

Theoretical foundation to model design decisions for hybrid (Human / AI) decision support systems

Eric Kempkens ^{1*}, Anvit Upadhyay¹, Zhongtian Wang¹, Rainer Stark¹

¹Technische Universität Berlin, Institut für Werkzeugmaschinen und Fabrikbetrieb, 10587 Berlin

*Corresponding author. E-Mail address: eric.kempkens@tu-berlin.de

Abstract: This contribution presents a theoretical foundation for modelling granular, incremental activities in the product development process (PDP) for utilization in the conception and realization of hybrid (Human/AI) decision support systems. This approach differs from other modeling approaches of the PDP by modeling granular activities while simultaneously considering the external factors influencing these activities. This holistic approach can be utilized to model singular decision cases within the PDP, which can be organized as a graph to capture the interdependencies between decisions within PDP activities. The resulting models can then be used as a foundation for context engineering to be utilized in the development of agentic AI systems.

The theoretical foundations proposed in this paper deliver a specialized modeling foundation to capture the granular, incremental activities of the design process. This has the goal of enabling the conceptualization of agentic AI systems, which can aid designers in their dynamic design process. The proposed model is validated through a study using a student project composed of a PDP, showing that the model can be applied onto real-world design processes to model design activities with the goal of conceptualizing AI enabled automation.

Keywords:

Hybrid Decision Support, Product Design, Agentic AI, Mathematical Modelling

1 Introduction

Hybrid Decision Support Systems (HDSS) for Product Design are systems that implement Artificial Intelligence (AI) automation in a way which does not diminish the role of human expertise, combining their strengths. HDSS systems compute solutions towards decision cases (DC) using AI methodologies, which are then evaluated, further processed or rejected by human operators. HDSS are aimed at increasing the time efficiency of workflows by aiding designers in finding solutions for DC faster [1].

The product design process (PDP) is a highly dynamic and non-linear process often resolved by implicit designer knowledge and design heuristics [2] and consists of many granular activities which add complexity in steps onto an artifact [3]. The granular decision cases (GDC) within the PDP inform and influence each other and contribute toward creating the final artifacts. A decision case is defined here as a situation in which attributes or outcomes must be decided on to move onto subsequent steps within an activity. Within the Detailed Planning process (DPP) of the PDP, a lot of GDCs can be identified which contribute to the final artifact. For example, in CAD modeling, examples of GDCs would be the form of an initial two-dimensional sketch, how it is subsequently extruded or rounded and how the geometries as well as functional elements are realized. A DC is situated within a larger design activity (DA). A DA here is defined as a process within the DPP of the PDP which has a clearly defined starting and ending point for each iteration of the activity, like the modelling of complete products or product components using CAD.

With recent breakthroughs in the field of AI, in terms of model capabilities as well as the rising availability of AI computing services and frameworks, there has been a push to apply these technical opportunities to the PDP [4-5]. HDSS are one way to enable a specialized AI integration into the PDP. HDSS systems aim to ease the navigation of complex product requirements, integrate usage-related information, offer validation and generate explainable solutions, among other important attributes. This is done by leveraging extreme data for AI methods, which is heterogeneous in nature and derived from many different sources within the PDP as well as the product life cycle [1]. The non-linearity of the PDP presents a challenge for the implementation of HDSS. In their role as assistance systems, they should ideally be implemented in a way which embeds itself into existing PDP environments and workflows without friction after initial adoption. One approach is to develop AI based full solvers, which offer solutions for the complete activity or a sub-activity composed of multiple DC generated by AI methods. This approach has the benefit of potentially speeding up PDP activities by providing possible solutions for substantial segments or complete DA. This can be done, for example, by using Generative Adversarial Networks to create product geometries [6]. But for this to be most effective, their results must be of high quality and satisfy all applicable constraints. If this is not the case, the designer might need to manually improve the solution generated, which could negatively impact or negate the net efficiency gains. Another way of implementing AI based support systems is to dynamically offer decision support for granular activities. This enables a form of HDSS, where the designer has more control within the individual DCs which culminate in the final artifact. Saving time within GDCs enables time savings at each step, potentially adding up to large time savings for the complete DA. Taking a granular approach in generating solutions within the DPP could aid in assuring the quality of generated results, lessening the cumulative time needed for manual adaptations.

In this paper, we propose a theoretical foundation to describe and model the PDP as a network of interconnected GDCs which can be modeled as a mathematical graph. Mathematical abstraction helps in generalizing information flows within the PDP, making it easier to standardize information objects which can be represented as sets of encoded information. From there, singular nodes and associated edges can be used for context engineering, meaning the fine-tuning and careful curation of inputs representing context for transformer-based AI processing, for HDSS implementations. This framework can be used to conceptualize the data pipeline needed for granular HDSS within the PDP using Natural Language Processing (NLP).

2 Research Problem

The design processes within the PDP are often complex and non-linear, especially during the DPP phase. This can include the modeling of product geometries and the placement of system elements, one example being the placement of electrical components on printed circuit boards. But it could also describe other DPP processes, like the placement of elements within the factory planning process. To develop intelligent HDSS which can aid designers dynamically within the DPP of the PDP, there is a need to model their granular workflow. This level of modeling the PDP is classified as the micro category of PDP modeling [7], which focuses on granular process steps and their associated contexts.

When modeling the PDP from the micro perspective with the intention to analyze and conceptualize possible implementations of HDSS, a challenge arises which is not specifically addressed by traditional PDP modeling approaches. To leverage reasoning capabilities of transformer models as proposed in [8], where semantic elaboration on numerical problems is seen to enhance the quality of generated solutions, there is a shift in paradigm required. Away from only utilizing rigid, coded formulation towards natural language descriptions when modeling the design process with the goal of HDSS automation. Contemporary PDP modeling approaches are not dedicated to a granular approach while also capturing

the context and data in natural language along with numerical parameters needed for AI automation, as seen in the comprehensive study by Wynn & Clarkson as well as the study by Yin and Zhang [7,9]. One classical approach is the Task-Episode Accumulation Model (TEA) [10], which semantically describes the PDP as a series of episodes building upon each other, which captures the evolution of the design state via goal-oriented phases with 6 operators, but does not document GDCs as defined here. The Design History System (DHS) [11] features a schematic to document GDCs but argues against (unstructured) textual semantic representations in favor of more rigid coded descriptions of arithmetic operations and attributes, representing pre-transformer paradigms of utilizing coding language abstractions in favor of semantic elaborations. However, to utilize transformer capabilities for HDSS within the DPP, a holistic perspective is needed which captures semantic descriptions as well as numerical parametric of GDC. This includes the objectives of GDC, as well as the specific context, the objectives, the constraints, and the environment of the DPP as well as relevant aspects of the wider PDP. HDSS approaches would benefit from leveraging semantic context and operation descriptions for generating decision results for designer consideration. The Engineering Operating Systems (EOS) model [12] delivers this holistic perspective for the PDP but offers no framework to model the GDC of DPPs.

Another shift in paradigm with automation based on modern sub-symbolic AI (SSAI) methodologies is the ability to adapt to different input formats when computing solutions to inquiries. This contrasts with the mostly fixed input formats of symbolic AI automation [13]. To describe DAs with the goal of applying SSAI automation, new ways of describing these activities on a granular level, while also considering external flows of information of non-fixed formats can be expected to be of great benefit. This could be used to conceptualize and develop effective AI agents (AIA) or agentic AI (AAI) using retrieval-augmented generation (RAG), which can leverage diverse domain-specific external knowledge bases for foundational Large Language Models (LLM) without a dedicated, resource-intensive training process [14]. A methodology for mathematical modeling of GDC can aid in standardizing information objects and flows for context engineering, which is an important foundation, as data quality is critical for building effective AI applications [15]. This can be applied to analyze DA to conceptualize effective transformer/LLM-based AIA/AAI architectures, as described in [16-18], which are specialized for DAs and can be adapted to different design environments. Since AI methodologies like RAG are novel concepts, classical PDP modelling approaches do not include avenues for their specific considerations.

3 Research Question

The research question derived from the problem statement is as follows: *How can a theoretical foundation be laid which offers a framework to model GDC during the DPP of the PDP while also considering the holistic insights of the EOS model?* A synthesis of both these aspects is needed, as the implementation of dynamic HDSS requires modeling granular activities while also considering the wider flow of information within the PDP to conceptualize effective, quality HDSS tools. This contribution aims to fill this gap by providing this theoretical foundation to model DA in a way which can aid the conceptualization and development of dynamic HDSS, which offer solutions toward building capable AIAs for singular DC. Standardized mathematical modeling helps with standardizing flows of information and adaptability of subsequent architectures.

4 Methodology

To construct the theoretical framework, mathematical modeling is used to describe how different information objects consisting of encoded information, represented as sets, influence DPP activities. For this, a large emphasis is placed on describing mathematical objects using set theory notation. This allows the modeling of information flows within DA of the DPP of the PDP in a way which is both fitting for

describing human design heuristics as well as non-specific information objects for automation using AI methodologies within this context. Describing flows of information using set theory notation has the benefit of being easily adaptable as well as lending itself well as representations of input sets for AI models. One further benefit is the capture of human workflows, which should help in conceptualizing and developing AIA/AAI tools which synergize with the natural process. To validate the framework laid out within this contribution, a case study is conducted. This case study was carried out using data recorded during a student project involving a PDP, in which all principal components found within the mathematical framework can be recorded, observed, and modeled.

5 Theoretical Framework

For the theoretical foundation for modelling granular decision-making process during the DPP of the PDP we apply mathematical abstraction to organize this process as a graph of individual decisions. Within this framework mathematical objects represent vectors containing encoded information relevant to their identification. A union ‘ $\cup$ ’ shall describe a broader, non-specific combination process, not just strictly a union of sets. This is done to represent the combination of information for possible use as input sets of information for AI models, which may shed unneeded or redundant information for rationalizing set size. Further ‘ $\rightarrow$ ’ describes a function or a system of functions of ambiguous nature and dynamics, taking various input objects and through a process of synthesis and manipulation arriving at a specified result object. They are standing in for the processing by human operators, AI based automation or other applicable black-box/grey-box methods.

The resulting artifact A of a design process P can be understood as a cumulation of a variable number of subprocesses $p \in P$. This artifact A can take the shape of a CAD model, a sketch or even a physical model. Thus, the cumulation of a DA can be expressed as $\forall p \in P, p \rightarrow A$, meaning that every subprocess p contributes toward the creation of a resulting Artifact A . Every subprocess p uses the objects c, d to arrive at a result r through $f(c, d) = r$. In this c represents a union of three components: environment e , documentation b and previous results r_r . Context c shall be analog to contextual information human designers use to work on and finalize a given subprocess p . This should include relevant information about the environment of the design process e , all the relevant documents b relating to the specifications of the final artifact A and possible subartifacts and finally information results $r_r \in R$ of other processes. We define $e \cup b \cup r_r \rightarrow c$, meaning that a union of all the information e, b, r_r shall result in the context c . The decision d shall be the result of a synthesis involving the specifications s of solving an objective o . Specifications s shall describe all the parameters and constraints relevant to solve objective o . Objective o shall describe the objective of the relevant process p . This can include different DAs tied to a concrete decision, like the manipulation of a polygon, the placement of a component, etc. in accordance with d . The specifications s are the result of a synthesis between the context c and objective o , thus $c \cup o \rightarrow s$, while a decision $s \cup o \rightarrow d$. This means that the logic of synthesizing specific information of a decision d is a non-linear process regarding o . Result r describes the results of a decision d made within the context of c . It has no need for uniformity and is able to describe the result of many different operations within a design process P . It is the result of a function $f(c, d)$, which can both represent the actions of a designer, the calculation processes of an AI method to generate the result of a decision case or similar processes. Since each object p results in a decision r and can each be influenced via their associated context c which is informed by results of previous processes r_r from which c is partially derived, P can be organized as a graph $g_P \subseteq P \times P$. Figure 1 shows a visual representation of a singular node p of a process Graph g_P .

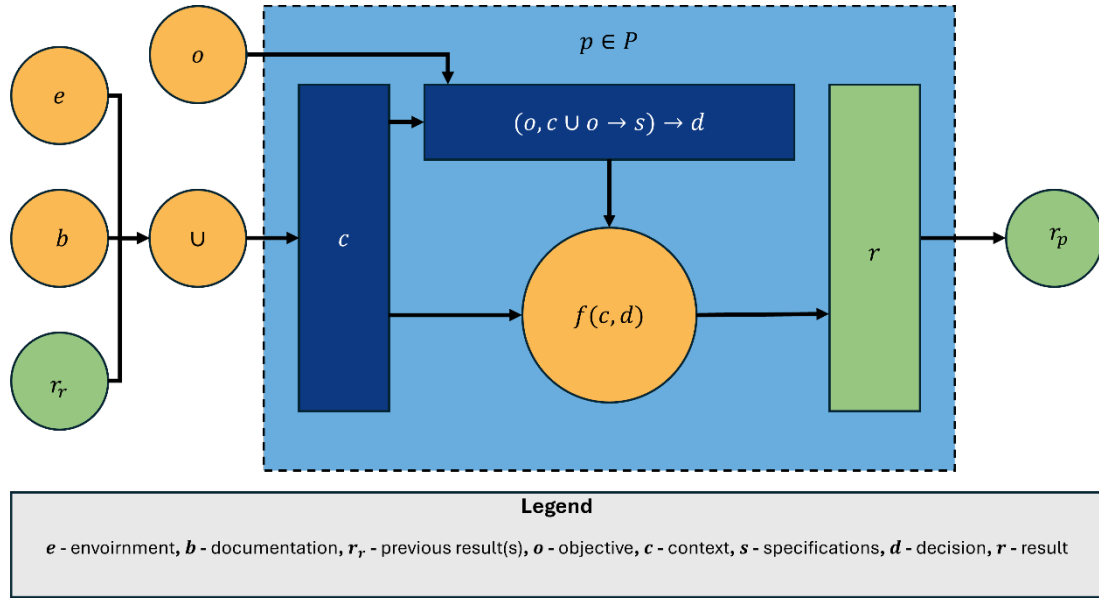


Figure 1: Visualization of the modelling approach for a subprocess $p \in P$

6 Case Study

To validate this approach, a case study using a student engineering project from an engineering module using the proposed framework has been carried out. Within the project work, 3 student groups are working towards delivering a concept and a working prototype of the same general product concept. The aim was to create a staircase ascender capable of transporting heavy loads like packages and groceries upstairs. From there, the student groups had to develop a product concept from scratch, including product specifications, technical solutions, as well as modelling their concept as a CAD model using Autodesk Fusion 360. The individual design decision cases of the modelling process needed to be resolved based on function, manufacturing as well as costs, leading to multifactorial considerations at each step where necessary during the modeling process.

The case study used the Fusion 360 projects of the student groups for applying the framework, as Fusion 360 project files preserve the causality between different functions of a modelling process within a timeline, making it possible to see how different processes influenced and added onto each other to arrive at the finished artifact. The theoretical modeling approach has been applied to the creation process of a chassis concept for a staircase ascender for one of the groups. The DDP process to model the chassis concept offers a manageable scope of GDCs while also being intricate enough to show that this model can be applied to real-world DPP processes.

Environment e was modelled via text as follows:

Modelling within Fusion 360, Development of Model Concept for Graded Student Project, Eventual Goal of Functional physical Prototype. Designer is part of a student team; they are the specialist for realizing concept design within CAD. Specifications of the technical solution are made as a team effort.

Documents b include the project documents created to specify the technical solution concept, technical specifications, as well as fixed physical specifications. The technical solution concept was delivered in the form of a one-page profile, consisting of 236 words and one figure visualizing the concept and features the following sections: *Abstract, Description, Team Specifics, Key Features of Prototype, Key Features of future Product, SWOT-Analysis*. The technical specifications are a written description, including all planned key components, consisting of 689 words. It features among others the following sections: *Concept, Motor Type and Chassis Material*. Additionally, technical specifications are further

defined as table of specifications. This table consists of 33 positions, detailing key features, and where possible includes metrics for laying down feature target values. For example, for the drive, the type of motor, the nominal performance targets of the motor at 13-40 Nm and target battery values at 12-24 V are laid down. All these documents combine to form the corpus of documentation on which the modeling processes of this student team are based on.

The model is applied onto the DPP of the chassis concept as P , using the Fusion 360 parametrics. This resulted in a protocol consisting of 30 individual subprocesses $p \in P$ and associated decision cases. Individual subprocesses were defined using the Fusion 360 timeline, splitting some consolidated steps from the parametric in favor of discrete sketches where a mirroring has been obscured. This better represents the underlying flow of decision cases. Function $f(c, d)$ has been recorded as the principal function used by the designer within the Fusion 360 environment. The names of the functions have been generalized. The protocol of the decision cases has been recorded in tabular form, with columns containing an index as well as $e, b, r_r, o, c, s, d, f, r$ for each identified $p \in P$. Table 1 shows an excerpt from this protocol, containing index of p as well as r_r, o, s, d, f, r for 8 of the 30 decision cases. Table 2 shows the corresponding contexts c for the p in Table 1. For this representation e and b are not represented here, as they do not differ regarding the earlier descriptions. Numerical parameters are represented using '...'. Visual representations of each step can be found in figure 3.

Table 1: Decision case protocol excerpt 1

$p \in P$	r_r	o	s	d	f	r
6	1	Sketch feet of chassis leg in + X + Z Plane	Shape, Size, Location	Shape: [quadrangle, chamfer in -Z, -X direction], Size: ... , Chamfer: ... , Location: ...	draw quadrangle, add chamfer	Sketch of Foot
7	6	Mirror feet sketch in - X, + Z Plane	Direction, Offset	Mirror +X Plane of Foot Sketch, Offset: ...	Mirror	Mirrored foot in - X, + Z plane
8	1,2,6,7	Sketch attachment point of chassis legs in the + X + Z Plane	Shape, Size, Location	Shape: quadrangle, Size: ... , Geometric_Center: + ... Z, + ... X	draw quadrangle	Attachment point sketch of chassis leg in the + X + Z Plane
9	8	Mirror attachment point sketch in - X, + Z Plane	Direction, Offset	Mirror + X Plane of attachment point sketch, Offset: ...	Mirror	Mirrored attachment point sketch of chassis leg in the + X + Z Plane
10	6,7	Extrude feet sketches	Type, Direction, Location	Type: Push, Direction: + Y, Depth: ... , Location: + ... Y	Push	Extrusion of the feet sketches
11	8,9	Extrude attachment points	Type, Direction, Location	Type: Push, Direction: + Y, Depth: ... , Location: + ... Y	Push	Extrusion of the attachment point sketches
12	1,2,7,10	Add custom reference plane	Location, Direction	Location: - ... X, Direction: Y, Z	add custom plane	Custom plane for guidance sketches for leg loft in the - X + Z Plane
13	1,2,6,7,8,9,10,11,12	Add reference sketch for loft	Shape, Size, Location, Orientation	shape: Quadrangle, Size: ... , Geometric_Center: ... , Orientation: Matches custom plane r_r 12	Sketch	Reference sketch for loft in the - X + Z Plane

Table 2: Decision case protocol excerpt 2

$p \in P$	c
6	Lay down the initial sketch for one foot of the legs which will house the axial connection for the endpoints of the chain drive. Add a chamfer which will act as the connecting face for the legs.
7	Mirror the sketch of the foot in the -X +Z Plane
8	Sketch the attachment point geometry to which the legs connecting the chassis to the feet of the chain drive will be connected. The attachment point is to be a reinforcing geometry attached to the main body
9	Mirror the sketch of the attachment point in the -X +Z Plane
10	Extrude feet sketches to form geometries for further modelling
11	Extrude attachment point sketches to form geometries for further modelling
12	Add custom reference plane to accommodate additional sketches for guiding a loft connection between the chassis attachment points and the feet in the -X Plane
13	Add sketch to guide the loft connection between the chassis attachment points and the feet in the -X +Z Plane

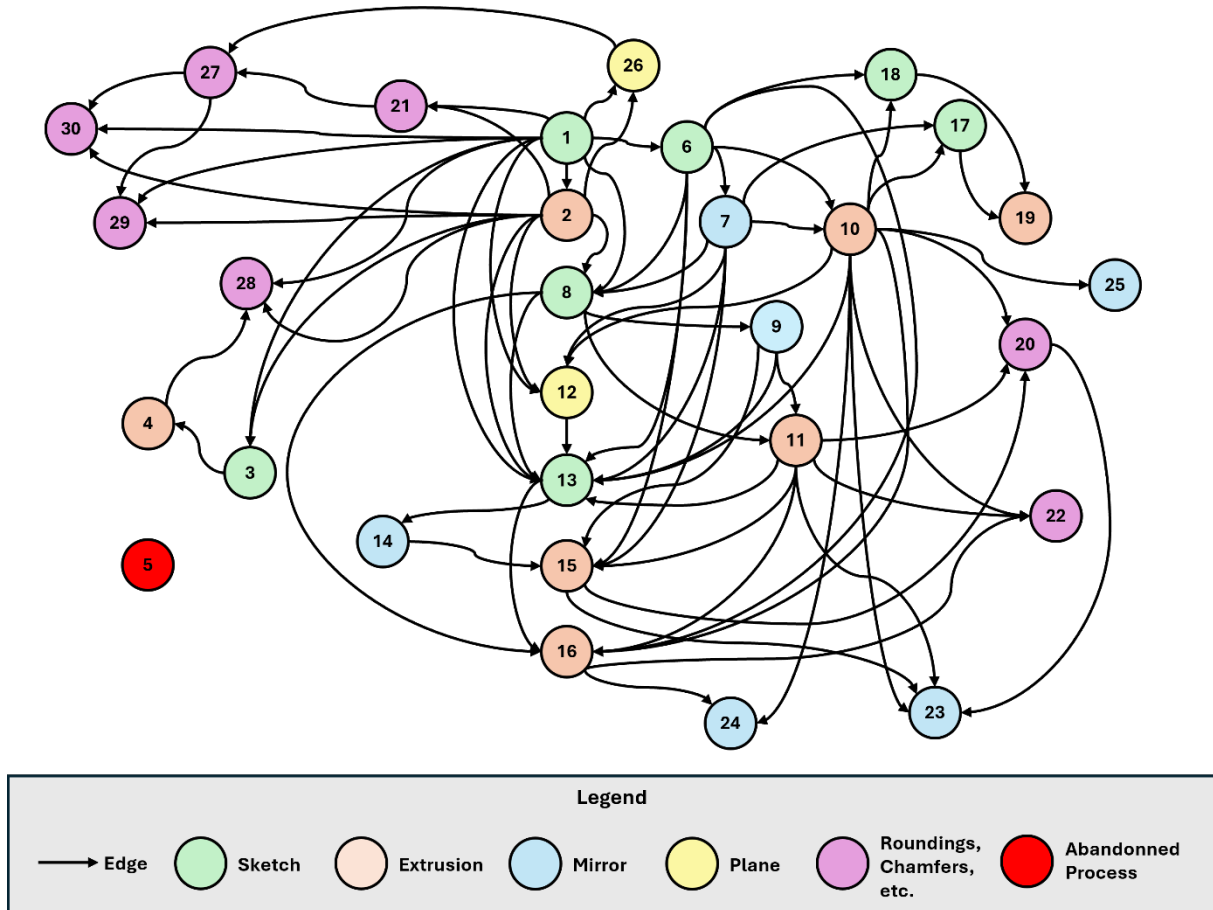


Figure 2: Graph visualization of the Chassis modelling process

Figure 2 shows a visualization of the graph g_P of the modelling process P of the chassis concept, showing the connections between the different nodes p . Green nodes represent sketches, orange nodes extrusions, blue nodes mirroring, yellow nodes custom planes, purple nodes roundings and similar geometric modelling manipulations. Red node 5 represents an abandoned sketch process, which is documented but has no direct bearing on the completion of the artifact. From all 30 subprocesses p there are 8 sketching based subprocesses, 7 extrusion-based processes, 6 mirroring-based processes, 2 creations of custom planes and 7 roundings and similar geometric manipulations. There are in sum 73

edges recorded, representing the influence different subprocesses have on one another. Figure 3 shows the resulting artifact A in Fusion 360 as well as screenshots showing results of $p \in P$ 6 to 13.

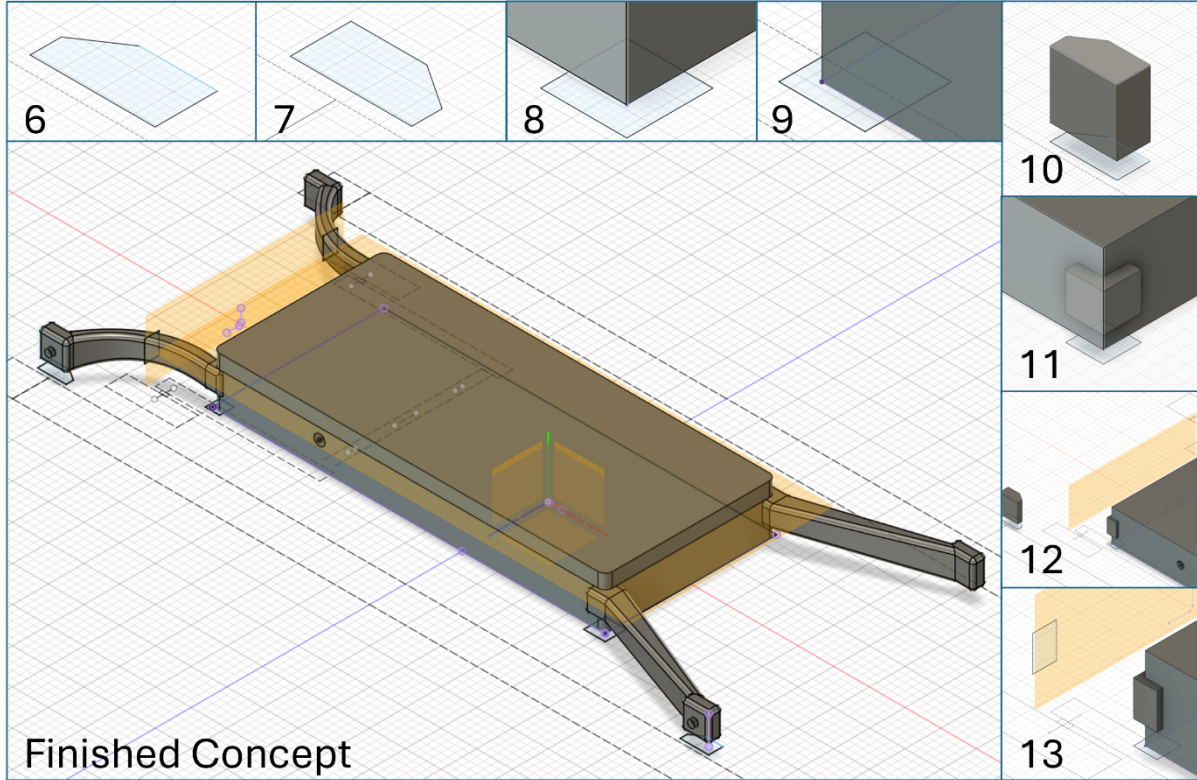


Figure 2: Results of $p \in P$ 6 to 13 and finished chassis concept in Fusion 360

This case study shows that the proposed theoretical foundation is a viable approach to document GDCs within the DDP of the PDP, resulting in a protocol which documents incremental design decisions with annotated context and parametric information, as well as being capable of modeling how different decision cases influence each other with graph visualizations. The resulting protocol is highly comprehensible in natural language and structured in a highly reproducible and adaptable way, which suggests that it lends itself to natural language processing.

This case study viewed environment object e and documentation b as static corpora of text. To reduce the size of input sets of tokens within NLP based processing, as well as to improve processing performance these objects would benefit from the removal of unneeded information. This might become necessary or highly beneficial for DDP processes of higher complexity, meaning a higher number of $p \in P$ and/or larger text corpora of e or b .

7 Application of the Framework

The proposed theoretical foundation describes the PDP as a graph of granular, incremental decisions, the complexity of which is tied to the complexity of the design process and the result artefact A . If one were to analyse a PDP P in its entirety, depending on the complexity of the task, the amount of work required to model each step from protocol would be prohibitively high if there were no automation used.

The theoretical foundation could also be used to model singular decision cases p . Instead of reconstructing the entire process graph of P one could just model specific p , for example if only certain

activities are of interest for automation. To do this only relevant result information r_p from previous steps and the objectives of the designers need to be captured, to only model a single node p in Graph P . This may come at the cost of losing some accuracy when modelling and gathering information sources and tracking the relations of GDC but offers a pragmatic approach to modeling singular activities. This allows the framework to be utilized for the analysis of GDC, without the necessity to model the complete graph of interconnected decision cases.

This framework has explicitly been developed with the intent of being applied to the analysis of design processes with the goal of developing HDSS automation tools. As such, a large focus is laid on information objects represented as sets of embeddings for the use in AI methods, as well as describing the process as a step-based cumulative decision process. This approach or similar methodologies might be adaptable onto all kinds of other activities in which an artifact is created through cumulative decision cases.

Transformer models offer a great opportunity to apply the proposed framework in conceptualizing HDSS applications. Inputs c, d can be translated into tokens for models with agentic capabilities to process $f(c, d)$. This can be achieved by utilizing, among others, the Model Context Protocol (MCP) standard, which offers the potential to realize widely inter-compatible framework to develop AI agents combining symbolic and SSAI capabilities and strengths. A powerful option of synthesizing b for context specific cases could be achieved using RAG which through specialized embedded vector search allows for the retrieval of context sensitive information from large corpora of text for augmenting processing results in transformer models [19]. RAG could be leveraged to find constraints for $f(c, d)$ from project specifications, design handbooks and norms, while frameworks like MCP could handle API calls, command injections or code generation for granular design tasks. This framework allows for the necessary modelling of activities and information to conceptualize such transformer-based agents, using current standards. This framework should also be adaptable to future AI frameworks and standards, if their inputs are vector based and their processing methods are of comparable nature.

8 Discussion and future research

This contribution presents a theoretical foundation to model GDC within DA. The framework combines the granular nature of decision-making cases within the design process while also including the holistic insights of the EOS model, providing an answer to the formulated research question. It also offers a framework to model GDC with semantic descriptions of context and operations along with numerical parameters, as demonstrated by the case study, which represents a shift in paradigm in PDP modeling for AI applications for the goal of subsequent automation. It can be used to analyze activities of the detailed planning phase with a focus on conceptualizing HDSS based AIA/AAI automation tools by providing the foundation for context engineering of input objects for transformer models, for example via RAG, as well as describing results of process functions and associated decision case parameters. The validity of the proposed theoretical foundation has been demonstrated by a case study using student project work, showing that it can be used to model DAs within the DPP.

One avenue of future research will focus on automated analysis utilizing the proposed approach to building knowledge repositories. If possible, this method would lend itself well to creating databases for documenting large numbers of incremental, singular design decisions with annotated context. Such a database might be used to retrieve relevant granular steps fitting the current context c and objectives o using RAG. This would require many documented P , which requires in turn specific documentation as well as manual effort to translate into the proposed format. Future development of this approach could also include the automatic documentation of $p \in P$ for environments aside from Fusion 360, as well as the automatic creation of protocol for P as found in the case study. Another avenue of future research

will focus on developing HDSS implementations for DPP activities of the PDP utilizing the proposed theoretical framework. The framework will therein be used to analyze design activities and find processes which offer great potential for HDSS implementations using AIA/AAI utilizing MCP and RAG or methodologies and standards of similar formats and capabilities.

References

- [1] “Hybrid Decision Support in Product Creation – Improving Performance with Data Science and Artificial Intelligence,” *I40S*, vol. 2025, no. 1, 2025. DOI: 10.30844/I4SE.25.1.18.
- [2] S. Yilmaz, C. Seifert, S. R. Daly and R. Gonzalez, “Design Heuristics in Innovative Products,” *Journal of Mechanical Design*, vol. 138, no. 7, p. 071102, 2016. DOI: 10.1115/1.4032219.
- [3] L. A. Stauffer and D. G. Ullman, “Fundamental Processes of Mechanical Designers Based on Empirical Data,” *Journal of Engineering Design*, vol. 2, no. 2, pp. 113–125, 1991. DOI: 10.1080/09544829108901675.
- [4] Y. P. Tsang and C. K. M. Lee, “Artificial Intelligence in Industrial Design: A Semi-Automated Literature Survey,” *Engineering Applications of Artificial Intelligence*, vol. 112, p. 104884, 2022. DOI: 10.1016/j.engappai.2022.104884.
- [5] B. Lee and J. Kim, “AI-Led Design Innovation: Understanding Design-Centric AI Methods and Assistance Types,” *Proceedings of the Design Society*, vol. 5, pp. 469–478, 2025. DOI: 10.1017/pds.2025.10061.
- [6] C. Krahe, A. Bräunche, A. Jacob, N. Stricker and G. Lanza, “Deep Learning for Automated Product Design,” *Procedia CIRP*, vol. 91, pp. 3–8, 2020. DOI: 10.1016/j.procir.2020.01.135.
- [7] D. C. Wynn and P. J. Clarkson, “Process Models in Design and Development,” *Research in Engineering Design*, vol. 29, no. 2, pp. 161–202, 2018. DOI: 10.1007/s00163-017-0262-7.
- [8] J. Wei et al., “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models,” *arXiv*, 2022. DOI: 10.48550/arXiv.2201.11903.
- [9] C. Yin and W. Zhang, “New Product Development Process Models,” in *Proceedings of the 2021 International Conference on E-Commerce and E-Management (ICECEM)*, Dalian, China: IEEE, 2021, pp. 240–243. DOI: 10.1109/ICECEM54757.2021.00054.
- [10] D. G. Ullman, T. G. Dietterich and L. A. Stauffer, “A Model of the Mechanical Design Process Based on Empirical Data,” *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 2, no. 1, pp. 33–52, 1988. DOI: 10.1017/S0890060400000536.
- [11] J. J. Shah, D. K. Jeon, S. D. Urban, P. Bliznakov and M. Rogers, “Database Infrastructure for Supporting Engineering Design Histories,” *Computer-Aided Design*, vol. 28, no. 5, pp. 347–360, 1996. DOI: 10.1016/0010-4485(95)00054-2.
- [12] P. Lunnemann, R. Stark, W. M. Wang and P. I. Manteca, “Engineering Activities—Considering Value Creation from a Holistic Perspective,” in *Proceedings of the 2017 International Conference on Engineering, Technology and Innovation (ICE/ITMC)*, Funchal: IEEE, 2017, pp. 315–323. DOI: 10.1109/ICE.2017.8279904.
- [13] M. Alam, P. Groth, P. Hitzler, H. Paulheim, H. Sack and V. Tresp, “CSSA’20: Workshop on Combining Symbolic and Sub-Symbolic Methods and Their Applications,” in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual Event, Ireland: ACM*, 2020, pp. 3523–3524. DOI: 10.1145/3340531.3414072.
- [14] M. Cheng et al., “A Survey on Knowledge-Oriented Retrieval-Augmented Generation,” *arXiv preprint*, 2025. DOI: 10.48550/arXiv.2503.10677.
- [15] K. Hiniduma, S. Byna, J. L. Bez and R. Madduri, “AI Data Readiness Inspector (AIDRIN) for Quantitative Assessment of Data Readiness for AI,” in *Proceedings of the 36th International Conference on Scientific and Statistical Database Management*, Rennes, France: ACM, 2024, pp. 1–12. DOI: 10.1145/3676288.3676296.
- [16] A. Vaswani et al., “Attention Is All You Need,” *arXiv preprint*, 2017. DOI: 10.48550/arXiv.1706.03762.
- [17] S. Brohi, Q. Mastoi, N. Z. Jhanjhi and T. R. Pillai, “A Research Landscape of Agentic AI and Large Language Models: Applications, Challenges and Future Directions,” *Algorithms*, vol. 18, no. 8, p. 499, 2025. DOI: 10.3390/a18080499.
- [18] R. Sapkota, K. I. Roumeliotis and M. Karkee, “AI Agents vs. Agentic AI: A Conceptual Taxonomy, Applications and Challenges,” *Information Fusion*, vol. 126, p. 103599, 2026. DOI: 10.1016/j.inffus.2025.103599.
- [19] M. Klesel and H. F. Wittmann, “Retrieval-Augmented Generation (RAG),” *Business & Information Systems Engineering*, vol. 67, no. 4, pp. 551–561, 2025. DOI: 10.1007/s12599-025-00945-3.