AI- and Data Science-Driven Decision Support: Transforming Early Product Creation and Collaboration in Product Development

Michael Tatschl[1]*, Gerolf Schalk[1,2], Hannes Hick[1]

[1]Graz University of Technology, Institute of Machine Components and Methods of Development, Austria
[2]Siemens Industry Software GmbH, Science, Academics & Transfer
*Corresponding author. E-Mail address: michael.tatschl@tugraz.at

Abstract:

Early product creation involves decision-making under uncertainty, driven by heterogeneous customer inputs, multidisciplinary dependencies, and iterative stakeholder coordination. Translating informal customer needs into precise system definitions remains a significant challenge. This paper proposes an AI- and data science (DS)-driven automation framework that integrates customer-interactive, human-in-the-loop requirements engineering with automated system concept development and evaluation. The approach is validated using an unmanned aerial vehicle use case in an industrial-grade digital engineering environment. The results demonstrate how integrated AI- and DS-supported workflows can improve efficiency, decision-making quality, coordination, and traceability in early product creation.

Keywords:
Engineering Automation, AI- and Data Science- Supported Product Development, Human-Machine Interaction, Decision-Support, Systems Engineering

1 Introduction and Problem Statement

The development of complex systems is characterized by a vast solution space in which numerous design variants may satisfy a given set of system requirements. Generating initial concepts, comparing them, and determining the most promising solutions within this space is particularly challenging due to complex dynamics, high uncertainty, adaptive decision-making behavior, and collaborative design iterations [1]. As a result, the definition of initial system concepts remains a largely manual and time-consuming activity, despite rapid advances in digital engineering.

A major source of uncertainty in early product creation lies in requirements engineering (RE) as a starting point of development. Customers typically express their needs in incorrect, ambiguous, and incomplete requirements in natural language, making it problematic for deriving technically precise and machine-interpretable requirements [2]. Engineers must therefore invest significant manual effort in interpreting, refining and validating these inputs before they can be used for downstream engineering activities such as system modeling and system design conception. This disconnection between customer intent and unambiguous, formal system representation limits the extent to which RE activities can be automated in early product creation [3]. Recent advances in data science (DS) and artificial intelligence (AI) offer substantial potential to support engineering activities. In particular, natural language processing (NLP) and AI techniques have been increasingly applied to support RE tasks. However, many existing approaches focus on isolated tasks and are not embedded into integrated engineering workflows. As a result, the overall impact of AI-based techniques often remains limited due to technical and organizational challenges, hindering the establishment of coherent automation workflows. [4]

This paper addresses this gap by proposing an AI- and DS-supported automation framework that seamlessly connects customer-driven RE with automated system design conception. This approach employs an AI-assisted requirements processing pipeline to transform customer-provided inputs into structured, consistent, and partially parameterized system requirements. These requirements serve as the foundation for the automated generation and evaluation of system design concepts, enabling rapid exploration of feasible solutions and providing a structured basis for coordination and decision-making between customers and engineers. The contribution of this work lies in demonstrating how DS- and AI-based approaches can be orchestrated within an automation framework for early product creation, with a particular emphasis on RE and automated system concept development. By focusing on technical workflow integration rather than isolated technical implementations, the proposed approach supports hybrid intelligence in product creation, where human expertise as well as DS- and AI-based methods jointly contribute to informed decision-making. Accordingly, this paper is positioned within the HIPPE track "*DS/AI Interaction on Product Creation*."

2 Related Work and Background

This chapter explores the state of the art in engineering, particularly in the automation of RE and product design concept development. Research in these fields has gained significant momentum in recent years, driven by advances in NLP, DS, AI, and data-driven analysis techniques, as discussed subsequently.

In current RE literature, most concepts focus on automating isolated tasks such as requirements elicitation, classification, consistency checking, and ambiguity detection [5]. While these approaches improve the efficiency and quality of individual activities, they typically operate as standalone solutions and are not embedded into a broader automation framework spanning early system conceptualization. In parallel, only few research efforts address the interaction of intuitive, interactive user interfaces (UI) [6] that support human-machine interaction during requirements formulation, analysis, and refinement. To address this gap, a dedicated multi-step RE agent is introduced in this paper. This agent, similar to suggestions from recent work [7], combines DS and AI capabilities to ensure structured input for subsequent design activities. Moreover, direct interaction with customers' requirement repositories in Application Lifecycle Management (ALM) systems is rarely considered [8], limiting the reuse of existing requirements data as well as hindering seamless transitions and data flow during system conceptualization. For the automation of system concept development, structured inputs must be represented and managed in a formalized manner to ensure continuity and traceability across the individual process stages. In this context, system models as employed in Model-Based Systems Engineering (MBSE) offer a rigorous means to formally capture system structure, behavior, and parametric relationships, and are therefore chosen as an information backbone for automated design workflows in this paper [9]. Within the system concept design phase, the resulting design space is systematically explored to identify feasible system configurations that satisfy requirements while adhering to technical and disciplinary constraints. Prior research has proposed a range of approaches that integrate MBSE with multidisciplinary design analysis and optimization methods to address the inherent complexity of such tasks [10, 11]. These contributions emphasize holistic frameworks for system concept development, enabling the coordinated consideration of multiple engineering domains and supporting the identification of optimized design solutions.

As a result, a central research gap emerges at the intersection of these research streams. While AI- and DS-based approaches in RE primarily focus on the linguistic and semantic processing of individual requirements, and model-based approaches in system design and optimization address the quantitative evaluation of predefined solution spaces, their integration into a coherent, end-to-end automation framework remains insufficiently addressed. Recent review work highlights that AI-based methods are often applied in a fragmented manner across different phases of product development, leading to siloed insights and a lack of continuity across product creation phases [12]. Against this background, existing work provides limited insight into how human-centered requirements definition can be systematically combined with AI-based analysis and automated system conceptualization within a single, continuous workflow. This gap is especially relevant for early product creation phases, where effective coordination and decision-making between customers and engineers are critical [1]. Without an integrated automation logic, requirements remain detached from system concept generation, limiting the applicability of DS- and AI-based methods to support tools rather than comprehensive product conception solutions.

This paper addresses this gap by proposing an AI- and DS-supported automation framework that integrates interactive RE and automated system design concept generation, rather than introducing new isolated algorithms. With the objective of developing a universally applicable AI- and DS-supported automation framework incorporating human-machine interaction via a UI enabling customer inputs to be iteratively refined, structured, and transformed into actionable input for automated system conceptualization, the following research question is investigated: "*How can RE and system conceptualization processes be arranged interactively and automated to support the coordination and decision-making in early product creation, and how can AI and DS be leveraged to support these processes?*"

3 Research Methodology

This paper addresses the research question by proposing a universally applicable AI- and DS-supported automation framework for interactive RE and automated system design concept generation. The proposed framework, illustrated in Figure 1, is structured into three main steps: 1) Requirements engineering, 2) System concept development, and 3) System concept communication.

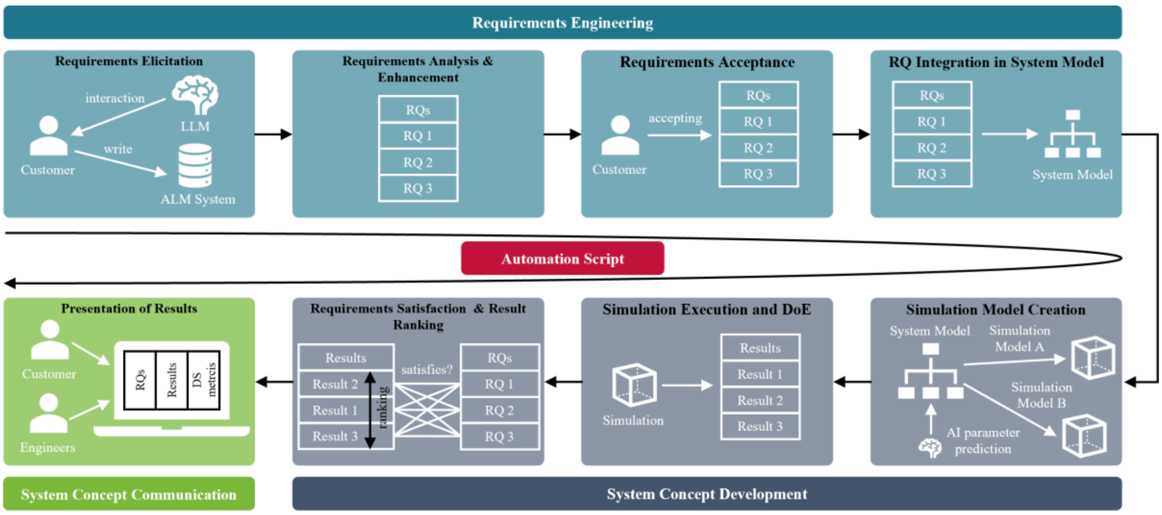


Figure 1: Automation Framework

The methodology follows a design-oriented research approach, in which a structured automation framework is developed, implemented, and validated against clearly defined objectives derived from the research question. To ensure traceability between the research question and the methodological contribution, the proposed framework is assessed based on the fulfillment of the following objectives:

- *Obj1*: The customer defines and refines requirements interactively using a machine-UI.
- *Obj2*: Customer requirements are automatically transferred into structured, correct, and unambiguous technical requirements.
- *Obj3*: System concepts covering the whole solution space are generated and evaluated against the requirements automatically.

- *Obj4*: The system concepts are ranked based on various criteria.
- *Obj5*: The system concepts are presented to the stakeholders by providing expressive metrics and visualizations highlighting trade-offs.
- *Obj6*: Activities where AI and DS approaches can be beneficially applied are identified.

Leveraging MBSE system models as a data backbone and a model-based definition of the workflow, the automation logic within a unified script gains access to all relevant engineering data in a formalized and machine-readable form. In MBSE, state-of-the-art descriptive system models created using semantically rich modeling standards provide abstraction and data traceability across system and product realization levels [13]. By using MBSE, execution of automated analysis, generation, and evaluation steps in a defined and reproducible sequence in enabled. Basis of the proposed automation is a process model, representing the workflow from Figure 1, but in more detail with several sub-workflows and input as well as output definitions. On the lowest hierarchical level, each workflow step is equipped with a script enabling its execution. The workflow is modeled in a model-based environment to make the information accessible and useable to the superior automation script, which identifies all workflow steps, applied methods, and executes the linked scripts in the defined sequence. This model-based approach enables a clear definition of the inputs and outputs needed to execute a workflow step and, additionally, allows a high variability by facilitating the substitution and complementation of the workflow or the applied methods. The data exchanged between workflow steps is equipped with metadata that defines how it should be stored and how to query it in the database.

For each workflow step, the methodology is described in detail in the following sections and assessed with respect to the fulfillment of the objectives listed above. To demonstrate practical applicability, the proposed framework is exemplified using a mechatronic system introduced in Chapter 4. While this use case provides a concrete validation scenario, the methodology itself is not limited to a specific product and is therefore applicable to a wide range of mechatronic systems.

3.1 Requirements Engineering

The goal of automated RE is to enable interactive, AI-supported definition, refinement, and structuring of customer requirements, while ensuring that the resulting requirements are consistent, machine-interpretable, and suitable as input for automated system concept development. This step primarily addresses *Obj1* and *Obj2*.

The automated RE approach combines human-machine interaction, rule-based analysis, DS, and state-of-the-art large language model (LLM)-based refinement within a single automation script. Customers interact with the system via a UI that allows them to input, review, and iteratively refine their requirements, as illustrated in Figure 2. In line with a recent human-in-the-loop RE initiative, this interaction helps to keep human intent and contextual knowledge central to the RE process, while AI- and DS-based methods support requirements formalization, validation, and traceability [14]. Customer requirements can be provided either through structured files (e.g., spreadsheet-based inputs) or via direct interfaces to a requirements repository in an ALM system, which are accessed through standardized application programming interfaces (APIs). This enables seamless reuse of existing requirements data and ensures compatibility with industrial requirements management environments. The automated requirements processing pipeline is triggered by the customer via the UI and operates on a combination of linguistic and semantic analysis techniques. Initially, requirement descriptions are analyzed using NLP methods to extract syntactic and semantic features. A DS-based classifier utilizing sentence embeddings is employed to assess the consistency and correctness of requirement descriptions by assigning the labels *consistent* or *not consistent*. A requirement is classified as consistent if it fulfills established RE principles, such as SMART criteria (specific, measurable, achievable, relevant, and traceable) and the requirements boilerplate [15].

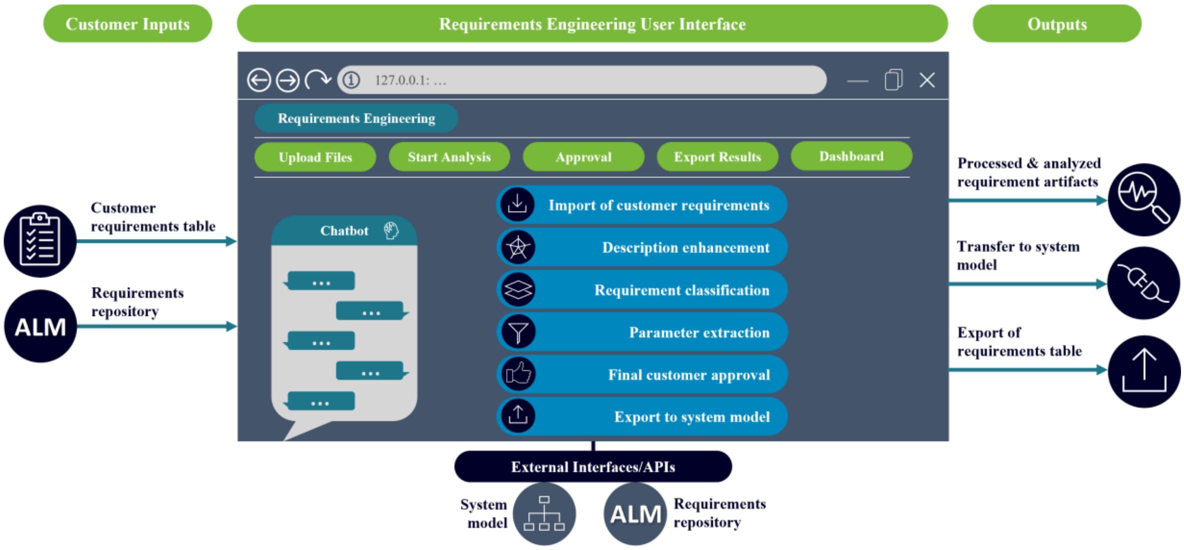


Figure 2: Requirements Engineering User Interface

To increase robustness, consistency classification results are verified by combining multiple analysis mechanisms – namely an NLP language library and a DS classifier trained on consistently and inconsistently formulated requirements – before assigning a final label. Requirements identified as inconsistent are subsequently refined using an LLM-based approach, which reformulates the requirement description in accordance with the requirements boilerplate via API-based access. However, LLM-based text generation still suffers from limited controllability, inexplicable decision-making processes, and hallucinations [16]. To address this challenge, the proposed approach applies constrained, rule-based utilization of the boilerplate to reduce generative behavior and undesired output. Subsequently, after description enhancement, the requirements are re-evaluated for consistency and, if validated, forwarded to the next processing step. In addition, the unambiguity of requirement descriptions is assessed with a correct reference requirements set which captures the context to ensure unambiguity and apply it to the customer requirements set. This context-capturing is executed via LLM-support and DS-measures such as cosine similarity to ensure high-quality customer requirements. In the next step, requirements are classified into quantitative and qualitative categories. This classification is based on predefined rule-based patterns (e.g., unit identifiers and domain-specific keywords) in combination with NLP-based analysis. For quantitative requirements, numerical values, units of measurement, and relational operators (e.g., less than, equal to, between, greater) are extracted and mapped to semantically meaningful parameter names. Finally, the customer can review the processed and analyzed requirements directly within the UI or export the results as a structured table. Upon customer approval, the requirements – together with their associated parameters – are transferred into a system model via the modeling tool's APIs. This structured and uniform transformation procedure enables the direct, model-based use of requirements and parameters as input for downstream engineering activities. Consequently, the automated RE pipeline clearly fulfills *Obj1* by enabling interactive customer involvement through a UI and *Obj2* by transforming informal requirements into structured and high-quality technical requirements by utilizing AI and DS. The resulting requirements form a consistent and machine-interpretable basis for system concept development and evaluation in subsequent workflow steps.

In addition to the core requirements processing pipeline, the UI provides analytical and interaction-oriented features that support transparency, validation, and decision-making during RE. A dedicated dashboard visualizes key indicators of the requirements analysis, such as the ratio of consistent and inconsistent requirements, the distribution of qualitative and quantitative requirements, and the criticality of requirements. These visualizations enable customers to assess the current quality, completeness, and reliability of the requirements set immediately and they can be exported as a structured report for documentation processes. Furthermore, the UI integrates an in-UI conversational assistant based on an LLM. This assistant supports users by answering questions related to the UI, general RE principles, and the analyzed requirements themselves. Rather than replacing expert judgment, this functionality serves as an interactive decision-support mechanism that enhances customer understanding and facilitates informed interaction within the automated workflow.

3.2 System Concept Development

The goal of this step is to evaluate the whole solution space and find optimized solutions while considering all requirements and technical constraints, aiming at fulfilling *Obj3* and *Obj4*. The system requirements elaborated in the step before are used as a trigger to start the automated system concept development workflow, providing - depending on the computation time - nearly instantaneous feedback on the feasibility of fulfilling the requirements and on key performance indicators of possible system concepts.

A data backbone providing semantically linked information about the requirements, the technical system and available quantitative models together with a high degree of formalization is essential to enable automatized data handling. MBSE models fulfil the requirements of formalization and are, therefore, chosen for describing the structure and behavior of the system. In this paper, the system model is a pre-created superset model, representing all potential variants of the system and its subsystems as well as the associated technical constraints within one model. The model comprises structural as well as functional descriptions of the system, extended with variation points indicating the allocation of structural blocks, behavioral blocks and parameters to a certain design variant. [9] All realizable variants of system are, therefore, defined by the content and degree of detail of the pre-created system model, establishing a bottleneck limiting the considered solution space. Requirements and quantitative models are linked to the data artifacts from the system model by integrating and relating all engineering data into an enterprise knowledge graph. In that way, the semantic linkage of data from heterogeneous sources is enabled, positioning the system model as a data backbone to make data actionable in a wider context, similar to the architecture proposed in [17]. The relationships between data points in the knowledge graph rely on a data meta model that defines how data must be structured to support the desired functionalities needed for the automation workflow. For example, system functions fulfil system requirements and can be decomposed into subfunctions. Simulation models are allocated to respective subfunctions, drawing the link from the descriptive system model to quantified analysis. If multiple alternatives exist to realize these functions, such as different variants of a subsystem, different simulation models may need to be considered. In these cases, the simulation models are further described with conditions on their applicability. Parameters, serving as input and/or output of functions, are described by value ranges and step sizes to define the possible solution space. Certain parameters cannot be described meaningfully with value ranges, such as the system mass, as they are highly dependent on other parameters. These parameters are estimated via calculations, which can either be rule-based formulas or predictive models. Technical constraints are linked to certain parameters and apply conditions on their parameter values.

Providing the customer with reliable results necessitates the full exploration of the solution space stretched by the parameters formalized in the system model. In literature, many possibilities of doing that are discussed [18]. For demonstration purposes and the sake of simplicity, full factorial design of experiments (DoE) is used in this paper. After applying full factorial DoE to generate parameter sets representing the whole solution space, these parameter sets are used as an input to system simulations quantifying the behavior of all the investigated concepts. The appropriate simulation model representing each function of a certain concept is selected based on the underlying conditions, integrated into a simulation environment and, if necessary, connected to other simulation models there. Load profiles necessary as an input for the system simulation must be provided initially or derived from the system requirements using a dedicated model. The simulation results of each concept are evaluated against the requirements, and underperforming parameter sets are dismissed. The remaining concepts are further evaluated and ranked regarding certain criteria, such as cost or mass.

The methodology proposed for the system concept development successfully achieves both objectives *Obj3* and *Obj4* by leveraging formalized data storage and modelled automation workflows to trigger the execution of quantitative models, evaluate the results against the requirements and a rule-based ranking approach. Regarding *Obj6*, neural networks in the form of predictive models to estimate high-level parameters in early stages of product development are identified to be a beneficial application of AI in this development stage.

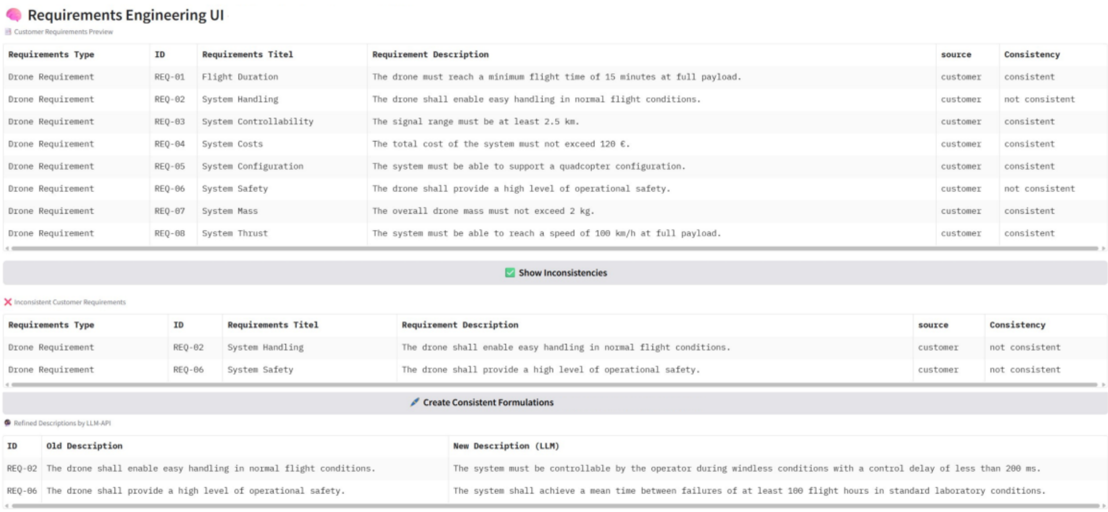
3.3 System Concept Communication

As a final step, the automatically generated and ranked concepts are communicated to the customer and engineers via a UI, providing a clear differentiation between the suggested concepts utilizing metrics regarding multiple ranking criteria. Furthermore, many different other expressive metrics, which are beneficial for decision making, can be presented - for example, an index quantifying the extent to which requirement-defined goals are exceeded, or a similarity index identifying concepts similar to previously developed systems, for which high confidence in development success can be assumed. An example of such a similarity index is the DS-based metric cosine similarity, showing the beneficial application of DS in this step for *Obj6*. Pareto fronts are used for visualizing the differences between concepts and potential trade-offs regarding key parameters, facilitating comparison. By providing stakeholders ranked results, DS-based metrics and pareto fronts, the UI facilitates comparison of solutions and provides a basis for discussions between customers and engineers, fulfilling *Obj5*.

4 Framework Application on Use Case

This section presents the application of the proposed automation framework to a multirotor unmanned aerial vehicle (UAV) use case and demonstrates its execution within an industrial-grade environment. In doing so, the feasibility of the framework in the context of early system conceptualization is illustrated. The UAV use case was selected due to its representative characteristics for early product creation, including its multidisciplinary nature and the presence of performance-related trade-offs.

For the application of the framework, a representative set of customer-defined requirements for a UAV system is provided via the RE-UI. The input comprises high-level expectations related to flight performance, handling, safety, cost, and system configuration. As illustrated in Figure 3, these requirements reflect realistic customer input encountered in early product creation and therefore include heterogeneous levels of detail as well as underspecified formulations. After the customer initiates the RE process, the UI automatically analyzes the textual descriptions and identifies requirements that violate consistency. In the presented use case, requirements related to handling and safety are flagged as *inconsistent* and are explicitly highlighted in the UI. For these requirements, the UI generates revised, boilerplate- and SMART criteria compliant formulations using an LLM-based refinement step. The proposed consistent descriptions are presented transparently to the customer, who can review, edit if necessary, and explicitly accept the updated formulations. This interaction demonstrates the human-in-the-loop character of the approach, where automated reasoning is combined with user validation rather than replacing it. Following acceptance, the workflow generates a consistent, structured set of system requirements and automatically extracts quantitative parameters in machine-readable form, such as maximum allowable mass and minimum flight duration. These requirements are transferred to the system model, where automated scripts impose the corresponding parameter constraints.



Requirements Engineering UI

Customer Requirements Preview

Requirements Type	ID	Requirements Titel	Requirement Description	source	Consistency
Drone Requirement	REQ-01	Flight Duration	The drone must reach a minimum flight time of 15 minutes at full payload.	customer	consistent
Drone Requirement	REQ-02	System Handling	The drone shall enable easy handling in normal flight conditions.	customer	not consistent
Drone Requirement	REQ-03	System Controllability	The signal range must be at least 2.5 km.	customer	consistent
Drone Requirement	REQ-04	System Costs	The total cost of the system must not exceed 120 €.	customer	consistent
Drone Requirement	REQ-05	System Configuration	The system must be able to support a quadcopter configuration.	customer	consistent
Drone Requirement	REQ-06	System Safety	The drone shall provide a high level of operational safety.	customer	not consistent
Drone Requirement	REQ-07	System Mass	The overall drone mass must not exceed 2 kg.	customer	consistent
Drone Requirement	REQ-08	System Thrust	The system must be able to reach a speed of 100 km/h at full payload.	customer	consistent

Show Inconsistencies

Inconsistent Customer Requirements

Requirements Type	ID	Requirements Titel	Requirement Description	source	Consistency
Drone Requirement	REQ-02	System Handling	The drone shall enable easy handling in normal flight conditions.	customer	not consistent
Drone Requirement	REQ-06	System Safety	The drone shall provide a high level of operational safety.	customer	not consistent

Create Consistent Formulations

Refined Descriptions by LLM-API

ID	Old Description	New Description (LLM)
REQ-02	The drone shall enable easy handling in normal flight conditions.	The system must be controllable by the operator during windless conditions with a control delay of less than 200 ms.
REQ-06	The drone shall provide a high level of operational safety.	The system shall achieve a mean time between failures of at least 100 flight hours in standard laboratory conditions.

Figure 3: Excerpt of Requirements Engineering User Interface – UAV System Use Case

Together with predefined technical constraints and all design variants and parameter ranges, a full-factorial DoE discretizes the feasible solution space. Drone mass, which cannot be meaningfully bounded by simple ranges due to strong parameter interdependencies, is estimated for each design point using a neural network trained on data from previously developed drones, aiming at improving early-stage accuracy over rule-based methods. Flight duration is evaluated via system simulation using an automatically generated flight profile derived from requirement values. All relevant models and data are managed in a knowledge graph, which is queried to assemble and execute the appropriate simulations for each parameter set. Designs that fail to meet requirements are discarded, while the remaining concepts satisfy all constraints but exhibit different performance trade-offs. For example, one concept achieves a long flight duration but at the cost of a relatively high mass, whereas another concept only marginally satisfies the flight duration requirement while exhibiting significantly lower mass. To support the identification of the most suitable solution for the customer, an UI is introduced, which can be seen in Figure 4. This UI enables a multi-criteria ranking, e.g., regarding mass and flight duration, directly by the customer or by the engineer. However, also additional metrics can be incorporated, which are evaluated separately using dedicated quantitative models. Examples for such metrics are cost, environmental impact, or DS-based metrics, including cosine similarity to previously developed products. Pareto fronts can be used as another way to visualize the trade-offs between different concepts, like the trade-off between system mass and flight duration, which is displayed in Figure 4.

UAV Concepts

Name	Mass (kg)	Flight Duration (min)	Cost (€)	Env Impa
Concept 25	1.9	28	115	0.6
Concept 4	1.5	27	110	0.25
Concept 24	1.1	26	120	0.35
Concept 1	1.8	25	120	0.3
Concept 3	2.0	25	90	0.5
Concept 9	0.5	24	115	0.3
Concept 20	0.9	25	85	0.35
Concept 15	0.7	24	110	0.35
Concept 26	2.0	24	110	0.25
Concept 11	0.3	23	80	0.3
Concept 21	0.9	22	100	0.2
Concept 6	0.3	22	100	0.5
Concept 10	0.5	22	95	0.35
Concept 12	0.3	21	110	0.4
Concept 14	0.7	21	90	0.5
Concept 16	1.2	22	100	0.3
Concept 13	0.7	20	120	0.25
Concept 19	1.3	19	90	0.3
Concept 18	1.3	19	90	0.35

Figure 4: Concept Analysis User Interface – UAV System Use Case

The proposed automation framework was implemented and executed in a controlled engineering environment, namely the Digital Lifecycle Lab (DLL) located at Graz University of Technology [19]. The DLL provides interconnected access to industrial-grade digital engineering solutions, including ALM systems, system modeling environments, and simulation tools. This integrated setup enables the reproducible execution of AI- and DS-based workflow steps under realistic engineering conditions. Within this environment, the proposed framework was validated with respect to its technical feasibility and its capability to support interactive RE and downstream automation steps. The DLL thus serves as a neutral validation environment focusing on workflow integration and methodological soundness.

5 Results and Discussion

The UAV use case demonstrates the feasibility and effectiveness of the proposed automation framework in early system conceptualization. The results show that interactive RE supported by a machine–UI enables customers to define and iteratively refine requirements in a transparent human-in-the-loop process (*Obj1*). Automated detection and LLM-based refinement of inconsistent or underspecified requirements led to boilerplate-compliant and technically interpretable formulations, while retaining user authority over final decisions. Following validation, customer requirements were automatically transformed into structured, unambiguous technical requirements and machine-readable parameters, which were consistently transferred into the system model (*Obj2*). This enabled seamless downstream automation while maintaining traceability between requirements, parameters, and generated system concepts. Based on these inputs, the framework automatically generated and evaluated system concepts covering the discretized solution space using a full factorial DoE (*Obj3*). Hybrid evaluation models were applied, combining system simulation for flight duration with a neural network–based mass prediction to enable realistic assessments in early design stages. All concepts violating requirements or technical constraints were automatically filtered out. The remaining feasible concepts exhibited trade-offs between competing criteria, such as mass and flight duration. These concepts were ranked using a dedicated UI that allows stakeholders to prioritize evaluation criteria according to their preferences (*Obj4*). To support informed decision-making, expressive metrics and visualizations, including Pareto front representations, were provided to highlight trade-offs between non-dominated solutions (*Obj5*). Finally, the use case highlights beneficial applications of AI and DS methods within the framework, including LLM-based requirement refinement, data-driven mass prediction, and similarity-based metrics leveraging a knowledge graph (*Obj6*). Overall, the results illustrate that the proposed framework effectively integrates RE, automated concept generation, AI-supported evaluation, and decision support into a coherent workflow suitable for early-stage system design.

At the same time, several limitations and opportunities for further improvement can be identified. As the current implementation constitutes a prototype, the emphasis lies on demonstrating methodological feasibility rather than achieving completeness, robustness, or industrial-scale performance. Therefore, several aspects must be further developed to ensure its applicability in an industrial context. In particular, the level of detail of both the system model and the data meta model needs to be increased to capture additional behavioral phenomena and a broader range of structural variants during early concept development. Moreover, the reliance on full factorial design of experiments is not scalable for large design spaces and should therefore be replaced or complemented by more efficient solution-space exploration strategies that balance computational effort and result fidelity. In addition, industrial applicability requires robust strategies for model selection and model coupling, as well as a systematic quantification of the uncertainties introduced by simulation-based and AI-driven models. Equally critical is the establishment of comprehensive data and model governance mechanisms, including quality assurance, version control, and traceability, to ensure reliability and reproducibility across the engineering lifecycle. From an infrastructure perspective, modeling and simulation environments must be extended to handle large volumes of heterogeneous data, and a fully integrated, API-based toolchain is necessary to support seamless automation across tools and disciplines. Finally, advanced preference modeling approaches that go beyond Pareto-front analysis and rule-based ranking, together with role-specific UIs for different stakeholder groups, are required to make results accessible, interpretable, and actionable for industrial decision-makers.

Future work is therefore required to enhance scalability, usability, and computational efficiency, and to assess the transferability of the proposed workflow to a broader range of mechatronic systems beyond the presented UAV use case. Nonetheless, the presented framework already enables an accelerated, consistent, and traceable development process, emphasizing structured communication and demonstrating the potential of AI- and DS–supported workflows to enhance early product creation.

References

[1] Fernandes, J. V., Henriques, E., Silva, A., and Pimentel, C. 2017. Modelling the dynamics of complex early design processes: An agent-based approach. Design Science 3, e19.

[2] Rosado da Cruz, A. M., and Cruz, E. F. 2025. Machine Learning Techniques for Requirements Engineering: A Comprehensive Literature Review. Software 4, 3, 14.

[3] Femmer, H., Unterkalmsteiner, M., and Gorschek, T. 2017. Which requirements artifact quality defects are automatically detectable? A case study. In 2017 IEEE 25th International Requirements Engineering Conference Workshops (REW). IEEE, 400–406.

[4] Necula, S.-C., Dumitriu, F., and Greavu-Șerban, V. 2024. A systematic literature review on using natural language processing in software requirements engineering. Electronics 13, 11, 2055.

[5] Umar, M. A., and Lano, K. 2024. Advances in automated support for requirements engineering: A systematic literature review. Requirements Engineering 29, 2, 177–207.

[6] Dąbrowski, J., Letier, E., Perini, A., and Susi, A. 2023. Mining and searching app reviews for requirements engineering: Evaluation and replication studies. Information Systems 114, 102181.

[7] Jin, D., Jin, Z., Chen, X., and Wang, C. 2024. MARE: Multi Agents Collaboration Framework for Requirements Engineering. arXiv preprint arXiv:2405.03256.

[8] Dąbrowski, J., Bennaceur, A., Rajbahadur, G. K., Nuseibeh, B., and Alrimawi, F. 2025. Prompt Me: Intelligent Software Agent for Requirements Engineering – A Vision Paper. In Requirements Engineering: Foundation for Software Quality. Hess, A. and Susi, A., Eds. Springer Nature Switzerland, Cham, Switzerland, 235–243.

[9] Object Management Group. 2023. OMG Systems Modeling Language (SysML®), Version 2.0. https://www.omg.org/spec/SysML/2.0.

[10] Bussemaker, J., Boggero, L., and Ciampa, P. D. 2022. From System Architecting to System Design and Optimization: A Link Between MBSE and MDAO. In: 32nd Annual INCOSE International Symposium. 32nd Annual INCOSE International Symposium, 2022-06-25 - 2022-06-30, Detroit, MI, USA.

[11] Habermehl, C., Höpfner, G., Berroth, J., Neumann, S., and Jacobs, G. 2022. Optimization Workflows for Linking Model-Based Systems Engineering (MBSE) and Multidisciplinary Analysis and Optimization (MDAO). Appl. Sci. 2022, 12, 5316.

[12] Witkowski, A., and Wodecki, A. 2025. Where does AI play a major role in the new product development and product management process? Management Review Quarterly.

[13] INCOSE 2021. SE Vision 2035 – Engineering Solutions for a Better World. INCOSE.

[14] Beg, A., O'Donoghue, D., and Monahan, R. 2025. Leveraging LLMs for formal software requirements – challenges and prospects. arXiv preprint arXiv:2507.14330.

[15] Maletz, M., Bajzek, M., and Hick, H. 2021. Systems Engineering Methods for Automotive Powertrain Development. In Systems Engineering for Automotive Powertrain Development. Hick, H., Küpper, K., and Sorger, H., Eds. Springer, Cham, Switzerland, pp. 402–441.

[16] Qiu, J., Fang, Q., and Kang, W. 2025. Towards controllable and explainable text generation via causal intervention in LLMs. Electronics 14, 16, 3279.

[17] Bajzek, M., Krems, D., Kollegger, S., Traußnig, T., Dörr, S., Tatschl, M., Kniewallner, J. and Hick, H. 2025. Accelerated Automotive Battery Development to meet Market Opportunities. INCOSE International Symposium, 35: 760-777.

[18] Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S. 2008. Global sensitivity analysis: The primer. John Wiley & Sons, Ltd.

[19] Siemens Digital Industries Software, "TU Graz uses Siemens Xcelerator solutions to help industries shift to interdisciplinary, sustainable product development,". accessed on Jan. 9, 2026. [Online]. Available: https://resources.sw.siemens.com/en-US/case-study-tu-graz/.