Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 268 (2025) 348–356

Procedia Computer Science

www.elsevier.com/locate/procedia

Complex Adaptive Systems Conference with Theme:
Transdisciplinary Systems and Solutions for Adaptability, CAS 2025

# Generative AI for the Design of Complex Systems: Roadmap and Space Mission Case Study

Ramón María García Alarcia[a]*, Alessandro Golkar[a]

[a]*Department of Aerospace and Geodesy, Lise-Meitner-Strasse 9, Ottobrunn, 85521, Germany*

**Abstract**

This paper sets the foundation for a roadmap aimed at improving the efficiency of designing complex systems by leveraging generative AI. Our focus is on integrating generative AI, specifically LLMs, with semantic system models and computation engines to improve system design capabilities. We introduce a prototype assistant tool for the design of space systems, validate it and demonstrate an improvement in the efficiency of the design process with a comprehensive case study, that includes the design of Earth Observation missions benchmarked to the European Sentinel infrastructure and the EventSat mission we are developing. The tool generates, among others, typical design artifacts such as System Requirements Specifications, Product Breakdown Structures, and Interface Control Documents. We validate it through the cosine similarity of the produced text to the ground truth, a design structure matrix to compare system dependencies and strength and manual evaluation with experts. Validation results align closely with the ground truth. Design time and associated development costs are significantly reduced when using the tool, promising an improvement in efficiency. We contextualize our work in the broader context of developing generative AI for the design of complex systems, by presenting a technology roadmap with our vision of future developments. We discuss the potential of incorporating generative AI into the design of complex systems, the status of technology and the necessary developments. Generative AI integration will enable efficient complexity management, paving the way for innovative advancements in systems engineering.

© 2025 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the Complex Adaptive Systems Conference 2025 (CAS 2025)

*Keywords:* generative artificial intelligence; complex adaptive systems; space systems; engineering design

* Corresponding author. Tel.: +49-89-289-55752
*E-mail address:* ramon.garcia-alarcia@tum.de

1877-0509 © 2025 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0)
Peer-review under responsibility of the scientific committee of the Complex Adaptive Systems Conference 2025 (CAS 2025)
10.1016/j.procs.2025.08.213

## 1. Introduction

Millions of people are already using generative Artificial Intelligence (AI) daily for many of their personal and professional tasks, improving their efficiency and focusing more on what matters to them [1]. Using generative AI models that process and generate text, pictures, and video for single-bound tasks is already taken for granted. However, more fundamental work remains to enable generative AI to complete more complex tasks that require multiple interactions with the environment, iterations, or compound reasoning. This is especially true for tasks that are highly technical and underrepresented in publicly available data, such as the ones pertaining to the design of complex systems. It will not be enough to have better and larger AI models, based on more data of higher quality. We will also need appropriate frameworks and guidelines for their utilization. Our work in this area aims to advance the research and development of trustworthy generative AI for engineering design. Incorporating generative AI into engineering design and systems engineering is an active area of study. The majority of works focus on utilizing Large Language Models [2] (i.e., "large-scale, pre-trained statistical language models based on neural networks" [3]). For instance, to support the complete process of designing technical systems in chip design [4] and in electronics [5]. LLMs are being used to support certain methodologies or tools of systems engineering, such as trade-space exploration [6] or working with Design Structure Matrices [7]. Or, more generally, to facilitate recovering information from past designs [8], and support work on requirements [9] [10]. Additionally, some commercial tools for systems engineering like Valispace are already adding LLMs to support requirements elicitation [11]. It is in this area, specifically in the design of space systems, that we ground our work. There is additional motivation for our work applied to the design of space systems. Strong reductions in launch costs, increases of launch capacity, and a relatively stable launch rate are expected with the new Starship rocket, bringing a cost-efficiency improvement of 1.6 orders of magnitude compared to the current Falcon 9 [12] [13] [14, p. 9] [15]. With this, the size and the weight of the satellites will take a secondary role, and it will be cheaper to launch something, learn in orbit, and replace orbital assets when needed. Generative AI-based design tools have the possibility to improve the agility with which current designs are made and the variability of the designs themselves for a more fast-paced design process.

In this paper, we present a methodology to apply generative AI to the design of complex systems and develop a tool which applies the methodology to the design space missions. The tool, from a mission statement expressed in natural language, subsequently produces system requirements, technical specifications, and design elements like system budgets, a Product Breakdown Structure (PBS), and an Interface Control Document (ICD), what corresponds to tasks in the design process phases 0, A, and partially B of the European Cooperation for Space Standardization (ECSS) standards. We present the results on the validation of the tool, as well as evaluate the improvement in cost-effectiveness brought by its usage in the design process. Through a case study on the design of three space missions, we demonstrate an improvement in efficiency in the portions of the design process covered by our tool, through a significant increase in the design speed, reduction of development costs, and outputs of overall acceptable quality. We contextualize these results in a broader roadmap for the development and adoption of generative AI-based methods and tools to support the design of complex systems, improving the efficiency of the design process. The main research questions are the following: How can we implement generative AI for the design of complex systems to reduce design time and enhance cost-effectiveness? And what are the prospects for the technological development and application of generative AI to the design of complex adaptive systems?

The remainder of the paper is organized as follows. In Section 2, we present our methodology to leverage generative AI for the design of complex (space) systems by proposing an integrated design assistant tool and diving into its three components -Large Language Model, semantic system models, and computation engines- and their integration. In Section 3, we present the case study we use to validate our tool and evaluate the cost-effectiveness of our methodology, based on the European Sentinel Earth Observation missions and an internally developed space mission. In Section 4, we present the results of the validation and cost-effectiveness, and an emerging technology roadmap. We conclude in Section 5 with remarks on the presented work as well as the future steps to be taken.

## 2. Methodology

### *2.1. Proposed tool architecture and design flow*

Early on during our research, we demonstrated that an off-the-shelf Large Language Model (LLM) is not suited by itself to support engineering design activities [16]. One of the issues with off-the-shelf LLMs is that they are

untrustworthy: they can generate texts that might read well, but that are factually incorrect (a behavior known as hallucinations [17]). Furthermore, LLMs struggle to point to individual sources that informed their generated text. These limitations mean that LLM-based tools must be used for small increments so that their outputs can be verified by users. Another issue with off-the-shelf LLMs is that they struggle to produce meaningful answers for areas of knowledge that are underrepresented on the web, such as complex systems like space systems. Moreover, since LLMs are fundamentally stochastic engines that predict the most likely word coming next in a sentence, they are also not suited for doing numerical calculations, applying deterministically equations or first principles.

Considering these limitations of off-the-shelf LLMs, we propose a new architecture for a tool that couples a fine-tuned LLM together with semantic models of the complex system under design. This provides the LLM with structured input and output formats, and the possibility of learning ontologies in the context of the conversation, reducing hallucinations and mitigating the lack of examples in the training data. We also propose adding a computation engine able to handle the numerical calculations for the system under design, through the application of the necessary equations. While our tool was specifically developed for space missions, this generic architecture can be applied to tools designing other complex systems by integrating other domain-specific semantic models and computation engines. [16]

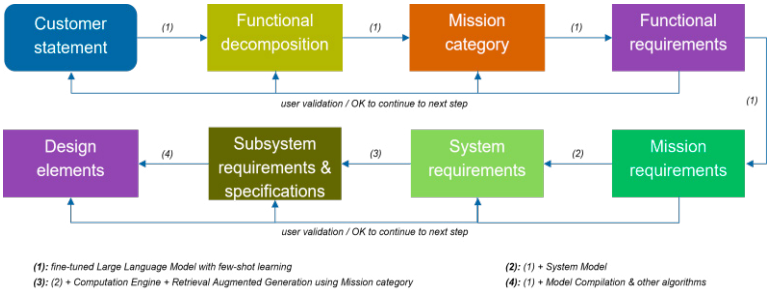



Fig. 1. Flow diagram of the design sequence implemented by our tool.

We also propose a design flow for the tool to follow, more veered towards space missions. Although requiring further demonstration the flow could be adapted, due to its commonalities, to the design of other systems also relying on systems engineering methods. **Error! Reference source not found.** depicts in a flow diagram the design f low that we envisioned: the tool starts with a high-level mission statement that further decomposes into functional requirements (what the system does), mission requirements (including cost and performance), system requirements (first level of technical requirements), and subsystem requirements (at a component level). From the technical specifications, linked to subsystem requirements, design elements such as a Product Breakdown Structure (PBS) in which certain components are already selected from available manufacturers, and an Interface Control Document (ICD) depicting the most common interfaces between subsystems (mechanical, electrical, logical, etc.) can be already created. With this, we achieve most of the technical design milestones that correspond to the European Cooperation for Space Standardization (ECSS) space mission lifecycle phases [18] 0, A, and, partially, B. As **Error! Reference source not found.** depicts, there is a go/no go decision point at each stage in which d ocumentation is being produced. In the early implementation of the tool that we have done to run the case study, the fine-tuned LLMs and the Retrieval Augmented Generation (RAG) capabilities mentioned in **Error! Reference source not found.** are not yet implemented. Thus, in this work, a generic LLM (in particular, gpt-4o) is used, and the LLM is not capable yet of retrieving relevant data from past missions given the mission categorization.

Our methodology differs fundamentally from other traditional methodologies found in mathematical programming or trade-space exploration. While in mathematical programming there is a need to very thoroughly specify the objective functions and constraints to obtain an optimal solution, in our methodology the design objectives and constraints are implicit in the user's mission statement and are limited by the design knowledge learned by the LLM from its training corpora. Additionally, the solution might not be optimal. The design problem can be specified and set up relatively easily, but with the pitfall of a lack of transparency in the solution process. Our methodology does not aim to perform a trade-space exploration, and it will result in a solution (or a series of solutions, if requested)

without necessarily exploring the complete design solution space. An exploration of the design space could eventually be produced by setting a high LLM temperature to increase stochasticity and running multiple instances. LLMs are already being explored to support trade-space exploration [6]. Our methodology is more useful, compared to others, when used in design tasks requiring the production of text-based design artifacts. These artifacts are created with high quality in an automated and quasi-instantaneous fashion, whereas classical methods require their manual curation. The methodology is limited by the black-box behavior of the underlying LLM, the dependance on a properly defined mission statement and the LLM domain knowledge. The applicability is also limited to design problems based on systems engineering and captured by the system models.

### *2.2. Semantic system models and computation engines*

We have built a generic system model of a space mission using Object-Process Methodology (OPM) [19]. This is due to OPM's convenient bimodal expression of models through a graphical Object-Process Diagram (OPD) and a textual Object-Process Language (OPL). OPL can be processed and generated by an LLM. Compared to other language representations of system models, such as the ones generated by SysML, OPL is cleaner and less verbose, fitting better the limited context windows of some LLMs. In previous work, we have described the structure and contents of the generic model, which includes a space segment (consisting of one or several spacecraft and their subsystems), ground segment (consisting of one or several ground stations and mission control infrastructure), and user segment (if existing, user terminals in which the space service is consumed), and we have studied the effects of its integration with an LLM [20]. In our setup of the design assistant tool, the generic space mission model is transformed by the LLM, which produces a specific model that includes numerical values for the subsystems and component parameters. The generated OPL is the basis for the further development of design elements, such as system budgets or artifacts like the Product Breakdown Structure (PBS) or the Interface Control Document (ICD). While integrating semantic models can improve the accuracy of the design results, it limits the applicability of the methodology to the systems represented by the model, and it can also limit the diversity of the design options that can be obtained. If generative AI becomes more accurate in the future, the trade-off between accuracy increment and diversity loss will be an important factor to consider when integrating semantic models.

We have developed the computation engine as a piece of software that includes a series of functions that implement physics equations, engineering first principles, and other algorithms to calculate the numerical parameters of space systems. The list of functions is shared with the LLM, which can request at any time, when producing an answer, that a function with certain input parameters is called. The computation engine system gives back the output of the function, a numerical value, to the LLM, which integrates the value into its natural language response. We developed the computation engine in the high-level, compiled programming language Rust, mimicking the structure and the properties of the previously presented system model of a space mission in an object-oriented programming (OOP) way [21]. Our method is extensible to other applications.

## 3. Case Study

Our tool is evaluated in a case study run for the design of four different missions: first, the Sentinel 1 [22] and 2 [23] Earth Observation missions from the Copernicus program of the European Commission. Each mission consists of a pair of satellites launched respectively in 2014, 2015, and 2016. Second, the EventSat mission, currently under development by our team at the Chair of Spacecraft Systems of the Technical University of Munich (TUM) [24]. The reason for choosing these missions is the availability of a higher amount of public information (in the case of the Sentinel missions) or internal one (in the case of EventSat), which is needed in the frame of this case study. It is difficult to find very detailed, public information about the design of specific satellites. The reason is that the design of a spacecraft, even when it is for publicly funded missions, is awarded to industrial players who are wary of disclosing industrial information that can result in competitive advantage. A reason for choosing these Sentinel missions is that they have been built by two different contractors but for the same customer and within the same Earth Observation capabilities framework. Thus, they represent the natural variability that one can find in the design of spacecraft, even when the different solutions aim to respond to similar or related problems.

The case study is based on generating three typical artifacts of the spacecraft design process: a System Requirements Specification (SRS), a Product Breakdown Structure (PBS), and an Interface Control Document (ICD). For each of the four missions, these artifacts are either directly available online (for instance, the SRS for the Sentinel satellites -

named “mission requirements document”- are readily available [25]) or are easy to compile from information generally available online. The case studies are run by requesting the previously introduced tool to generate an SRS, PBS, and ICD for each of the four missions. The tool’s input is a detailed mission statement (a description including goals and objectives) for each mission. The design artifacts are requested sequentially, and the conversation context is always preserved, so that when producing a more advanced artifact, the LLM has the previously produced ones. This aims to mimic the way a design process is conducted, having deeper-level artifacts depend on, respect, and be traceable back to higher-level artifacts. Once the design session has ended, all the output is saved, and the three artifacts are then independently compared by different methods to the ground truth, which has either been directly fetched from the source or reconstructed from available information. Additionally, we let a group of experts, in a double-blind scenario, review both the tool-generated design artifact and the ground truth one and grade each according to different subjective quality metrics. This allows us to compare how closely the tool produces designs similar conceptually to real designs, and how good the tool-generated designs are by themselves, without any comparisons. This serves as a validation of the tool. In Section 3.1, we explain the validation process in depth. Then, we estimate the design time and the development costs. This serves to evaluate whether our methodology and tool effectively improve the efficiency of designing complex systems such as spacecraft. We explain this process in Section 3.2.

### *3.1. Validation of key design artifacts*

The SRS for the Sentinel missions are available online in PDF format [25]. Requirements are manually extracted from the documents and stored in text files. The content of the text files is converted to sentence embeddings (vector representations in which sentences with similar meanings are mapped into vectors with small Euclidean distances between each other). We perform the same for the tool’s output. Then, we compute a cosine similarity metric between the sentence embeddings of the tool’s output ( $\vec{O}$ ) and the references ( $\vec{R}$ ), as per $s_c = \frac{\vec{o} \cdot \vec{R}}{\|\vec{o}\| \|\vec{R}\|}$. The result is then a number that when approaching 0 indicates vector orthogonality, and thus words of opposite meaning, and when approaching 1, identical vectors, hence words with the same meaning. The PBS for the Sentinel missions is recreated from the information available online from ESA, mostly regarding the spacecraft systems, components, and the number of them. We also created a PBS for EventSat with our internal information. The PBS is eminently a graphical artifact, where the structure of a product -in our case, the satellite- is represented via a series of interconnected blocks, offering a decomposed view of the system. The issue with this modality of data is that it is hard to compare and that the core of our tool, the LLM, only produces text. This is why we opted to represent the PBS as Object-Process Diagrams (OPD) [19], which have a natural language representation, the OPL. To validate the output of the tool, we use the cosine similarity metric again, this time between the tool’s generated OPL, and the OPL generated from the OPM representation of the PBS.

The ICD is a document fundamentally containing text and tables to describe the interfaces between the components of a system. From this perspective, it would also be possible to apply cosine similarity techniques to validate the tool’s output. However, when done this way, only information about the overall conceptual similarity is obtained, not about the actual dependencies, and particularly their strength. The latter is typically encoded numerically, which precludes the usage of word embeddings. This is why we have decided not to follow the previous approach for SRS and the PBS validation. For this artifact, we construct a Design Structure Matrix (DSM) [26] per case study and interface type, depicting the available subsystems in rows and columns. In the off-diagonal elements of the matrix, we find non-zero numerical values that depict the strength of the dependencies. Due to the scarcity of deep reference design data, we consider only the following high-level dependencies: mechanical, electrical, and digital. For the mechanical interfaces, we consider weight as the main contributor; for electrical ones, power consumption; and for digital ones, data rate. The LLM component of our design assistant tool is requested to encode the ICD information in the form of a mechanical, electrical, and digital DSM after having generated the ICD artifact. We manually build DSMs for reference missions. Then, the DSMs are compared using matrix similarity metrics. We use a normalized matrix distance metric to compare the DSMs, as per $d(\vec{o}, \vec{R}) = \frac{\sum_i^n \sum_j^n |o_{ij} - n_{ij}|}{\sum_i^n \sum_j^n |n_{ij}|}$.

### *3.2. Estimation of design time and incurred development cost*

We estimate for each case study the design time by measuring the time that the tool runs, from the insertion of the user statement to the production of the last design artifact, i.e., the ICD, and the time needed for human inspection of

all produced output, adding a margin to account for potential mistakes corrections. We set the margin to be the average cosine similarity value. The design time is then $T_D = M(T_R + T_I)$. To estimate the development costs of the generated designs, we use the parametric cost estimation process from SMAD [27]. From these values, we can extract considerations on the cost-efficiency of the system design. For this analysis, we have excluded the EventSat mission due to the premature development status preventing such comparisons.

## 4. Results and discussion

### *4.1. Validation of the tool-generated design artifacts*

**Error! Reference source not found.** and Table 2 show below the results of the objective and subjective evaluation of the tool-generated design. The cosine similarity values show that the tool-generated SRS and PBS are very similar conceptually to the reference ones, with score values at or above 0.85. A slight decrease in similarity is observed for the PBS, which can be expected due to an increased level of technical detail. The matrix distance values show worse results for the ICD artifact, depicting a wider variability with interface intensities being offset in some cases by more than 100% with respect to the reference, and suggests more work is needed at the tool level to address the challenges brought by a more complex design artifact. This shortfall may have been avoided if the Retrieval Augmented Generation (RAG) module introduced in the methodology had been developed in the version of the tool used in this work. It could palliate the lack of domain knowledge about space components and their typical interfaces and values. However, the validation results are already positive in the higher-level artifacts and tasks. The results of the subjective validation, performed by a small team of six people from the Chair of Spacecraft Systems, are also positive. They cover only the SRS and PBS artifacts. As seen below, generative AI-based SRS and PBS excel in understandability when compared to real references. While the artifacts are a bit ambiguous, probably due to the tendency of the LLM to stay at a higher level, they are seen as correct and realizable, as well as traceable to previous knowledge.

Table 1. Results of the objective, metric-based evaluation of the design artifacts.

| | **System Requirements Specifications** | | **Product Breakdown Structure** | | **Interface Control Document** | | | |
|---|---|---|---|---|---|---|---|---|
| Case Study | Metric | Value | Metric | Value | Metric | Value | | |
| | | | | | | Mech. | Elec. | Data |
| Sentinel 1 | Cosine similarity | 0.915 | Cosine similarity | 0.876 | Matrix dist. | 1.078 | 1.167 | 1.555 |
| Sentinel 2 | Cosine similarity | 0.906 | Cosine similarity | 0.908 | Matrix dist. | 0.768 | 1.029 | 0.905 |
| EventSat | Cosine similarity | 0.849 | Cosine similarity | 0.848 | Matrix dist. | 1.000 | 1.869 | 1.000 |

Table 2. Results of the subjective, expert-based evaluation of the design artifacts.

| **System Requirements Specifications & Product Breakdown Structure (Aggregated)** | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Case Study | Understandability (0 – 5) | | | Unambiguity (0 – 5) | | | Correctness & Realizability (0 – 5) | | | Traceability (0 – 5) | | |
| | Tool | Ref. | Δ | Tool | Ref. | Δ | Tool | Ref. | Δ | Tool | Ref. | Δ |
| Sentinel 1 | 4.50 | 3.17 | +1.33 | 3.17 | 2.83 | +0.34 | 3.58 | 3.17 | +0.41 | 3.67 | 3.25 | +0.42 |
| Sentinel 2 | 4.17 | 2.83 | +1.34 | 2.83 | 2.91 | -0.08 | 3.25 | 2.83 | +0.42 | 3.17 | 2.92 | +0.25 |
| EventSat | 4.33 | 3.92 | +0.41 | 2.90 | 2.67 | +0.23 | 3.50 | 3.00 | +0.50 | 3.60 | 3.25 | +0.35 |

### *4.2. Evaluation of time- and cost-efficiency*

Table 3 below shows the results of the evaluation of time- and cost-efficiency in the case study. The figures are only produced for Sentinel 1 and 2 designs of the case study, since not enough information is available at the current stage of development of EventSat to make a proper comparison.

Table 3. Results of the evaluation of time- and cost-efficiency.

| **Case Study** | **Tool runtime** | **Number of words** | **Reading time** | **Margin** | **Total time** | **Development cost** |
|---|---|---|---|---|---|---|
| Sentinel 1 | 6 min 10 s | 526,149 | 37 min | 10.45% | 48 min | 19M FY2000$ |
| Sentinel 2 | 6 min 57 s | 676,613 | 47 min | 9.3% | 1 hour | 28M FY2000$ |

Regarding the time efficiency, since design phases 0 and A typically last for months, the results promise improvement. However, a better understanding should be achieved in the evaluation of real design time when using generative AI-based tools. This can come from the direct use of prototype tools by industry and space agencies. On the other hand, the low obtained development costs (compared to development costs of the reference space missions, which are one order of magnitude above) call for more research both on the cost estimation models as well as on the generation of the model's key parameters. While lower design times will imply lower costs, the gap between the estimated costs and the real ones is likely to close more with a higher degree of model fidelity.

*4.3. Technology Roadmap*

The efforts of this work are contextualized against a broader technology roadmap of generative AI for complex systems design. With it, we aim to summarize the status of technology and our vision for its development in a short, medium, and long term, inspiring or helping guide future work. The roadmap is shown by Figure 2. Currently, tools are beginning to incorporate AI models to enhance their capabilities. Based on the state of current systems, AI-based design assistants will be mostly computational in nature and able to reproduce what is available to them in the dataset. The next milestone for AI-based design assistants will require improved cognitive abilities; that is the ability to create new designs well out of the scope of their training datasets, performing a certain degree of reasoning. In the 30s, as seen in Figure 2, we expect the design process to change and to be tackled multiple specialized, and fully autonomous AI agents that will be able to talk to each other and work together. Each agent would take the role of a specialty engineer, interfacing naturally with human supervisors and the environment. Finally, between the 40s and the 50s, there could be some level of integration between AI agents and human minds at a deeper biological level. We acknowledge the broad ethical and societal implications of this, for which we call for an increased dialogue between technologists and social scientists when planning for technology development at large time scales.

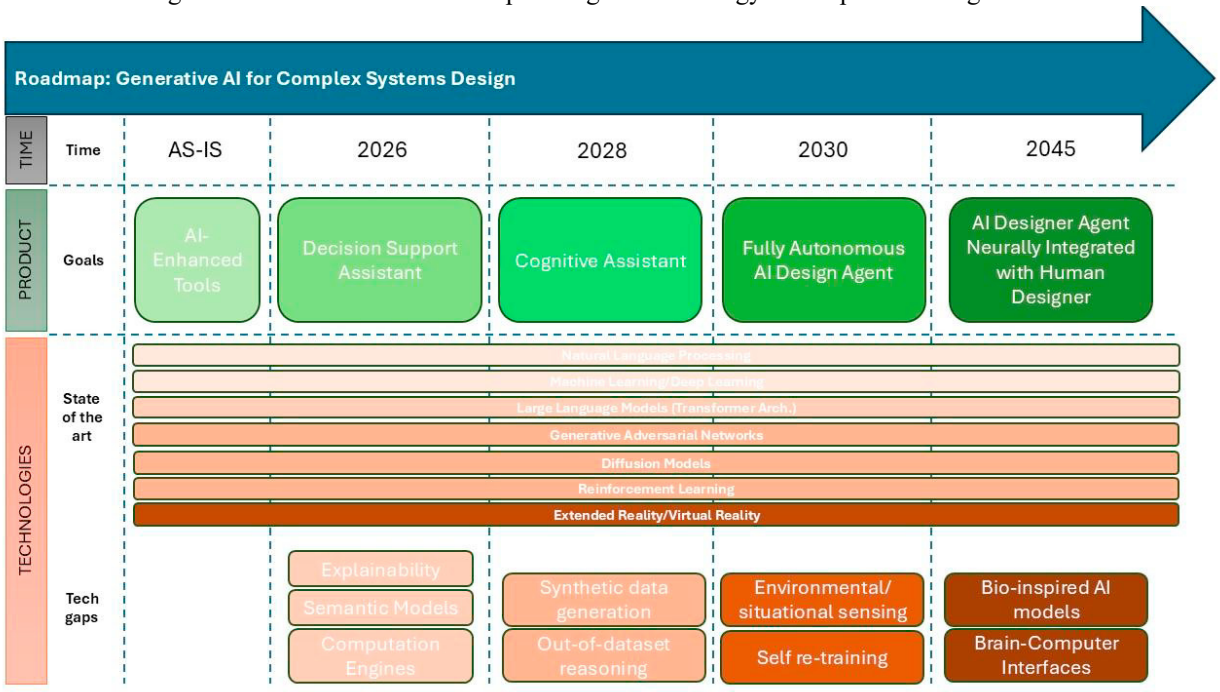



Figure 2: Technology roadmap of generative AI for Complex Systems Design.

A series of existing, AI-based technologies have already played a relevant role in engineering (e.g., Neural Networks have been widely used in varied fields and tasks [28]). Many will still be supporting technological evolution. For instance, Natural Language Processing (NLP), Machine Learning (ML) and Deep Learning (DL) models will play a role. And very particularly the Large Language Models (LLM) emerging out of training DL's

transformer architecture in vast amounts of data. Fine-tuning for increased domain knowledge and tool calling to leverage on external resources and augmenting capabilities will also be relevant. Currently, we also have Generative Adversarial Networks (a type of DL model) and Diffusion models, which will be the cornerstone to have creative cognitive assistants, in their fusion with the transformer architecture, maybe in the form of advanced hybrid architectures. Reinforcement Learning is also a relevant technology that will allow AI agents to develop policies for the environment based on the section of the environment they are trained in, making them more responsive and robust. Extended Reality and Virtual Reality (XR/VR) technologies, although already existing, still have room for maturation and will become more useful when fully autonomous agents and neurally integrated AI agents emerge, enabling better interfacing with human engineers. Again, many technologies already exist, and we expect them to mature, come together, evolve into hybridized technologies, and fundamentally back the decision-support assistant, the cognitive one, and the fully autonomous AI agents.

There are, however, a series of technology gaps that need to be filled to enable the different products. The most immediate gap to achieving a reliable decision support assistant is the explainability of LLM outputs. Two of the most important ways in which this can be achieved is the development of comprehensive and accurate semantic models of systems to be exchanged with the AI models, which will require an effort from industry and practitioners and to which we contribute partially with our work. The same applies to computation engines that will overcome the challenges with numerical computations of LLMs, and to which we also contribute. To achieve cognitive assistants, it will be necessary to improve the capabilities of generating vast amounts of quality synthetic data for training, as well as advancing towards equipping AI models with symbolic reasoning. The tech gaps are wider and harder to elicit for the next technology generations. For fully autonomous AI agents, a higher degree of environmental and situational sensing will be needed for quality interactions, and self-re-training for continuous human-like improvement of capabilities. In the far-away future, it might not be far-fetched to think that human minds might be able to use AI agents, including design agents, through tighter biological integration. To achieve this safely and successfully, bio-inspired AI models and brain-computer interfaces will be required.

## 5. Conclusion

In this paper, we have presented early findings on the application of generative AI technologies into the design of complex adaptive systems, in particular space systems, proposing one architecture for the application of LLMs integrated with external blocks such as semantic system models and computation engines, and developing a prototype design assistant tool based on this architecture. The tool can be used to support engineers in the initial phases of the design of space missions (Phases 0 and A in the ECSS terminology). It can generate design artifacts such as the System Requirements Specification, Interface Control Document, and Product Breakdown Structure starting with a detailed mission statement as an input. We have run a case study with three different space missions, validating the outputs of the LLM-based design assistant tool, and providing an initial evaluation of the incurred design time and development costs to assess improvements in the cost-effectiveness of the design process when implementing generative AI techniques into it. The results of this work show that, for higher-level design artifacts such as SRS or PBS, generative AI can produce quality outputs that resemble semantically the references and are well valued in subjective expert-based evaluations. For other design artifacts with a deeper level of complexity, such as the ICD, results still show room for improvement. Fine-tuning and RAG, two elements of our proposed framework that have not yet been implemented in this work, could enhance the domain and component-specific knowledge of the LLM, helping better understand the typical interfaces in space system components, and their typical values. From the point of view of cost-effectiveness, even though the underlying models and assumptions need further rework and maturation, and this will close the gap to real figures, the introduction of a generative AI-based design assistant tool brings a significant reduction in design time and development cost. Finally, we have introduced in this work a technology roadmap to demonstrate how we envision generative AI to be applied to the design of complex systems. Generative AI technologies are changing and improving at a very fast pace, it is, therefore, hard to accurately predict their timeline. However, we envision a not-so-far-away era in which autonomous AI agents will take an important role in designing complex systems and in which such AI capabilities will be integrated more closely with human beings. The work we have presented, and on which we intend to continue building on top, aims to contribute to the achievement of a first step: trustworthy decision support assistants to engineer space missions and other complex systems.

## References


[1] M. Skjuve, P. B. Brandtzaeg, and A. Følstad, "Why do people use ChatGPT? Exploring user motivations for generative conversational AI," *FM*, Jan. 2024, doi: 10.5210/fm.v29i1.13541.

[2] F. Chiarello, S. Barandoni, M. Perišić, and G. Fantoni, "Generative large language models in engineering design: opportunities and challenges," *Proceedings of the Design Society*, vol. 4, pp. 1959–1968, May 2024, doi: 10.1017/pds.2024.198.

[3] S. Minaee *et al.*, "Large Language Models: A Survey," Feb. 20, 2024, *arXiv*: arXiv:2402.06196. doi: 10.48550/arXiv.2402.06196.

[4] M. Liu *et al.*, "ChipNeMo: Domain-Adapted LLMs for Chip Design," Apr. 04, 2024, *arXiv*: arXiv:2311.00176. doi: 10.48550/arXiv.2311.00176.

[5] R. Zhong *et al.*, "LLM4EDA: Emerging Progress in Large Language Models for Electronic Design Automation," Dec. 28, 2023, *arXiv*: arXiv:2401.12224. doi: 10.48550/arXiv.2401.12224.

[6] G. Apaza and D. Selva, "Leveraging Large Language Models for Tradespace Exploration," *Journal of Spacecraft and Rockets*, vol. 0, no. 0, pp. 1–19, doi: 10.2514/1.A35834.

[7] S. Jiang, M. Xie, and J. Luo, "Large Language Models for Combinatorial Optimization of Design Structure Matrix," Nov. 19, 2024, *arXiv*: arXiv:2411.12571. doi: 10.48550/arXiv.2411.12571.

[8] P. Darm, F. Marchetti, G. Garcia, P. M. Redondo, A. Riccardi, and G. Fernandez, "Leveraging language models semantic similarity capabilities to facilitate information reuse in system engineering".

[9] A. Berquand, P. Darm, and A. Riccardi, "SpaceTransformers: Language Modeling for Space Systems," *IEEE Access*, vol. 9, pp. 133111–133122, 2021, doi: 10.1109/ACCESS.2021.3115659.

[10] A. Tikayat Ray, B. F. Cole, O. J. Pinon Fischer, R. T. White, and D. N. Mavris, "aeroBERT-Classifier: Classification of Aerospace Requirements Using BERT," *Aerospace*, vol. 10, no. 3, Art. no. 3, Mar. 2023, doi: 10.3390/aerospace10030279.

[11] "Introducing the ValiAssistant, Your New Favourite Colleague – Valispace." Accessed: Dec. 20, 2024. [Online]. Available: https://www.valispace.com/introducing-the-valiassistant/

[12] "The Space Review: The new era of heavy launch." Accessed: Dec. 02, 2024. [Online]. Available: https://www.thespacereview.com/article/4626/1

[13] "Musk's SpaceX Targets 400 Starship Launches within Four Years | Nasdaq." Accessed: Dec. 02, 2024. [Online]. Available: https://www.nasdaq.com/articles/musks-spacex-targets-400-starship-launches-within-four-years

[14] "Falcon 9," *Wikipedia*. Dec. 02, 2024. Accessed: Dec. 02, 2024. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Falcon_9&oldid=1260776646

[15] A. A. Romera, "SpaceX closes out record-breaking 2023, prepares for more records in 2024," NASASpaceFlight.com. Accessed: Dec. 02, 2024. [Online]. Available: https://www.nasaspaceflight.com/2023/12/falcon-roundup/

[16] R. M. Garcia Alarcia and A. Golkar, "Architecture of a generative design tool for spacecraft and user front-end implementation through a chatbot smart design assistant," presented at the IAC 2023 Congress Proceedings, 74th International Astronautical Congress, 2023. Accessed: Aug. 28, 2024. [Online]. Available: https://mediatum.ub.tum.de/node?id=1724403&change_language=en

[17] G. Perković, A. Drobnjak, and I. Botički, "Hallucinations in LLMs: Understanding and Addressing Challenges," in *2024 47th MIPRO ICT and Electronics Convention (MIPRO)*, May 2024, pp. 2084–2088. doi: 10.1109/MIPRO60963.2024.10569238.

[18] "ECSS-M-ST-10C Rev.1 – Project planning and implementation (6 March 2009) | European Cooperation for Space Standardization." Accessed: Dec. 04, 2024. [Online]. Available: https://ecss.nl/standard/ecss-m-st-10c-rev-1-project-planning-and-implementation/

[19] D. Dori, *Object-Process Methodology: A Holistic Systems Paradigm; with CD-ROM*. Springer Science & Business Media, 2002.

[20] R. M. García Alarcia, P. Russo, A. Renga, and A. Golkar, "Bringing Systems Engineering Models to Large Language Models: An Integration of OPM with an LLM for Design Assistants:," in *Proceedings of the 12th International Conference on Model-Based Software and Systems Engineering*, Rome, Italy: SCITEPRESS - Science and Technology Publications, 2024, pp. 334–345. doi: 10.5220/0012621900003645.

[21] R. M. Garcia Alarcia and A. Golkar, "A computation engine for numerical system requirements generation in LLM-based spacecraft design assistants," presented at the 75th International Astronautical Congress (IAC), 2024.

[22] "Copernicus: Sentinel-1 - eoPortal." Accessed: Dec. 04, 2024. [Online]. Available: https://www.eoportal.org/satellite-missions/copernicus-sentinel-1

[23] "Copernicus: Sentinel-2 - eoPortal." Accessed: Dec. 04, 2024. [Online]. Available: https://www.eoportal.org/satellite-missions/copernicus-sentinel-2

[24] L. Schuberth *et al.*, "LEVERAGING EVENT-BASED CAMERAS FOR ENHANCED SPACE SITUATIONAL AWARENESS: A NANOSATELLITE MISSION ARCHITECTURE STUDY," presented at the 75th International Astronautical Congress (IAC), 2024.

[25] "ESA - Scientific and technical mission documents." Accessed: Dec. 04, 2024. [Online]. Available: https://www.esa.int/Applications/Observing_the_Earth/FutureEO/Preparing_for_tomorrow/Scientific_and_technical_mission_documents

[26] S. Eppinger and T. Browning, *Design Structure Matrix Methods and Applications*. 2012. doi: 10.7551/mitpress/8896.001.0001.

[27] W. J. Larson and J. R. Wertz, *Space Mission Analysis and Design*. United States: Torrance, CA (United States); Microcosm, Inc., 1992. [Online]. Available: https://www.osti.gov/biblio/7369177

[28] A. Y. Alanis, N. Arana-Daniel, and C. López-Franco, *Artificial Neural Networks for Engineering Applications*. Elsevier, 2019. doi: 10.1016/C2018-0-01649-7.