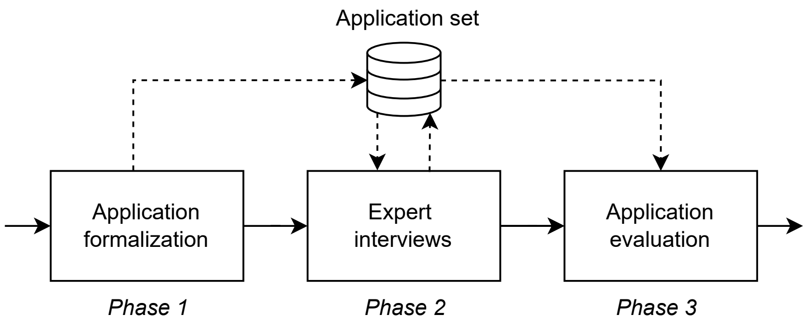


Figure 3: Methodology for identifying and evaluating DFI-based applications.

### 4.1. Phase 1: Application Formalization

Based on a literature review and analysis of existing tools, an initial set of potential application areas is identified. These were formalized by defining, the problem scope, input and output data types, technical integration steps, potential technologies and algorithms, as well as integration points along the product lifecycle. Example of starting concepts were:

- Model structure analysis, comparable to [19]
- Modeling sequence analysis, comparable to [16].
- Automatic conversion of DFI into a neutral exchange format.

### 4.2. Phase 2: Expert Interviews

Structured interviews were carried out with eight domain experts from academia and industry to gather insights. The sample included three CAD design specialists with practical experience in automotive and mechanical engineering, as well as five systems engineers and knowledge management experts specializing in PLM integration from different companies in Germany. The interviews were conducted primarily online over a period of three months and were organized into four thematic sections: expert self-assessment, an introduction to the DFI concept and example models, a discussion of potential application candidates, and an open ideation and refinement phase.

### 4.3. Phase 3: Application Evaluation

The final phase focuses on finalizing and assessing the identified applications of DFI. After incorporating insights from literature and expert interviews, this phase filters and structures the applications based on their data requirements (CAD, PDM, PLM) and fulfillment of defined requirements and constraints (see Table 2). Ultimately, eight applications are presented.

## 5. Results and Discussion

A total of eight applications (A1 to A8) were identified that systematically exploit the potential of DFI to improve product development processes. These applications were developed based on literature analysis, expert interviews and iterative evaluation. They represent concrete approaches for making structured modeling information from CAD systems usable. Key challenges are addressed such as design reuse, model quality, requirements tracking and manufacturing integration using modern technologies such as graph neural networks, knowledge graphs and natural language processing. The applications were classified according to the data input (see Figure 4) - i.e. whether they exclusively require CAD data (DFI), additional PDM data (e.g. version trees, documents) or also PLM data (planning and production information).

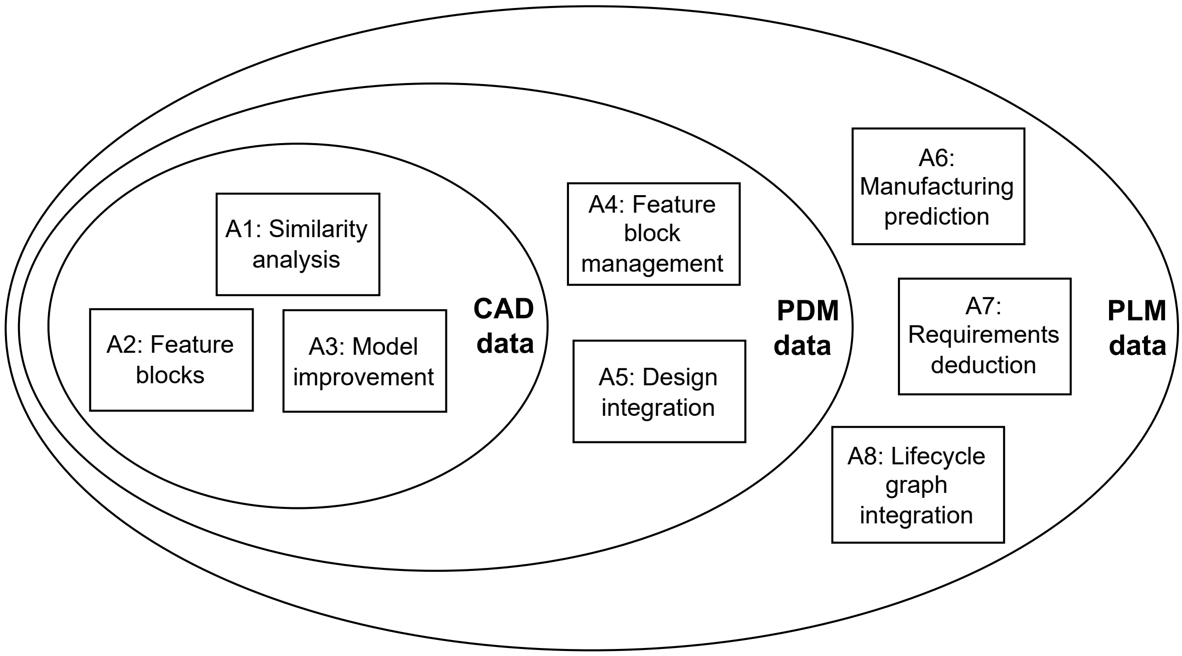


Figure 4: Application categorization by input data.

### 5.1. Applications using CAD data

#### 5.1.1. A1: Design Similarity Analysis

As model similarity analysis has been manual in the past, design retrieval and automated comparison capabilities should have been established. Current solutions mostly rely on geometrical and keyword-based similarities [2, 16, 18]. However, these inputs convey little, if any, modelling intent, even though this should be the desired measure of comparison for reuse.

This application uses the semantic structures within DFI to perform a similarity analysis of CAD models. In contrast to conventional methods that are primarily based on geometry or metadata, A1 enables a comparative evaluation based on the modeled design intent. This could be implemented by converting the DFI into a feature knowledge graph structure, which is embedded using graph embedding techniques. The aim is to increase efficiency in the reuse of existing designs, for example through automated suggestions of similar components.

#### 5.1.2. A2: Feature Blocks

Partial design reuse remains difficult due to lack of modular, reusable substructures within CAD models. Common features are often repeated manually. A2 focuses on the identification of reusable substructures (feature blocks) in CAD models. These modular design patterns are extracted from the graph structure, analyzed and evaluated using learning-based methods such as GNNs. Thus, these feature blocks can be classified, cataloged and made available across different projects. During new part design, the system could suggest frequently reused feature blocks based on geometric context and function (e.g., mounting patterns). The application aims to increase design modularity and provide standardized building blocks.

Initial approaches for this application are already being researched. [18] use hierarchical tree clustering to identify reusable patterns in DFI. Additionally, they provide five characteristics that are necessary for good design patterns. [24] finds and classifies patterns of holes that other patterns can substitute for specific parts.

#### 5.1.3. A3: Model Improvement

This application tackles the issues of design complexity and quality deficiencies in existing CAD models. It identifies design patterns that are inefficient, difficult to reuse, or potentially error-prone. These patterns are then compared with 'good' examples in a multi-stage analysis process, after which automated suggestions for improvement are generated. Additionally, model components with high automation potential can be identified (e.g. for naming or parameter settings). The application is intended to assist with model optimization.

Regarding this application, [19] provides design quality metrics applicable to DFI together with a methodology for an assistance system communicating model quality. [9] find a way to utilize DFI for model repair. Ranking design patterns for their quality may require additional manual labeling to enable learning-based algorithms. Another part of automation analysis could be the preparation for CAE tasks like finite element analysis.

### 5.2. Applications using PDM Data

#### 5.2.1. A4: Feature Block Management

Feature blocks evolve over time. Keeping track of changes and variant-specific adaptations is challenging, especially across versions and product lines. Based on A2, A4 introduces a systematic approach to manage feature blocks across versions and variants. For this purpose, PDM data such as version trees and change notes are also integrated. Not only are feature blocks identified, but also tracked, and synchronized across model histories. This enables transparent tracking of structural changes within assemblies or product families, documentation of changes, and promotion of reuse at the modular level. One possible use case is that, when updating a critical design feature, engineers are informed of all the affected parts and can propagate or isolate the change as required.

As this application is an extension to application A2, the referenced literature remains relevant. Additionally, [25] introduces a version management approach for parametric design models. Commercial solutions like the 3D Experience Compare app provide visual comparisons between models.

### 5.2.2. A5: Design Integration (Design Knowledge Assistant)

Understanding the design structure of an existing model can be complex and time-consuming. Design knowledge should be retrievable in a human-friendly format to help overcome misunderstandings. A5 links DFI with natural language, enabling interactive access to design knowledge. A knowledge graph, generated from CAD and PDM data, forms the basis of an assistance system that can respond to domain-specific queries using a large language model. This application is designed to improve understanding of existing CAD models. It not only accesses structured data, but also context-related documents and histories.

[26] provides a general knowledge base construction method for multimodal input. The part knowledge base construction process described by [15] is limited. [27] describes a recent approach for question answering combining LLM with 3D CAD knowledge. It does not seem that an approach using DFI and part versioning exists yet.

## 5.3. Applications using PLM Data

### 5.3.1. A6: Manufacturing Prediction

Manufacturing planning is often decoupled from design. Late-stage manufacturability checks lead to costly redesigns and delays. This application uses DFI and production data (PLM) to predict manufacturing information such as machining processes and sequences. Production patterns are recognized based on structured DFIs and transferred to new parts. GNNs or transformer-based models are used to derive production templates from comparable parts. The aim is to take manufacturing aspects into account as early as the design phase and thus make production planning more efficient.

[28] present a methodology to create machining program templates from part structure, using inference and semantic matching. This approach could be used as a benchmark for evaluating the new application. For capturing process intent for preselecting machining templates, [29] generate a process knowledge graph from DFI, which could be an intermediate step in increasing prediction accuracy.

### 5.3.2. A7: Requirements Deduction

Requirements are documented separately from CAD, making it hard to trace decisions or validate compliance. To better integrate product planning and product design, requirements need to be mapped to regions of design features. This allows better requirements tracing and more intuitive knowledge management. The seventh application involves the semantic derivation of technical requirements from the model structure. Linking DFI with planning documents (e.g. specifications and functional specifications) enables the automatic assignment of requirements to model areas. This application could be an important step towards automated requirements tracing and promotes consistency between the planning and design phases.

[30] created a similar application structure to retrieve parts that fit planning documents. [31] have performed semantic mapping of requirements to components, but applying this to certain features or feature blocks seems even more promising.

### 5.4. A8: Lifecycle Graph Integration

A8 broadens the scope of application to cover the entire product lifecycle. DFI is integrated into a cross-domain lifecycle knowledge graph linking requirements, design features, production steps, and quality measures. The aim is to create a lightweight, semantically sound extension to digital twins or digital threads. The Lifecycle Graph enables complete traceability across process and system boundaries, opening up new possibilities for product analysis, traceability, and systemic learning.

[32] translate semantic design rules into a knowledge base in preparation for design rule recommendation. A similar process could be used for this application. A mapping procedure of requirements and gearbox component models is presented by [31]. Relevance of a lifecycle graph, also known as an engineering graph, is increasing. Such systems are able to integrate a whole product database into one graph [33]. Industry relevance has been recognized as well.

## 6. Conclusion and Outlook

This research study investigates the potential of DFI as a strategic knowledge resource in product development. Through the formalization of DFI, validation of its industrial relevance via expert interviews and evaluation of specific use cases, this research demonstrates how semantically enriched feature data can lay the groundwork for intelligent engineering applications. Eight application scenarios were developed and classified, ranging from design reuse and model quality improvement to requirement tracing and lifecycle integration. Each application leverages modern technologies, such as GNNs, knowledge graphs or natural language processing, to make design intent, modelling patterns and legacy decisions accessible, explainable and reusable.

Future work should focus on the technical implementation and empirical validation of selected applications, paying particular attention to scalability, data availability, and integration into existing engineering workflows. Developing uniform data standards for feature graphs and linking them with PDM and PLM systems is an important step towards industrial implementation. Further increasing the practical benefits could be achieved by extending the Feature Knowledge Graph to include probabilistic conclusions, user interactions or collaborative processing. Ultimately, DFI-based systems could serve as a semantic memory layer within model-based development environments, converting CAD data into a valuable source of technical knowledge.

## Acknowledgments

I would like to thank the experts who were interviewed, as their valuable insights contributed significantly to the development of the application scenarios. My thanks also go to Prof. Dr.-Ing. Sandro Wartzack for his valuable support.

## Bibliography

[1] Ulrich, K. T. u. Eppinger, S. D.: Product design and development. New York, McGraw-Hill Education 2016

[2] Li, X., Chen, C.-H., Zheng, P., Jiang, Z. u. Wang, L.: A context-aware diversity-oriented knowledge recommendation approach for smart engineering solution design. Knowledge-Based Systems 215 (2021)

[3] Steininger, S., Camci, H. u. Fottner, J.: Current State, Potentials and Challenges for the Use of Artificial Intelligence in the early Phase of Product Development: A Survey. IEEM 2024. Piscataway, NJ: IEEE 2024

[4] Mueller, P. u. Mikelsons, L.: Exploring the Potentials and Challenges of Deep Generative Models in Product Design Conception, 2024

[5] Stark, J.: Product lifecycle management (Volume 1). 21st century paradigm for product realisation. Decision engineering. Cham: Springer 2022

[6] VDI 2221 Blatt 1:2019. *Entwicklung technischer Produkte und Systeme - Modell der Produktentwicklung*

[7] Jackson, C. u. Buxton, M.: The design reuse benchmark report: seizing the opportunity to shorten product development, 2007.

[8] Abulawi, J.: Ansatz zur Beherrschung der Komplexität von vernetzten 3D-CAD-Modellen, 2012

[9] Yang, J. u. Han, S.: Repairing CAD model errors based on the design history. Computer-Aided Design 2006

[10] Cao, W., Robinson, T., Hua, Y., Boussuge, F., Colligan, A. R. u. Pan, W.: Graph Representation of 3D CAD Models for Machining Feature Recognition With Deep Learning. Proceedings of the ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference. 2020

[11] Lisa Ehrlinger u. Wolfram Wöß: Towards a Definition of Knowledge Graphs. International Conference on Semantic Systems. 2016

[12] Xiao, Y., Zheng, S., Shi, J., Du, X. u. Hong, J.: Knowledge graph-based manufacturing process planning: A state-of-the-art review. Journal of Manufacturing Systems 70 (2023), S. 417–435

[13] Ye, Z., Kumar, Y. J., Sing, G. O., Song, F. u. Wang, J.: A Comprehensive Survey of Graph Neural Networks for Knowledge Graphs. IEEE Access 10 (2022), S. 75729–75741

[14] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P. u. Bengio, Y.: Graph Attention Networks, 2017

[15] Huet, A., Pinquié, R., Segonds, F. u. Véron, P.: A cognitive design assistant for context-aware computer-aided design. Procedia CIRP 119 (2023), S. 1029–1034

[16] Krahe, C., Kalaidov, M., Doellken, M., Gwosch, T., Kuhnle, A., Lanza, G. u. Matthiesen, S.: AI-Based knowledge extraction for automatic design proposals using design-related patterns. Procedia CIRP 2021

[17] Roj, R.: A comparison of three design tree based search algorithms for the detection of engineering parts constructed with CATIA V5 in large databases. Journal of Computational Design and Engineering 1 (2014) 3,

[18] Bai, J., Luo, H. u. Qin, F.: Design pattern modeling and extraction for CAD models. Advances in Engineering Software 93 (2016), S. 30–43

[19] Mendgen, R.: Methodische Vorgehensweise zur Modellierung in parametrischen und featurebasierten 3D-CAD-Systemen. Zugl.: Darmstadt, Techn. Univ., Diss., 1999.

[20] Bian, S., Grandi, D., Liu, T., Jayaraman, P. K., Willis, K., Sadler, E., Borijin, B., Lu, T., Otis, R., Ho, N. u. Li, B.: HG-CAD: Hierarchical Graph Learning for Material Prediction and Recommendation in Computer-Aided Design. Journal of Computing and Information Science in Engineering 24 (2024)

[21] Mueller, P., Mueller, S. u. Mikelsons, L.: GeoBiked: A Dataset with Geometric Features and Automated Labeling Techniques to Enable Deep Generative Models in Engineering Design, 2024

[22] Xu, X., Peng, W., Cheng, C. u. Willis, K.: Inferring CAD Modeling Sequences Using Zone Graphs, 2021

[23] Buchgeher, G., Gabauer, D., Martinez-Gil, J. u. Ehrlinger, L.: Knowledge Graphs in Manufacturing and Production: A Systematic Literature Review. IEEE Access 9 (2021), S. 55537–55554

[24] Vasantha, G., Purves, D., Quigley, J., Corney, J., Sherlock, A. u. Randika, G.: Common design structures and substitutable feature discovery in CAD databases. Advanced Engineering Informatics 48 (2021),

[25] Cristie, V. u. Joyce, S. C.: Versioning for parametric design exploration process. Autom. in Construction 2021

[26] Kim, J., Kwon, Y., Jo, Y. u. Choi, E.: KG-GPT: A General Framework for Reasoning on Knowledge Graphs Using Large Language Models. Findings of the Association for Computational Linguistics: EMNLP 2023

[27] Kienle, C., Alt, B., Katic, D., Jäkel, R. u. Peters, J., QueryCAD: Grounded Question Answering for CAD Models, 2024

[28] Huang, B., Zhang, S., Huang, R., Li, X. u. Zhang, Y.: An effective retrieval approach of 3D CAD models for macro process reuse. The International Journal of Advanced Manufacturing Technology 102 (2019) 5-8,

[29] Liang, J., Zhang, S., Zhang, Y., Huang, R., u. Zhang, H.: A knowledge graph-based approach to modeling & representation for machining process design intent. Advanced Engineering Informatics 62 (2024)

[30] Jeon, S. M., Lee, J. H., Hahm, G. J. u. Suh, H. W.: Automatic CAD model retrieval based on design documents using semantic processing and rule processing. Computers in Industry 77 (2016), S. 29–47

[31] Holder, K., Zech, A., Ramsaier, M., Stetter, H., Rudolph, S. u. Till, M.: Model-Based Requirements Management in Gear Systems Design Based On Graph-Based Design Languages. Applied Sciences 7 2017

[32] Huang, R., Han, Z., Fei, M., Huang, B. u. Jiang, J.: An effective process design intent inference method of process data via integrating deep learning and grammar parsing. Advanced Engineering Informatics 58 2023

[33] Schweitzer, G. M. J.: Einsatz von künstlicher Intelligenz in der Produktentwicklung auf Basis eines Engineering-Graph im Kontext von Model-based-Systems-Engineering, 2024