Proceedings of the Design Society, Volume 6: DESIGN 2026
https://doi.org/10.1017/pds.2026.10596

A structural framework for generative engineering and design assistance systems development

Timo Köring✉ **and Detlef Gerhard**

Ruhr-Universität Bochum, Germany

✉ timo.koering@rub.de

ABSTRACT: Engineering software is evolving through the integration of artificial intelligence, creating new opportunities for enhanced assistance within product development. This paper proposes a use context model to systematically align and classify the functionalities of Generative Engineering and Design software with respect to the combination of product development phase, the nature of the task, and the level of support provided. Based on this model, a methodological guideline is proposed, offering a structured framework for the development and application of these tools in product development.

KEYWORDS: generative engineering, generative design, computational design methods, design support system, design guidelines

1. Introduction

The ongoing integration of generative artificial intelligence (AI) into engineering development processes and software represents a paradigm shift in the product development process. Automated design generation, the efficient analysis of design parameters, systematic optimization of design variants, and the increasing automation of CAx workflows collectively enable more informed, data-driven development and decision-making processes. Recent advances in the field of machine learning and generative AI have fostered the rise of a variety of different approaches and new software tools, which differ in their methodology, functionality and range of applications. Overall, current trends indicate a discernible transition towards a synthesis process in product development, although the extent and nature of support can vary significantly across tools and applications. This breadth of functional capabilities offers substantial potential for integrating generative technologies into established development processes. At the same time, it opens up opportunities for establishing new processes and methodological frameworks in engineering. However, this diversity also presents notable challenges, particularly with respect to the application characteristics and the methodological integration of these functionalities across heterogeneous specifications and tasks.

In order to characterize and leverage the varying levels of automation support available within such software solutions, this paper addresses two core research questions: 1. *How can the diverse functional components of such software be systematically defined and orchestrated within a coherent, scalable framework?* And 2. *How can these functionalities be practically mapped to different levels of automation support and actions throughout the process of product development?* This typically requires a modular and hierarchically structured architecture, ensuring scalability, transparency, and reproducibility in the development and application of generative engineering tools. To establish this coherent framework between software functionalities and the corresponding levels of support and the action, this paper proposes a use context model and a methodological guideline for the modular and practical integration within product development. The objective is to enable a systematic allocation of the different levels of support, from manual to semi-autonomous synthesis processes to AI agent-based synthesis processes, to the phases and tasks or activities of product development in accordance with VDI 2206 and VDI 2221, and in particular to the software functionalities. The resulting structured reference framework integrates

https://doi.org/10.1017/pds.2026.10596 Published online by Cambridge University Press

generative AI technologies with traditional engineering practices in a manner that ensures scalability, transparency, and is compatible with established development guidelines and environments. This contribution enables researchers and practitioners to systematically design, evaluate, and implement AI-supported product development systems with clearer mappings between capabilities and development stages, overcoming the current fragmentation in tool integration.

2. State of the art

The implementation of a framework that incorporates Generative Engineering and Design (GE&D) streams into the product development process is predicated on two fundamental elements. Firstly, there is a need for a common understanding of GE&D, encompassing its areas of application and functionalities. Secondly, there is a need for a common understanding of the product development process, including its underlying guidelines and potential.

2.1. Generative engineering and design

Parametric design laid the foundation for the automated creation of different variants. By changing predefined parameters, different configurations and characteristics could be rapidly generated in accordance with established rules and constraints. The subsequent transition from explicit geometry generation to the definition of boundary conditions within the framework of computational design synthesis (CDS) was conceptually formalised by Cagan et al. (2005) in a flowchart model. This idea has been used, adopted and further developed in various research projects, methods and systems. Within the commercial sector, the term generative design (GD) has become prevalent, typically characterised by a strong focus on geometric models. Such systems are generally distinguished by a tight integration of CAD and CAE tools, enabling the implicit incorporation of simulation processes within design space exploration, particularly in the context of parameter studies (Park & Dang, 2010). However, the scope and potential applications of CDS and GD extend beyond these aspects. For this reason, the term Generative Engineering and Design (GE&D) is used in this paper as a comprehensive umbrella term for a number of streams such as generative design, design automation, design space exploration and AI-driven design.

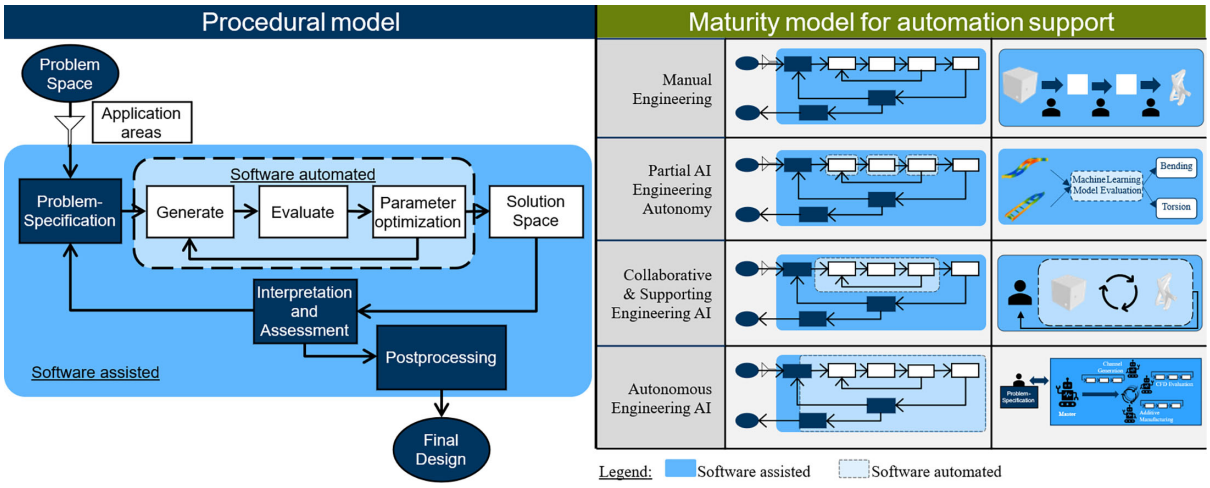


Figure 1. Procedural model and maturity model for the level of support

In order to adequately reflect and represent the flows of operations within GE&D software systems, a procedural model for the application of such software tools was conceptually developed, as shown on the left in Figure 1 (Köring et al., 2025a). This procedural model delineates the sequence of steps from the initial problem space to the final design, thereby enabling a distinction between software-assisted and software-automated sub-steps. The procedure begins with the problem description and proceeds through an iterative cycle of generation, evaluation, and parameter optimisation, resulting in a solution space. This solution space is then interpreted and assessed, and depending on the result, the problem description is refined or subsequent post-processing steps are initiated. The model explicitly highlights the interaction between the user and the internal software process steps, and thus provides a methodological guideline for structured application. Building upon this foundation, the procedural model is expanded

https://doi.org/10.1017/pds.2026.10596 Published online by Cambridge University Press

into a maturity model to illustrate different levels of support functionalities, as shown on the right side in Figure 1. This hierarchical subdivision provides a basis for application within product development that allows AI assistance systems to be systematically located, transitions to be defined and risks to be addressed. Moreover, it clarifies the distribution of roles and responsibilities, thereby supporting informed decision-making regarding tool selection and process integration. This subdivision serves as a roadmap for the development of a methodical specification, the assignment of tasks and functionalities for the development and integration of such systems, which offer active assistance functionality.

In order to facilitate the integration of generative systems into product development processes, numerous static and classification frameworks have been proposed and developed in the literature, most of which primarily address the underlying technical architecture. Nagaraj and Werth (2020) proposed a general yet static framework that encompasses both data-based and physics-based methods to achieve the goal of complete automation and optimization of the design with respect to the functional requirements of the product. Their approach centres on an agent-based architecture employing reinforcement learning and transfer learning as modelling strategies for design solution generation. However, the framework is conceived as a fully automated optimization system rather than an assistance-oriented framework, and consequently, it does not explicitly address the role of human interaction, operational systems, or differentiated forms of support functionalities. Regenwetter et al. (2022), in contrast, make a fundamental contribution to the systematization of generative approaches though a modular overview of deep generative models (DGMs) in engineering design. Modular components such as generative approaches (variational autoencoders, generative adversarial networks and reinforcement learning methods) are organized with tasks such as shape and topology optimization, material design and functional synthesis. However, key challenges remain in the limited availability of high-quality design data sets, the integration of functional performance requirements into generative processes, and the evaluation of the creativity and novelty of generated solutions. This creative potential of generative tools was investigated by Peckham et al. (2024), who found that the generative performance of all specialized engineering tools is largely concentrated within the detailed design phase, whereas only generative AI-based tools exhibit genuinely generative capabilities for the earlier conceptual and design phases. In their literature analysis, Steininger et al. (2025) examined the current state of GenAI in CAD-based product development. Their findings indicate that GenAI is primarily used in design generation (leveraging GANs in combination with AEs), followed by design retrieval (using convolutional neural network architecture) and design reconstruction (with encoder-decoder structures).

Further research has presented more detailed architectures and characterisation frameworks. Müller et al. (2025) presented a schematic representation of an AI system architecture using the concept of retrieval-augmented generation (RAG). Herrmann et al. (2021) proposed a methodical structure for development environments based on the generative parametric design approach (GPDA). According to them, a GPDA development environment consists of three system components, including a synthesis tool, an analysis tool and a generative parametric geometry model, which enables the search for optimal variants within the solution space. Potential and challenges are highlighted in particular in the modelling of generic interfaces and in a specific methodology to support the construction of development environments according to this principle.

2.2. Product development process

The product development process represents the structured transformation of requirements into validated technical solutions. Within the German engineering design methodology, the VDI guideline 2221 and the VDI guideline 2206 provide the conceptual foundation for this process (VDI 2221; VDI/VDE 2206). However, these guidelines not only provide procedural orientation but also a semantic and structural foundation for embedding emerging paradigms such as GE&D. By defining the methodological points at which computational methods can be systematically anchored, they ensure the validity and reproducibility. A comprehensive understanding of their process stages and task structures is therefore essential for mapping (AI-driven) assistance and determining the appropriate level of automation and interaction within next-generation product development systems. This correlation between product development tasks and support from novel tools has already been examined in a number of studies. Berger et al. (2024) investigated the application of generative artificial intelligence in industrial settings to identify opportunities and challenges associated with its adoption across different phases of the product life cycle. Kretzschmar et al. (2024) examined the potential of large language models for the product development process and evaluated their capabilities in a matrix showing which main tasks can be supported. Gerhard et al. (2025) showed examples

https://doi.org/10.1017/pds.2026.10596 Published online by Cambridge University Press

of possible applications of AI methods and processes in product development, structured within the V-model according to VDI 2206. These applications either replace conventional approaches or complement them in the sense of assistance and deliver far better results in terms of quality or quantity, for example by significantly reducing the effort required for certain work steps. In their opinion, GE&D tools support product development activities, particularly from the final phase of idea generation and concept development, continuing through the core phase of development and implementation of solution components, and extending to simulation and validation tasks within the framework of system integration. This also encompasses short-cycle iterations for verification and validation activities, thereby enhancing responsiveness and design robustness through continuous feedback integration.

3. Use context model

While existing works provide modular overviews and architectural models (e.g., modular DGM and RAG architectures), a gap remains in connecting software functionalities explicitly with procedural actions, support levels and standardized development phases. The presented GE&D concepts, processes, systems and implementations, together with the fundamentals of methodical product development and its phases will synthesized into a unified model. This model enables the systematic integration of GE&D tools to address the gap. This integration is formalized in the use context model as a three-dimensional model (Figure 2). The first dimension corresponds to the product development phase in which the process to be supported occurs. The second dimension represents the procedural steps or actions within the procedural model as a part to fulfil the tasks within the phases. The third dimension is the level of support, or automation support, by the software. Figure 2 shows the dimensions with their specific characteristics, which are explained in detail in the subsequent section.

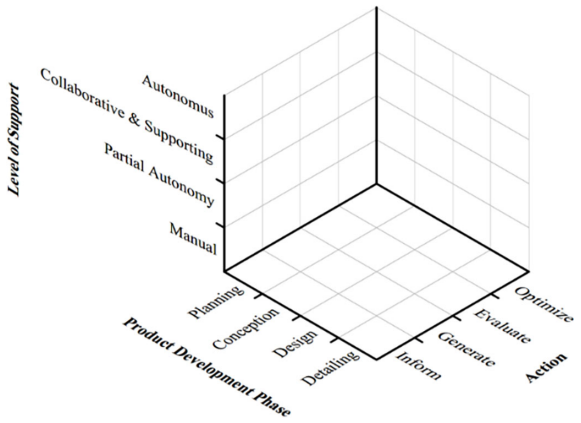


Figure 2. Use context model for GE&D tools

The first dimension, i.e. the product development phase, is divided into planning, conception, design and detailing, which also align with the phases of the VDI 2221. The functionalities of the GE&D tools are specifically related to the later phases of conception, design and detailing, as these represents the main domain of application for GE&D tools (Peckham et al., 2024). Simultaneously, however, they also offer potential for cross-functional and downstream activities as well as for ensuring that design properties and requirements are consistently satisfied. In future research, a more in-depth examination is necessary to analyse and break down the phases into specific tasks, activities and deliverables.

Within the second dimension, drawing upon and extending the process steps of the procedural model, the dimension represents the actions: inform/collect, generate (or execute), evaluate (or assess), and optimize. Informing/collecting refers to all activities aimed at gathering information relevant to specific tasks. Therefore, different sources and qualities of information must be taken into account. This also includes the resulting information for the process step of interpretation and assessment. The generate or execute action involves producing one or a multitude of solutions based on the specified boundary conditions. The objective is to find ideas and solutions that transcend standards and explore alternative solutions. The evaluate/assess action includes "securing product properties", integrating both subjective and objective criteria to enable a comprehensive assessment. The action of optimize focuses on efficiently refining design parameters based on prior evaluations. These actions generalize the steps that are required within the phases to fulfil the tasks, but are also found in other domains like augmented

https://doi.org/10.1017/pds.2026.10596 Published online by Cambridge University Press

reality. In the augmented reality literature, these actions are commonly categorized as: 1) inform, 2) plan, 3) act and 4) control (Siewert et al., 2021). A similar distinction is applied to the types and level of human interaction with automation, encompassing: 1) information acquisition; 2) information analysis; 3) decision and action selection; and 4) action implementation (Parasuraman et al., 2000).
The third dimension is the level of support or automation, which directly influences the intended user group and the detailed implementation of the system. This paper distinguishes between four levels of support: (1) manual engineering, (2) partial AI engineering autonomy, (3) collaborative and supporting engineering AI, and (4) autonomous engineering AI as levels of support. The definition of the level of support has a decisive influence on the scope of the supported action, and therefore this classification plays a particularly critical role. In the case of an autonomous system, for example, the integration of (almost) all actions is logically necessary.

4. Analysis and mapping of functionalities for the use context model

With the use context model, a connection between the phases of product development, the specific procedural actions and the level of support or automation is established. The focus now shifts to the functionalities and categorizations of GE&D software tools to support specific combinations of the use context model. The identification and classification of these functionalities are based on an in-depth literature review, analysis of existing software tools, consideration of user needs, and the examination of prevailing technology trends. The framework developed by Regenwetter et al. (2022) and the classification proposed by Köring et al. (2025a) provide the foundational basis for this analysis. Here, the emphasis is placed on geometry-based implementations, many of which resemble conventional 3D CAD systems, as discussed in Steininger et al. (2025), rather than on all types of input and conversion functionalities, including LLM-based approaches as described by Kretzschmar et al. (2024).

4.1. Analysis of functionalities

The functionalities are divided into different key areas within GE&D and according to their primary purpose. This includes traditional or manual functions, technical and optimization approaches, methods for design space exploration, various types and the integration of machine learning (ML) and AI, as well as capabilities for data analysis and interfacing with information management systems. Certain functionalities may be assigned to more than one category. However, this presentation focuses on the typical components in a systematic manner, without claiming to provide an exhaustive overview.
Traditional functions include functionalities such as sketching, parametric modelling and CAD-based design, which collectively form the foundation for engineering design. These functionalities are grouped together because they are established geometry- and parameter-centric functionalities, requiring mostly direct user interaction. They are complemented by mathematical functions, physical simulations and classic topology optimization. However, their applicability is limited in complex, multidimensional design spaces, as exploratory and optimization processes can only be executed manually to a restricted extent. Building on traditional approaches, new computational methods emerged within the category of *technical and optimization approaches*, enabling partial automation of design processes. Functionalities are assigned to this category when they introduce algorithmic goal orientation while still relying on explicitly defined rules or models. These methods include voxel-based modelling, as well as goal-oriented approaches such as data-driven, field-driven or algorithm-based design and design automation in general. These methods shift the scope of functions towards goal-based system optimization, whereby design parameters are specifically adapted to performance indicators or functional requirements. These methods are designed to facilitate the efficient investigation of large design and solution spaces. They also support engineering decision-making processes through algorithmic and systematic approaches. The *design space exploration* is a central component of GE&D processes. This category includes functionalities whose primary purpose is the structured investigation of multidimensional design spaces, rather than the generation of individual design instances. This is utilized through methods such as Design of Experiments, sensitivity analyses, trade-off analyses and optimization techniques to investigate the multidimensional design space in a structured manner. Modern approaches further incorporate graph-based models and ML, thereby extending the methodology from a purely deterministic framework to an exploratory, data-driven process capable of autonomously discovering and evaluating novel solutions. This development enables the integration of Design by Objective to automatically balance complex conflicting objectives and generate optimal design variants supported by data. In particular, the

https://doi.org/10.1017/pds.2026.10596 Published online by Cambridge University Press

integration of evolutionary algorithms, *machine learning and artificial intelligence* plays a crucial role in solution space exploration and optimization, and is considered a separate category when functionalities rely on learned models, adaptive behaviour, or probabilistic inference rather than deterministic rules. This role in generative engineering encompasses a wide range of applications: from generative design and personalized design proposals to automated decision support. In particular, neural networks, large language models and generative adversarial networks (GANs) enable a semantic and context-related interpretation of design requirements. Consequently, ML functions serve as an active partner in the creative design process, recognizing patterns, generating suggestions and being guided via prompt engineering in natural language. *Data analysis*, encompassing pattern recognition, classification, anomaly detection and support for data-driven decisions, represents another category. Functionalities are assigned here when their primary contribution lies in extracting, structuring, or interpreting information rather than directly modifying design geometry. ML-based systems can learn from historical design data and make predictions regarding optimal design parameters. A robust data infrastructure for the *information management* constitutes an essential foundation for both AI integration and design systems in general. This includes connections to office and database systems, script processing (e.g. via JSON), web connections and graphical result processing. These components ensure that generative systems can be seamlessly embedded into existing business and development workflows and that results are efficiently processed and visualized. Another important aspect, is the GUI or, more generally, the interaction interface. This is crucial for visualization and interaction or manipulation. Typical examples include CAD-based environments, low-code environments, and web-based or proprietary environments.

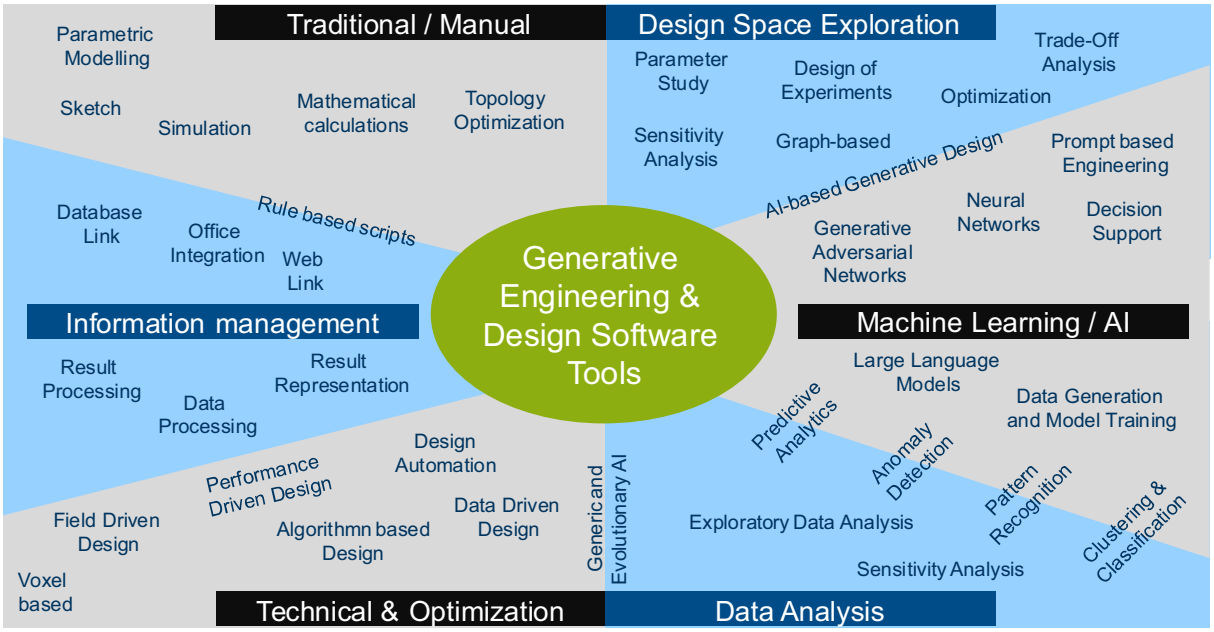


Figure 3. Overview of categories and typical functionalities of GE&D-tools

Figure 3 shows the categories with some representative examples, although this list is not intended to be a comprehensive inventory of available functionalities or methods. Other categories not discussed or shown here may include the manufacturing characteristics, particularly in the areas of additive manufacturing, general manufacturing boundary conditions, and assemblies or aspects of product life cycle and sustainability support. Within some categories, certain areas are only summarized, so that, for example, the database link in information management is representative for the connections to material databases, PLM/ERP and others. Similarly, machine learning and AI are included, but not their associated management.

4.2. Mapping of functionalities

Once the functionalities have been identified and systematically summarized within GE&D-tools, they can be assigned to the use context model, which enables the practical integration and application within product development systems. Since different functionalities and, in particular, different actions may be relevant across multiple product development phases, the assignment is defined according to which

https://doi.org/10.1017/pds.2026.10596 Published online by Cambridge University Press

functionalities support or facilitate procedural actions and specific levels of support, rather than according to chronological phases of the product development process.
The systematization shown in Figure 4 categorizes the functionalities of GE&D-tools along the four procedural ***actions***. The allocation of functionalities to these actions is guided by two mapping criteria: the primary intent of the functionality, and the typical outcome for the engineering design process. The goal of the systematization is not to enforce strict boundaries, but to make visible how diverse functionalities collectively support the full design process. The graphic illustrates that the functionalities permeate all actions and thus offer holistic support. Overall, it shows that manual, traditional, advanced and AI/ML-based approaches can be applied in all actions, indicating that the procedural actions are independent of specific implementation technologies or levels of automation.

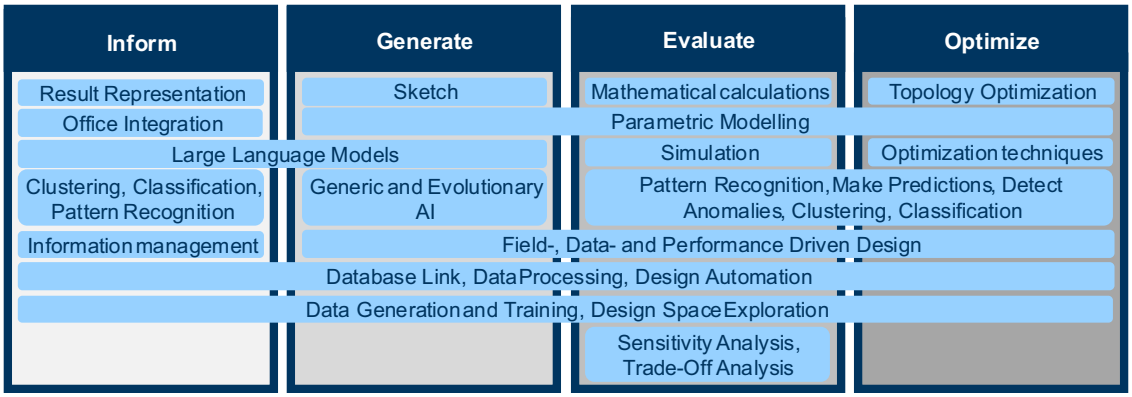


Figure 4. The procedural actions of the use context model mapped to GE&D-tools functionalities

Figure 5 shows the systematic integration of the functionalities presented above to enable different ***levels of support***. The assignment of functionalities does not aim to define or evaluate the autonomy levels themselves, but to clarify what specific functionalities are typically embedded in systems for different levels of support. These functionalities were sorted based on two main criteria: the extent of human intervention required and, conversely, the scope and degree of the automation of the functionality. One focus is on the definition of the GUI, as interaction paradigms reflect how functionalities are accessed and orchestrated at different support levels. While manual engineering primarily focuses on the CAD environment, increasing assistance also brings low-code and web-based systems to the forefront. In general, it can be seen that the number of functionalities increases with increasing levels of support, reflecting a cumulative integration of automated and data-driven capabilities rather than a replacement of lower-level functionalities.

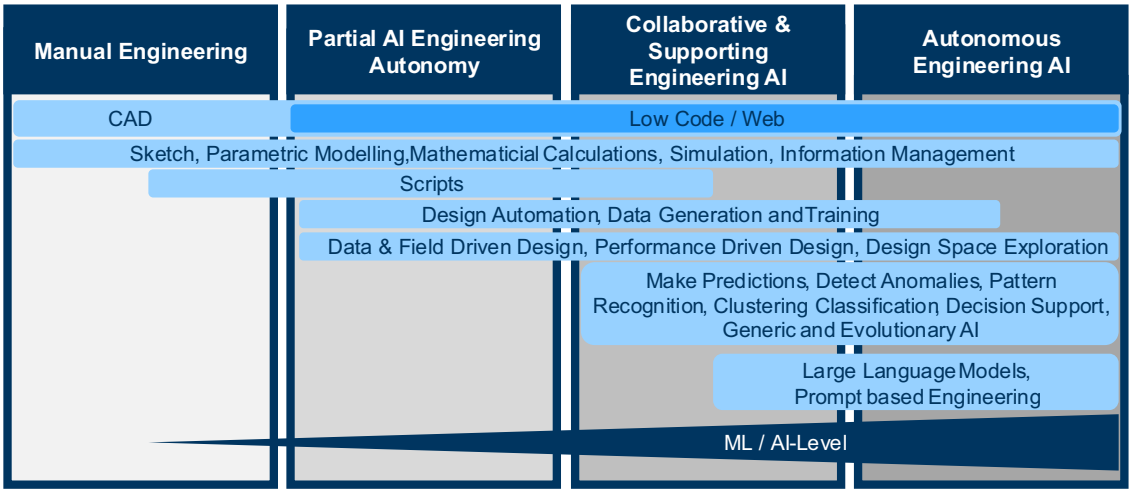


Figure 5. The level of support of the use context model mapped to GE&D-tool functionalities

5. Assistance system implementation guideline

The previous findings serve as a foundation for the development of a generative engineering and design assistance system (GE&D-AS) guideline. The definition of the use context model and the corresponding

https://doi.org/10.1017/pds.2026.10596 Published online by Cambridge University Press

allocation of the GE&D software functionalities to the actions and levels of support therefore build the conceptual foundation for developing a methodological guideline for system development. This development guideline is structured into three overarching phases: *conceptual phase*, *technical implementation*, and *training and evaluation*. However, this paper concentrates primarily on the conceptual phase.

5.1. Conceptional phase

The conceptual phase constitutes the basis for the GE&D-AS, as it involves the definition of the intended level of support functionality and the functional objectives. These two parameters significantly determine the subsequent system architecture, the interaction logic between humans and the system, and the technical and organizational embedding within existing development processes. The aim of this phase is to create a technical, methodological and organizational basis for the subsequent implementation and evaluation phases. The design phase begins with a precise definition of the intended application context and the overarching system objectives. The design of the interaction between humans and the system is the central element of this concept and phase. Its foundation lies in the classification within the support continuum, considering the action to be supported, and the positioning within the product development phase. The desired level of support or autonomy has direct implications for the actions to be implemented, system interaction, the GUI, and the integration of AI/ML and API applications. CAD-based, low-code and web-based environments are considered to be primary GUI paradigms, with their selection depending on the application scenario, integration context, and the target user group. The GUI also defines the interaction model specification between the user and the system, including the interaction style, whether visual or text-based. Furthermore, based on the objective definition and the level of support or autonomy, the functional components are subsequently specified and selected as described in the previous section, considering Figure 4 and 5. Analogous to the analyzed functionalities, these can cover the following areas: design and simulation functions (e.g. geometry modelling, topology optimization, field simulations), data and AI components (e.g. ML, generative models, classification, clustering, anomaly detection), design space exploration functionalities (DoE, sensitivity analysis, trade-off optimization, multi-objective optimization), decision-making and feedback systems (result interpretation, visualization), data and system integration (databases, API interfaces). The selection and orchestration of these modules depends not only on the functional requirements, but also on technical feasibility, interoperability and potential synergies with existing engineering systems. A robust and scalable data architecture serves as the infrastructural basis for ML and AI-supported decision-making processes, where applicable. Key aspects include the definition of relevant data sources and interfaces (e.g. CAD, simulation) and the management of training and validation data. In addition, feedback loops must be implemented to enable continuous learning and system adjustments during operation.

5.2. Technical implementation, training and evaluation

The subsequent technical implementation builds directly upon the concept and specification defined in the conceptual phase. A modular software architecture ensures flexible expandability and seamless interoperability among the system components. Standardized API interfaces are used to integrate external data sources, simulation environments and CAD/CAE systems. Depending on the nature of the support functionality, the integration of common ML and AI frameworks is necessary to support generative and adaptive processes. These can be used simultaneously to generate and vary technical designs that undergo geometric or topological optimization processes and to automatically generate design proposals. The generated solutions can be evaluated using coupled simulation methods, such as the finite element method (FEM) or computational fluid dynamics (CFD).

The subsequent training and evaluation phase serves to validate and verify the developed architecture and functional modules. First, a review is carried out using reference data sets and established benchmarks to test functionality. When integrating AI models, the model quality is also evaluated in terms of design quality, computing power and robustness. Specific criteria for validation and verification must be determined individually. An iterative training process ensures continuous performance improvement of the system. User feedback from real projects and new data sets are integrated into the learning process and system architecture, enabling adaptive improvement of both algorithms and user interaction. This results in a dynamic, optimizing assistance system that achieves increasingly higher levels of automation and support in development work as the database grows.

https://doi.org/10.1017/pds.2026.10596 Published online by Cambridge University Press

5.3. Practical example

To demonstrate practical application, the example of the partial AI engineering automation from Figure 1, based on Köring et al. (2025b), is used. The overarching objective is to create a partial AI engineering automation assistant for the automatic evaluation of modal frequencies in the context of truck ladder frames. Following the use context model, this step deals with the action "evaluate" in the detailing phase of product development. A visual low-code environment using ML and API applications is used to create and implement this assistance. This environment orchestrates the necessary functional components, which originate from design and simulation, as well as from the field of machine learning. The "evaluate" functionality is integrated with parametric modelling functions, FE-simulation options and an ML model for classification, as illustrated in Figures 4 and 5. In addition, a system for representing the results is incorporated as the GUI. The data basis is provided by parameter tables that are accessed via an API and scripts. With the help of the model, whose exact technical implementation, training and evaluation are detailed in the mentioned publication, partial AI engineering automation. This is achieved through the implementation guideline and the systematic assignment of functionalities to procedural actions and levels of support within product development.

6. Conclusion and outlook

Advancements in technical systems for development and design are driven by the integration of generative AI, novel forms of collaboration and functionalities. This progression is marked by a rapid evolution in the field. This transition to data- and AI-based adaptive systems that partially or fully automate the design, simulation and optimization of technical products creates a need for a methodological basis for the selection, design and implementation of these new GE&D assistance systems. This methodological basis must take into account different levels of support or autonomy, actions, functional requirements and user interfaces.

The foundation for this is a use context model, which aligns the phases of the product development process with the action to be performed, and the levels of support in order to facilitate the effective utilization of the functionalities of GE&D software tools within product development. As the procedural actions to be performed within each phase of the product development process are similar, particular emphasis is placed on the assignment of software functionalities for the level of support and the action. This classification or systematic assignment of existing software functionalities to the various tasks or actions, and a differentiated consideration of the different levels of support within the product development processes, serve as the basis for the three-part guideline for the successive development of such systems.

The conceptual phase of this guideline builds the structuring framework for deriving the functional, methodological and data-based components of the GE&D-AS. By systematically considering the level of support or autonomy, interaction design and data architecture, a coherent concept is created that methodically supports the transition from traditional, manual development processes to collaborative or autonomous AI-supported design systems. In the technical implementation, training and evaluation phases, the developed concept is then implemented and assessed for its applicability.

In subsequent work, it is necessary to characterize and break down specific tasks and activities within the product development phases in a more detailed manner. This will enable a better alignment of functionalities with specific phases. Also, the functionalities of GE&D software tools can be further explored and more categories can be integrated. This enables improved consolidation of functionalities in order to define possible combinations and characteristics for specific tasks. However, the focus should be on more technical implementation, training and evaluation in order to enable practical use and validation through more practical examples and applications of the methodology. In this context, specific validation and verification criteria should also be defined in a generally applicable manner. In conclusion, the developed use context model, together with the alignment of functionalities can serve as a basis for the development of a modular system and decision tree that simplifies and guides the systematic integration of GE&D-tools in the product development process.

References

Berger, E., Braun, F., Mehlstaubl, J., & Paetzold-Byhain, K. (2024). Applications of Generative AI in the Product Development Process: A Scoping Review. In *DS 133: Proceedings of the 35th Symposium Design for X (DFX2024)* (pp. 55–64). The Design Society. https://doi.org/10.35199/dfx2024.06

https://doi.org/10.1017/pds.2026.10596 Published online by Cambridge University Press

Cagan, J., Campbell, M. I., Finger, S., & Tomiyama, T. (2005). A Framework for Computational Design Synthesis: Model and Applications. *Journal of Computing and Information Science in Engineering*, 5(3), 171–181. https://doi.org/10.1115/1.2013289

Gerhard, D., Neges, M., Köring, T., & Uzair, M. (2025). Optimierung von Entwicklungsprozessen durch KI-gestütztes Generatives Engineering und Design. *Zeitschrift Für Wirtschaftlichen Fabrikbetrieb*, 120(s1), 70–75. https://doi.org/10.1515/zwf-2024-0140

Herrmann, K., Altun, O., Wolniak, P., Mozgova, I., & Lachmayer, R. (2021). Methodischer Aufbau von Entwicklungsumgebungen nach dem Generative Parametric Design Approach. In *DS 111: Proceedings of the 32nd Symposium Design for X*. The Design Society. https://doi.org/10.35199/dfx2021.14

Köring, T., Gerhard, D., & Neges, M. (2025a). A Procedure Model for Systematic Application of Generative Engineering and Design Software Tools. In P. Sureephong, C. Danjou, & A. Bouras (Eds.), IFIP Advances in Information and Communication Technology: Vol. 740. *Product Lifecycle Management. Integrating Digital Technologies for Sustainability and Innovation: 21st IFIP WG 5.1 International Conference, PLM 2024, Bangkok, Thailand, July 7–10, 2024, Revised Selected Papers, Part I* (1st ed. 2025, Vol. 740, pp. 43–57). Imprint: Springer. https://doi.org/10.1007/978-3-031-93319-6_4

Köring, T., Gerhard, D., Neges, M., Seddik, F., Kömm, L., & Paetzold-Byhain, K. (2025b). A machine learning approach towards automated classification of modal analysis results. *Proceedings of the Design Society*, 5, 159–168. https://doi.org/10.1017/pds.2025.10030

Kretzschmar, M., Dammann, M. P., Schwoch, S., Berger, E., Saske, B., & Paetzold-Byhain, K. (2024). Key Concepts, Potentials and Obstacles for the Implementation of Large Language Models in Product Development. In *Entwerfen Entwickeln Erleben 2024 : Menschen, Technik und Methoden in Produktentwicklung und Design* (pp. 381–393). Technische Universität Dresden. https://doi.org/10.25368/2024.EEE.031

Müller, B., Roth, D., & Kreimeyer, M. (2025). KI-Adaption in der Produktentwicklung. *Zeitschrift Für Wirtschaftlichen Fabrikbetrieb*, 120(s1), 44–50. https://doi.org/10.1515/zwf-2024-0158

Nagaraj, D., & Werth, D. (2020). Towards a Generalized System for Generative Engineering. In N. Petkov, N. Strisciuglio, & C. M. Travieso-González (Eds.), *Proceedings of the 3rd International Conference on Applications of Intelligent Systems* (pp. 1–6). ACM. https://doi.org/10.1145/3378184.3378227

Parasuraman, R., Sheridan, T. B., & Wickens, C. D. (2000). A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics. Part A, Systems and Humans : A Publication of the IEEE Systems, Man, and Cybernetics Society*, 30(3), 286–297. https://doi.org/10.1109/3468.844354

Park, H.-S., & Dang, X.-P. (2010). Structural optimization based on CAD–CAE integration and metamodeling techniques. *Computer-Aided Design*(42), 889–902. http://doi.org/10.1016/j.cad.2010.06.003

Peckham, O. R., Hicks, B., & Goudswaard, M. (2024). Are generative design tools creative? A characterisation of tools throughout the design process. *Proceedings of the Design Society*, 4, 683–692. https://doi.org/10.1017/pds.2024.71

Regenwetter, L., Nobari, A. H., & Ahmed, F. (2022). Deep Generative Models in Engineering Design: A Review. *Journal of Mechanical Design*, 144(7). https://doi.org/10.1115/1.4053859

Siewert, J. L., Neges, M., & Gerhard, D. (2021). Ein Klassifizierungssystem für Industrielle Augmented Reality Anwendungen. In R. H. Stelzer & J. Krzywinski (Eds.), *Reihe Technisches Design: Vol. 15, Entwerfen Entwickeln Erleben in Produktentwicklung und Design 2021* (pp. 401–416). TUDpress. https://doi.org/10.25368/2021.37

Steininger, S., Zhao, J., & Fottner, J. (2025). Enhancing computer-aided design with deep learning frameworks: a literature review. *Proceedings of the Design Society*, 5, 1515–1524. https://doi.org/10.1017/pds.2025.10165

VDI 2221. *Design of technical products and systems Model of product design (VDI-Richtlinie 2221:2019-11 Blatt 1)*. Beuth Verlag.

VDI/VDE 2206. *Development of mechatronic and cyber-physical systems (VDI/VDE-Richtlinie 2206:2021-11)*. Beuth Verlag.

https://doi.org/10.1017/pds.2026.10596 Published online by Cambridge University Press