Proceedings of the Design Society, Volume 6: DESIGN 2026
https://doi.org/10.1017/pds.2026.10591

In search for working principles using large language models: an experimental study

Fabio Hanspach[1,✉], Kilian Gericke[2], Ute Dietrich[3] and Boris Eisenbart[1]

[1] Swinburne University of Technology, Australia, [2] University of Rostock, Germany, [3] Hochschule für Technik und Wirtschaft Berlin, Germany

✉ fhanspach@swin.edu.au

ABSTRACT: The use of artificial intelligence, especially large language models (LLMs), is increasingly explored to support early system development. This paper evaluates low-threshold LLM-based tools for supporting conceptual design. Through an experiment, two LLM-based tools were compared generating alternative solutions using an existing function model of an electro-mechanical system as input. Functions were provided in natural language and using the Functional Basis. Results show limitations and significant potentials for effective and efficient conceptual design support.

KEYWORDS: artificial intelligence (AI), early design phase, generative design, systems engineering (SE), design methods

1. Introduction

Progressing digitalisation in design and engineering presents both huge opportunity but also new challenges for industrial operations. The nature of product and system design is changing, transitioning from conventional mechanical and mechatronic systems to smart, highly interconnected socio-technical or cyber-physical systems (CPS) (Tekaat et al., 2021). This entails increasing functional and physical interconnection between human actors, hardware and software, leading to asynchronous development cycles caused by the multi-domain interactions. The resulting increase in the number of disciplines participating in the development of products and shortening time-to-market windows lead to faster development cycles, meaning that designers deal with a comparatively larger solution space, yet have less time to do so. This highly demanding task represents a ubiquitous and significant challenge, requiring effective and reliable computational approaches, while integrating and supporting cross-disciplinary collaboration at the same time (Brahma & Wynn, 2023; Koh et al., 2012). The fields of Design Theory and Methodology (DTM) and systems engineering (SE) provide approaches that aim to assist engineering designers in managing complexity, enhance collaboration among teams and create viable systems designs (INCOSE, 2023; Pahl et al., 2007). Specifically for the creation of an initial viable system design, Pahl et al. (2007) propose function modelling to assist and guide the transition from a requirements list to an initial solution proposal to match these requirements. The generation of function structures that describe desired product features and their logical flows encompasses one of the first essential steps in the conceptual design stage and prepares the subsequent 'creative leap'. The latter encompasses the generation of a first possible solution by matching individual desired functions with alternative working or solution principles. This is a critical cognitive task and contributes significantly to the characteristics of the product in so far as it has a decisive influence on the product's components, structures and their combination in an initial product/system architecture (INCOSE, 2023; Pahl et al., 2007). Consequently, design engineers must develop and manage a variety of alternative solutions, considering all the additional complexities mentioned above. Cognitive overload and management of complexity is a growing issue demanding effective support in conjunction with seamless cross-disciplinary collaboration (Brahma & Wynn, 2023; Koh et al., 2012).

https://doi.org/10.1017/pds.2026.10591 Published online by Cambridge University Press

In the advent of computational, digitalised design and engineering support, e.g. through machine learning and artificial intelligence (AI), there is an opportunity to provide designers and engineers with appropriate and effective support. However, while AI has seen rapid uptake in early and late product design stages (Poulsen et al., 2025), the central task of generating a first system/product design and assigning concrete components to make up a coherent product/system architecture, still lacks effective computational support (Timperley et al., 2025).

The research presented here seeks to explore what are the concrete barriers for AI, in particular large language models (LLM) with their well-established natural language communication skills, to provide relevant support in the generative design process. Through this, the aim to bridge the gap in computational support between initial visual or textual ideation - for which many AI-tools exist - and conceptual design support, which is not adequately supported yet. In the long-run, this should unlock new capabilities for AI-assisted design workflows and setting the foundation for more intelligent, traceable and adaptable design systems. In line with established research into design methodology (Bender & Gericke, 2021), system design (Hubka & Eder, 1988), and systems engineering (INCOSE, 2023), the paper will focus on the use of function level modelling and connecting it to associated generative AI approaches to produce an initial proposal of alternative working principles. As a first step towards this goal, this paper explores:

- Are LLM-based tools able to support designers searching for working principles?
- How does the formalisation of input data (function models) affect the variation and accuracy of results?

The research questions will be answered through an experimental study utilising common LLM-based tools to create and formalise preliminary, alternative solution proposals from an existing function model of an electro-mechanical system. Extending the insights obtained, the shortcomings in the utilised tools that need to be overcome in the future to provide adequate support are then advanced. This will then set exact targets for training and further development in the field.

2. Theoretical framing

Prescriptive approaches from both DTM and SE propose the use of function modelling as discussed above to translate requirements into an initial design proposal. Function models aim to assist engineers in breaking up complex requirements and desired operations/features that a to-be-developed product is meant to fulfil into more manageable partial design tasks. That means, while all functions, put together into a cohesive flow, incorporate a full product's design, designers can focus on a (sets of) function(s) at a time to overcome preconceptions/design fixation, explore a (larger) solution space and find the most appropriate option to implement consequently (Eisenbart, 2014; INCOSE, 2023; Pahl et al., 2007).

While there is a large variety of definitions and function models/modelling approaches proposed in literature (Eisenbart, 2014; Erden et al., 2008), perhaps the most widespread approach has been proposed by Pahl et al. (2007; Pahl & Beitz, 1977). The model incorporates abstraction of what a product or system is meant to achieve as an overall transformation of some type of inputs to corresponding outputs. This establishes a causal relationship between these very inputs and outputs - by the logic of the corresponding transformation. And by doing so, the product/system fulfils its overall purpose, however this purpose may be natured. Gradually, this overall transformation relation is then broken up and refined through determining what the 'essence' or 'crux' (Pahl et al., 2007, p.169) is as per what the product as a whole is intended to achieve and what partial functions are required to achieve it. This can be captured in short statements using natural language capturing the nature of each function and relevant connotation (see Section 3 for relevant examples used in this study). For example, the main function of a carjack could be worded as "lift car up". Gradually, this initial set of functions is then further refined by logically dissecting the causal relations of inputs and outputs and required steps that - combined - enable the overall transformation of inputs to desired outputs that was initially established to occur (Eisenbart & Gericke, 2020). The derived combination of all functions specifies the causal transformations carried out by the product with respect to the flows of "operands" going through it, where operands are typically specifications of energy, materials, and signals.

Using such a generated breakdown of functions, Pahl et al. (2007) then propose the use of morphological charts which matches individual functions with fundamental working and/or solution principles that will fulfil each function. These are still at an abstract level and permit designers to consider a wide range of options to compare based on utility and relative merits. Selection of the most appropriate option then - in

https://doi.org/10.1017/pds.2026.10591 Published online by Cambridge University Press

combination - builds the initial principle solution to be further developed in detail. It should be noted that design catalogues (Roth, 2000) can provide additional support at this exploratory stage.
The research presented here adopts the discussed function modelling and solution finding approach by Pahl et al. (2007) as it follows a logical flow of inputs and outputs - which lends itself to computational processing - and is recognised as effective and one of the most widespread approaches used across disciplines. More so, essential research has been undertaken into formalisation of this very function modelling approach to facilitate logical and/or computational processing of functions and solution proposals. The most prominent example is the *Functional Basis* (FB) (Hirtz et al., 2002; Stone & Wood, 2000). This is essentially a taxonomy of terms to be used to abstract natural language function descriptions into distinct types of operations to describe the transformation of inputs to outputs for each function in the model. This follows two main objectives. Firstly, it allows logical processing as per the basics type of operation in the FB. Secondly, engineers may avoid early fixation on a specific solution principle by the operation that is described in natural language otherwise, opening up solution exploration to even more conceptually distant options.
Given the very purpose of the FB to assist with computational/logical processing and wide exploration of possible solution proposals, it is worth considering its application in the present research. Consequently, a function model is used here, produced following the approach by Pahl et al. (2007) both in its original form using natural language as well as after formalisation of the same model using the FB. Special attention will be paid to the results creative variety and fluency. This follows the assumption that the FB will permit wider exploration versus that the use of natural language is closer to the lived reality of engineers in practice (Eckert, 2013). Thus, a comparison of the solutions being proposed will be interesting to the development of future AI-driven conceptual design support.

3. Study design

For the experimental study, an existing, publicly available function model of a hot glue gun by the authors Gericke & Eisenbart (2017, see Figure 1) was chosen for three main reasons. Firstly, it has an appropriate level of complexity and is still intuitively intelligible to readers without requiring significant knowledge of the used product and technology applied. This allows broad discourse and focus on the performance of the LLM-based tool, rather than the model itself. Second, it rigorously follows the methodology of Pahl et al. (2007) including specifications of flows of material, energy and signal. Finally, it represents a published and peer-reviewed function model that is accessible to the wider research community thus allowing scrutiny and easing replication of this work by other researchers.

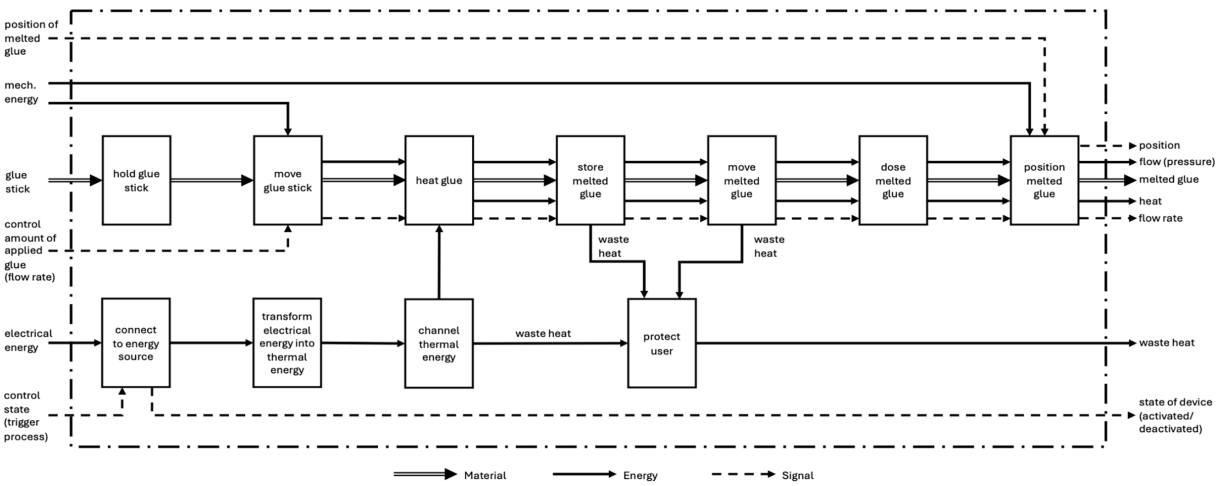


Figure 1. Function model of a hot glue gun, adapted from (Gericke & Eisenbart, 2017) - function model-natural language (short: FM-NL)

Forthwith, this will refer to as *function model-natural language* (short: FM-NL). In formalising the model using the FB, the taxonomy was applied on the secondary level of semantic abstraction (see Hirtz et al., 2002) to the existing FM-NL. It must be noted, this required a slight adjustment of the concrete functional breakdown used in the model to comply with FB formalisation. Mainly, import and export type functions were added. Secondly, internal operand flow descriptions between each partial function

https://doi.org/10.1017/pds.2026.10591 Published online by Cambridge University Press

block were added, where the original model featured the overall input and output flow denominations (see Figure 1). The resulting model is shown in Figure 2. This is referred to as *function model-functional basis* (short: FM-FB). Table 1 documents the translation of each function from FM-NL to FM-FB.

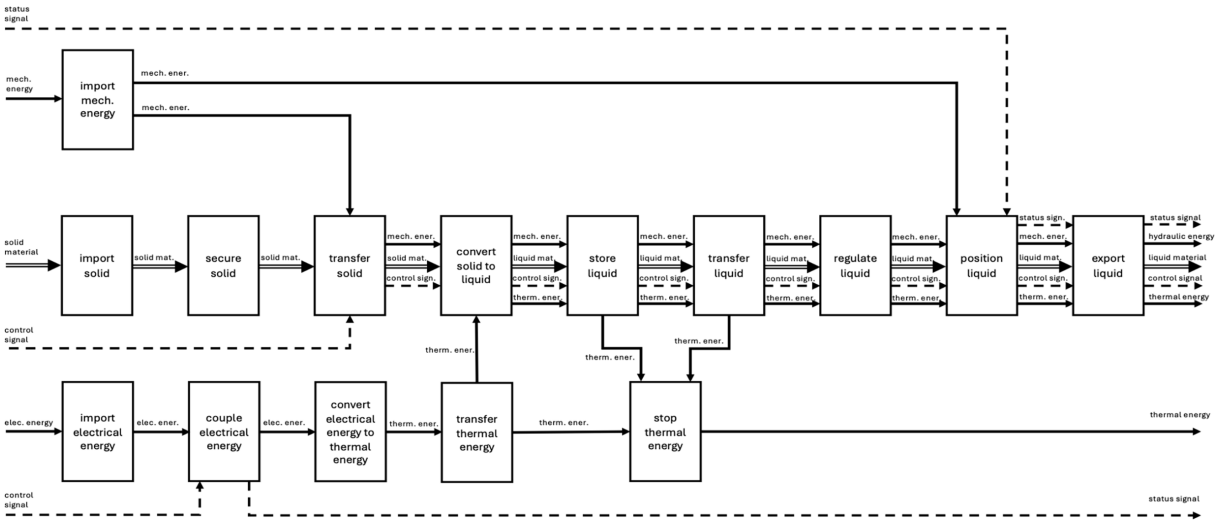


Figure 2. Function model of a hot glue gun with functional basis applied - *function model-functional basis* (short: FM-FB)

Table 1. Translation of functional model of a hot glue gun from natural language (FM-NL) to FB (FM-FB)

Natural Language (Gericke & Eisenbart, 2017)		**Functional Basis (second level, Hirtz et al., 2002)**	
Function-ID	**Function Name**	**Function-ID**	**Function Name**
–	–	FU_1	import mechanical energy
–	–	FU_2	import solid
FU_1	hold glue stick	FU_3	secure solid
FU_2	move glue stick	FU_4	transfer solid
FU_3	heat glue	FU_5	convert solid to liquid
FU_4	store melted glue	FU_6	store liquid
FU_5	move melted glue	FU_7	transfer liquid
FU_6	dose melted glue	FU_8	regulate liquid
FU_7	position melted glue	FU_9	position liquid
–	–	FU_10	export liquid
–	–	FU_11	import electrical energy
FU_8	connect to energy source	FU_12	couple electrical energy
FU_9	transform elect. ener. to therm. ener.	FU_13	convert elect. ener. to therm. ener.
FU_10	channel thermal energy	FU_14	transfer thermal energy
FU_11	protect user	FU_15	stop thermal energy

It stands to reason that the internal flows of operands per every function may assist finding *appropriate* solution proposals for them more easily, i.e. finding such working principles that match the associated operand flows more closely. The rationale behind this step is that the internal flows might improve interpretability of the functions by the LLM-based tool. As such, a derivation of the original FM-NL was created wherein operand flows were added for each function - where flows match FM-FB. The resulting third model is referred to as *function model-natural language with flow description* (short: FM-NL-flow, Figure 3). All three models will be used in the experiment and results compared.

https://doi.org/10.1017/pds.2026.10591 Published online by Cambridge University Press

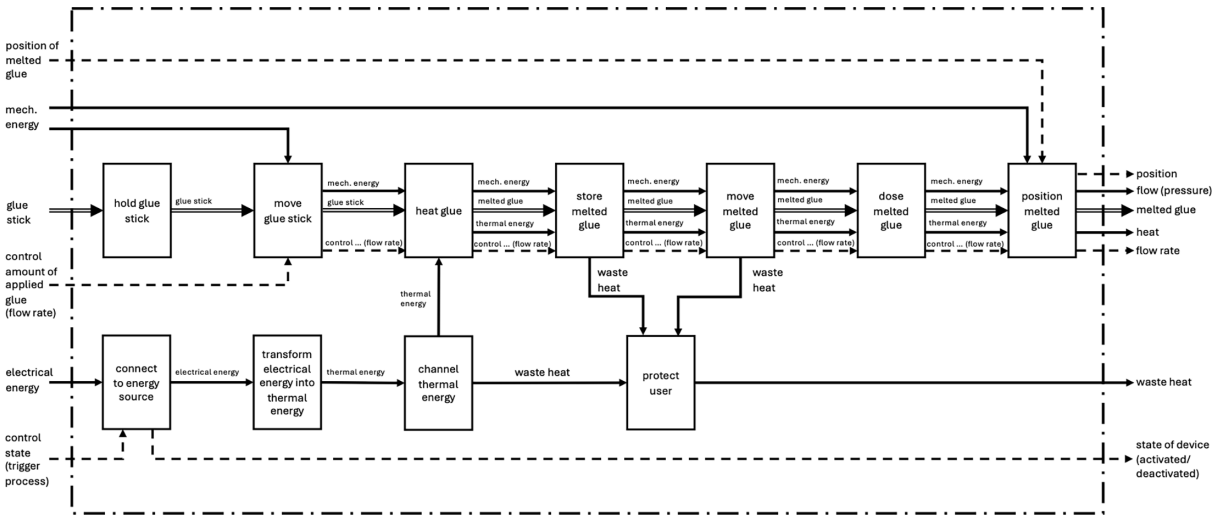


Figure 3. Adapted version of function model-natural language (short: FM-NL) incorporating internal operand flows, as per FB logic

Trough the rapid developing landscape of AI, especially those using LLMs, a conscious decision was made to limit the scope of this paper to LLMs with low-threshold access and availability. Equally, it allows replication of the results presented in the following by other researchers and easy expansion on our work. Therefore, ChatGPT and Copilot were selected. The selected model was ChatGPT 5 (Mode: Auto; Date: 30.10.2025) and Copilot 365 (Version number: bizchat.20251023.62.2) used with its default settings. The tools were not pre-trained for this specific application, which facilitates an unbiased analysis of their current capabilities. For all experiments, the memory functions of the tools were turned off, and the option of "Temporary Chat" was activated to further contribute to comparability and prevent progressive training of the underlying LLM algorithm while running the experiment. Each experiment was conducted in a single prompt with the identical prompt procedures. Figure 4 illustrates an excerpt of the final prompt with the applied scheme for the FM-NL.

1. Introduction

```
Generate a matrix of working principles (per
Pahl & Beitz) for each listed function.
Functions = rows; working principles =
columns.
Each cell: one distinct, concise noun phrase
(≤10 words).
Output only the matrix - no explanations or
extra text.
Input:
Function Model List:
```

2. Schema structure

```
Function Model – Glue gun
IDs:
- FU_<n> = Function ID
- FL_<n> = Flow ID
- IN_<n> = System Boundary Input ID
- OUT_<n> = System Boundary Output ID
- n = unique integer
- type ∈ {M, E, S} where M = Material, E =
Energy, S = Signal
Formats
Function:
- FU_<n> – name: <function name>
System Boundary:
- IN_<n> – type: M|E|S – name: <System input
name>
- OUT_<n> – type: M|E|S – name: <System
output name>
Flow:
- FL_<n> – from: <ID>, name: <from name> ->
to: <ID>, name: <to name> –
  type: M|E|S – name: <flow name>
  where <ID> ∈ {FU_<n>, IN_<n>, OUT_<n>}
```

3. Model-dependent description

```
Function:
- FU_1 – name: hold glue stick
- FU_2 – name: move glue stick
- FU_3 – name: heat glue
- FU_4 – name: store melted glue
[…]
System Boundary – Input:
- IN_1 – type: M – name: glue stick
[…]
System Boundary – Output:
- OUT_1 – type: M – name: melted glue
[…]
Flow:
- FL_1 – from: IN_1, name: glue stick -> to:
FU_1, name: hold glue stick – type: M –
name: glue stick
- FL_2 – from: FU_1, name: hold glue stick -
> to: FU_2, name: move glue stick – type: M
– name: none
- FL_3 – from: FU_2, name: move glue stick -
> to: FU_3, name: heat glue – type: M –
name: none
- FL_4 – from: FU_3, name: heat glue -> to:
FU_4, name: store melted glue – type: M –
name: none
[…]
```

Figure 4. Prompt excerpt - function model-natural language (short: FM-NL)

Each prompt consists of three parts. First the prompt introduction, it was kept short to reduce influence while providing only a directive for the model by describing the task, structure and minimal constraints of the output to ensure comparability. The requested result of each experiment was a morphological matrix organising working principles for several functions. According to Pahl and Beitz (Bender & Gericke,

https://doi.org/10.1017/pds.2026.10591 Published online by Cambridge University Press

2021, p. 257) a working principle is defined as a solution principle for fulfilling a function at the first concrete level, consisting of the underlying physical, biological or chemical effect as well as geometric and material characteristics (working geometry, working movement and material).
This is followed by the structure of the developed schema for translating the graphical representation of the function models into text. To systematically capture the visual information, three formats were established: function, system boundary (input/output) and flows. The final part emerges from the application of the scheme and is therefore model-dependent.
After conduction the experiments the LLM generated proposals were independently analysed by two domain experts with extensive and long-term experience in function modelling and design research. The analysis was guided by the following qualitative criteria:

- formal correctness after (Pahl et al., 2007), this paid special attention to whether proposed solutions were at the level of working principles as per the original definition of the term or at a more concrete level already or whether there were any other formal deficiencies;
- similarity/congruence of the proposed working principles from the same model and LLM tool, which would indicate the same solution being proposed twice or more times, even if phrased slightly different;
- incorrect or nonsensical entries.

The expert assessments were subsequently evaluated for alignment by the first author of this paper. Whenever the expert assessments did not align a final assessment was made.

4. Findings

In this experimental study, a total of six experiments were conducted: firstly, using the three function models (FM-NL, FM-NL-flow and FM-FB) as inputs to ChatGPT and correspondingly to Copilot. After evaluation of the LLM proposals by the researchers, the LLM tools performances were compared quantitatively.

4.1. General findings

Table 2 provides an overview of the results produced from all three runs for both ChatGPT and Copilot. On average, across all experiments, less than 30% of the results were evaluated to be formally correct at the level of abstraction that abides by the definition of working principles by Pahl et al. (2007). The majority of the proposals by either LLM tool were at the level of solution principles and thus already at a more concretised level compared to pure electro-mechanical working principles. While these had to be initially flagged as formally incorrect as per the prompt request and the Pahl et al. (2007) methodology, conversely, it must also be noted that there were no proposals found to be even more concrete than the solution principle level either.

Table 2. Overview of results after evaluation

AI	Input-Model	Working Principle	Solution Principle	Nonsensical	Similarity
ChatGPT	FM-NL (n=66)	24 (36%)	34 (52%)	8 (12%)	0
	FM-NL-flow (n=77)	22 (29%)	46 (60%)	9 (12%)	4
	FM-FB (n=120)	35 (29%)	78 (65%)	7 (6%)	1
Copilot	FM-NL (n=33)	9 (27%)	23 (70%)	1 (3%)	0
	FM-NL-flow (n=33)	8 (24%)	22 (67%)	3 (9%)	0
	FM-FB (n=45)	12 (27%)	29 (64%) 1 (2%)*	3 (7%)	0

*formal incorrect description, which is neither a working principle nor a solution principle

The nonsensical proposals, which accounted for a small percentage and were found to be related mainly to specific functions. Particularly notable here is FU_10 “channel thermal energy”, which pertains to melting the glue stick in the glue gun and thus must have *good* thermal conductivity. However, in almost all input models, LLMs effectively proposed *insulation* strategies instead as viable solutions. As can be seen in Table 3, this includes “reflective heat shield” while other runs produced “ceramic insulator”,

https://doi.org/10.1017/pds.2026.10591 Published online by Cambridge University Press

"insulating sleeve" and the likes for the same function, respectively. While it can be easily seen how these solution proposals relate to the concept of heat transfer, it does not present a viable alternative and any such case was removed from further analysis.

Another finding pertains to the limited intelligibility of the output format. While the use of up to 10 words to describe and explain the generated proposal was allowed, no output used more than four words, with the vast majority consisting of only two to three words, respectively (see Table 3). For the evaluation, the research team needed to request further information manually in about 20% of cases to help comprehension of what exactly the proposed option was. Furthermore, the results reveal a pattern in how both models interpreted the instruction to generate the matrix. Although the prompt did not specify the number of columns, both models produced a uniform number of proposals per function, with no variation across rows. ChatGPT generated matrices containing six, seven, or eight proposals per function (depending on the experiment), whereas Copilot consistently produced exactly three proposals per function. Table 3 provides an illustrative example of the output pattern generated by the experiment ChatGPT for FM-NL, consisting of six proposals per function.

Table 3. Representative excerpt of an output generated by the experiment ChatGPT for FM-NL

FU_ID	Function	Working principles/solution principles					
FU_3	heat glue	resistive cartridge heater	PTC heater	induction coil	hot air convection	infrared radiator	thermal oil jacket
FU_4	store melted glue	heated reservoir	melt chamber	accumulator cavity	insulated sump	buffer channel	heated manifold
FU_10	channel thermal energy	copper heat spreader	alum. heater block	thermal conduction path	heat pipe assist	forced-air duct	reflective heat shield

4.2. Comparative analysis

First, the variation and fluency of the outcomes are examined in a comparison between the two LLM tools used. After removal of duplicates and nonsensical proposals, on average, ChatGPT generated roughly twice as many working principles (ChatGPT: $M = 2.2$, $SD = 1.5$; Copilot: $M = 0.8$, $SD = 1.0$) and solution principles (ChatGPT: $M = 4.1$, $SD = 1.9$; Copilot: $M = 2.0$, $SD = 1.1$) per function compared to Copilot. This is using *the exact same prompt for either LLM*. Further, it is found that with less than a handful of exceptions, all proposals from Copilot also appear in the results generated by ChatGPT. This indicates that there is a fundamental difference in the underlying algorithm of the two tools. Since, both are considered to have large training data to support their functions - and there is significant overlap in the proposals made, as discussed - the difference is likely due to a different search and selection strategy. In other words, Copilot appears to apply a more selective strategy in what it generates and displays to the user.

Next, attention is turned to variation/fluency as per the input model used in each experiment. A clear influence on the quantity of generated alternative solution proposals can be observed. In both ChatGPT and Copilot, the application of the FB taxonomy led to significantly more working and solution principles being generated and thus to be offering engineering designers a potentially larger solution space to evaluate and choose from (Table 4): for ChatGPT FM-FB 112 versus 58/64 and for Copilot FM-FB 41 versus 32/30, respectively. There is no such clear trend for the inclusion of the interim operand flows, where ChatGPT produced more options from their incorporation (64 versus 58), while Copilot generated fewer (32 versus 30).

Equally not as clear cut is another phenomenon, but no less noteworthy. It is found that ChatGPT produces not only the most amount of proposals using the FB formalised function model as input, there is then also a smaller amount of nonsensical generated outputs (from 12% in either natural language model to only 6%), making this effectively the highest quality output. The same is not true for Copilot where this is more mixed (see Table 2).

It must be noted that an assessment of the generated outputs in terms of their relative quality was deliberately not performed.

https://doi.org/10.1017/pds.2026.10591 Published online by Cambridge University Press

Table 4. Final series of valid proposals derived from the experiments conducted

AI	Input-Model	Functions per input-model	Working Principles	Solution Principle
ChatGPT	FM-NL (n=58)	11	24 (41%)	34 (59%)
ChatGPT	FM-NL-flow (n=64)	11	22 (34%)	42 (66%)
ChatGPT	FM-FB (n=112)	15	35 (31%)	77 (69%)
Copilot	FM-NL (n=32)	11	9 (28%)	23 (72%)
Copilot	FM-NL-flow (n=30)	11	8 (27%)	22 (73%)
Copilot	FM-FB (n=41)	15	12 (29%)	29 (71%)

This is because following established creative techniques, the mentioned DTM (Hubka & Eder, 1988; Pahl et al., 2007) and SE (INCOSE, 2023) approaches, the crux of this stage in the overall development process is to generate a large variety of alternatives for the engineering team to draw inspiration from, compare and select the best one by its efficacy and merit with respect to the specific aims and KPIs relevant to each individual design process. Meaning, the relative quality of any proposal is case-dependent. And a larger pool of options to represent a viable alternative is considered conducive at this stage of the process. This is also relevant in relation to the observed behaviours between ChatGPT and Copilot as per the generated qualities, which is discussed in the following section.

5. Discussion and conclusion

It can be argued, that the LLM tools are indeed able to produce relevant, alternative solution proposals at an abstract level (see Table 2), which still matches the overall purpose of the conceptual design stage, i.e. to arrive at an assortment of abstract alternative solutions to choose from. Furthermore, the results across all experiments demonstrate a notable advantage for designers, supporting the goal of obtaining as many alternatives as possible in an extremely short amount of time. No manual creative technique could produce a comparable number of options in the time required by the LLMs. However, the extent to which the potential solution space was actually covered was not investigated. Therefore, these results should not be interpreted as a recommendation to rely exclusively on LLMs for this crucial activity within the design process.

The study shows that the FB input model, function model-functional basis (short: FM-FB), achieved the best overall performance in combination with ChatGPT. This is a remarkable result that warrants further investigation. The FB formalisation produced almost twice as many viable alternative working principles and solution principles compared to the natural-language representations while using ChatGPT (see Table 4). Adding the internal flow description (FM-NL-flow) to the function model-natural language (short: FM-NL) did not lead to a significant increase in the number of valid suggestions or working principles generated. This could indicate that the input format for the LLM should be adjusted for better understanding. The findings suggest that there is significant benefit to be had from applying FB formalisation in the generation of a function model of a system under development, when the aim is to produce as many alternative solution proposals as possible to choose from, although it is well documented that most practitioners prefer to use natural language representations of functions in function modelling (compare Eckert, 2013 and Eisenbart, 2014). It stands to reason that a well trained LLM would actually be able to assist designers in the translation of the textual descriptions of functions into equivalent formalised ones using the FB taxonomy. As such, a conclusion and recommendation from this study is that, firstly, the use of the FB taxonomy in function modelling is effective for the use of LLM tools for the generation of initial solution proposals in the conceptual design stage. Secondly, future developments of a more dedicated and trained AI specific to the use of generating initial system concepts would provide assistance or translation routines in terms of formalising natural language formulations into such using the FB taxonomy or its derivates.

Even general-purpose LLM tools are able to create a large volume of alternative solution proposals. The feasibility of the generated proposals was not assessed in this study; however, we believe that providing a larger amount of alternatives is beneficial for designers at least, regardless of whether all of them are feasible. Based on the results the proportion of formally incorrect suggestions and the difficulty of interpreting the brief text-based proposals pose major obstacles to convenient application. Therefore, a more detailed definition of the working principles and a structured representation of the desired output

https://doi.org/10.1017/pds.2026.10591 Published online by Cambridge University Press

format are suggested to further enhance the performance of the LLM tools. In addition, design catalogues, as described by Roth (2000), may serve as valuable datasets for training LLMs. In order to strengthen the significance and generalisability of the study, a variety of comprehensive experiments should be conducted with a broader spectrum of multidisciplinary complex function models. Furthermore, a central challenge is the lack of traceability and reproducibility, which makes it difficult to confidently assess the coverage of the solution space. Yet this also highlights an opportunity: engineering methodology, design expertise and domain knowledge can be leveraged to strengthen traceability, improve selection mechanisms, and ultimately increase the overall performance of LLM tools in future work.

The results of this work provide promising insight into how LLM tools can support the early design phase by rapidly generating a large number of alternative solution proposals for designers, while also highlighting the limitations of their current capabilities. A function model of an electromechanical system (hot glue gun) was selected as an example of an appropriate level of complexity. Further research is needed to generalise the results and evaluate their applicability to multidisciplinary and more complex product designs.

References

Bender, B., & Gericke, K. (2021). *Pahl/Beitz Konstruktionslehre* (9th ed.). Springer. https://doi.org/10.1007/978-3-662-57303-7

Brahma, A., & Wynn, D. C. (2023). Concepts of change propagation analysis in engineering design. *Research in Engineering Design*, 34(1), 117–151. https://doi.org/10.1007/s00163-022-00395-y

Eckert, C. (2013). That which is not form: The practical challenges in using functional concepts in design [Article]. *AI EDAM*, 27(3), 217–231. https://doi.org/10.1017/S089006041300022X

Eisenbart, B. (2014). *Supporting interdisciplinary system development through integrated function modelling* [PhD, University of Luxembourg, Luxembourg].

Eisenbart, B., & Gericke, K. (2020). Function in Engineering. In *The Routledge Handbook of the Philosophy of Engineering* (pp. 245–262). Routledge. https://doi.org/10.4324/9781315276502-22

Erden, M. S., Komoto, H., van Beek, T. J., D'Amelio, V., Echavarria, E., & Tomiyama, T. (2008). A review of function modeling: Approaches and applications [Article]. *AI EDAM*, 22(2), 147–169. https://doi.org/10.1017/S0890060408000103

Gericke, K., & Eisenbart, B. (2017). The integrated function modeling framework and its relation to function structures. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, 31(4), 436–457. https://doi.org/10.1017/S089006041700049X

Hirtz, J., Stone, R. B., McAdams, D. A., Szykman, S., & Wood, K. L. (2002). A functional basis for engineering design: Reconciling and evolving previous efforts. *Research in Engineering Design - Theory, Applications, and Concurrent Engineering*, 13(2), 65–82. https://doi.org/10.1007/s00163-001-0008-3

Hubka, V., & Eder, W. E. (1988). *Theory of Technical Systems*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-52121-8

INCOSE. (2023). *Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities*, Fifth Edition (5th Edition) [Book]. John Wiley and Sons US.

Koh, E. C. Y., Caldwell, N. H. M., & Clarkson, P. J. (2012). A method to assess the effects of engineering change propagation. *Research in Engineering Design*, 23(4), 329–351. https://doi.org/10.1007/s00163-012-0131-3

Pahl, G., & Beitz, W. (1977). *Konstruktionslehre*. Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-02288-7

Pahl, G., Beitz, W., Feldhusen, J., & Grote, K.-H. (2007). *Engineering Design: A Systematic Approach*. Springer London. https://doi.org/10.1007/978-1-84628-319-2

Poulsen, V. V., Guertler, M., Eisenbart, B., & Sick, N. (2025). Advancing systems engineering with artificial intelligence: a review on the future potential, challenges and pathways. *Proceedings of the Design Society*, 5, 359–368. https://doi.org/10.1017/pds.2025.10050

Roth, Karlheinz. (2000). *Konstruieren mit Konstruktionskatalogen : Band 1: Konstruktionslehre*. Springer Berlin Heidelberg : Imprint : Springer. https://doi.org/10.1007/978-3-642-17466-7

Stone, R. B., & Wood, K. L. (2000). Development of a Functional Basis for Design [Article]. *Journal of Mechanical Design.*, 122(4), 359–370. https://doi.org/10.1115/1.1289637

Tekaat, J. L., Anacker, H., & Dumitrescu, R. (2021, September 13). The Paradigm of Design Thinking and Systems Engineering in the Design of Cyber-Physical Systems: A Systematic Literature Review. ISSE 2021 - 7th IEEE International Symposium on Systems Engineering, Proceedings. https://doi.org/10.1109/ISSE51541.2021.9582548

Timperley, L. R., Berthoud, L., Snider, C., & Tryfonas, T. (2025). Assessment of large language models for use in generative design of model based spacecraft system architectures. *Journal of Engineering Design*, 36(4), 550–570. https://doi.org/10.1080/09544828.2025.2453401

https://doi.org/10.1017/pds.2026.10591 Published online by Cambridge University Press