

Article

Prompt-Strategy-Driven SysML-v2 Artefact Generation Using Large Language Models for Model-Based Systems Engineering

Armin Stein ^{*}, Umut Volkan Kizgin , Niklas Waldmann , Bjarne Käberich, Souhaïel Ben-Salem, Aaron Dlugosch, Ajay Kumar Thakur and Thomas Vietor 

Institute for Engineering Design, Technische Universität Braunschweig, Herrman-Blenk Strasse 42, 38108 Braunschweig, Germany

* Correspondence: armin.stein@tu-braunschweig.de

Abstract

Large Language Models (LLMs) show growing potential for transforming natural-language engineering information into formal model-based artefacts. However, their reliability for SysML-v2 artefact generation remains insufficiently understood, particularly regarding prompt-strategy selection, model-dependent variability, and task-specific performance. This paper proposes a prompt-strategy-driven method for LLM-assisted SysML-v2 artefact generation in Model-Based Systems Engineering. The method is grounded in a systematic literature analysis and a Harvey-Balls-based assessment of existing approaches, which reveal gaps in reproducibility, prompt evaluation, and formal modelling support. The proposed workflow integrates input preparation, prompt-strategy selection, LLM selection, artefact generation, syntax validation, and quality evaluation. It is evaluated using a traction battery system case study across three representative modelling tasks: requirements generation, block definition modelling, and state-machine modelling. Four LLMs are compared using five prompt strategies and assessed through F1-score analysis and LLM-as-a-Judge evaluation. Within the investigated traction-battery case study, the observed artefact scores differed across prompt strategies and modelling tasks, and no single strategy achieved the highest observed score across all tasks. Overall, the findings indicate that LLMs can support early-stage SysML-v2 modelling as human-in-the-loop assistants, while expert validation remains necessary to ensure syntactic correctness, semantic consistency, and domain validity.

Keywords: large language models; prompt engineering; SysML-v2; model-based systems engineering; artificial intelligence; requirements engineering; systems engineering; retrieval-augmented generation; LLM-as-a-judge evaluation



Academic Editor: Isidoros Perikos

Received: 17 June 2026

Revised: 16 July 2026

Accepted: 21 July 2026

Published: 23 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

1.1. Background and Problem

Modern engineering systems increasingly combine mechanical, electrical, electronic, and software-intensive subsystems. Their development requires consistent handling of requirements, architectures, interfaces, and behavioural models across multiple disciplines in a dynamic and often changing environment and trend-landscape. Model-Based Systems Engineering (MBSE) addresses this challenge by replacing fragmented document-based descriptions with formal system models that support consistency, traceability, verification, validation, and reuse [1,2]. SysML-v2 strengthens this model-based paradigm by providing a more precise textual notation and improved semantic foundations for formal system

representation [3]. However, the creation of SysML-v2 artefacts remains difficult and time consuming because it requires both domain knowledge and modelling expertise. Large Language Models (LLMs) provide a promising opportunity to reduce this entry barrier. LLMs are a class of deep-learning-based artificial intelligence models trained on large text corpora to process and generate natural language and structured textual outputs. Their ability to process natural-language input and generate structured text or code-like outputs suggests potential for supporting formal engineering artefact generation [4,5]. In the context of MBSE, LLMs may assist engineers in transforming textual system descriptions into requirements, structural models, and behavioural models. Nevertheless, generated artefacts are not automatically reliable [6]. They may contain syntactic errors, incomplete model elements, hallucinated relations, or semantically inconsistent structures. Therefore, the use of LLMs for SysML-v2 artefact generation requires systematic prompt design, syntax validation, and task-specific quality assessment.

1.2. Research Gap

Recent studies indicate that LLMs and related AI techniques can support selected Systems Engineering and MBSE activities, including requirements analysis, model generation, model querying, and knowledge retrieval [7–9]. However, existing approaches remain fragmented. Many studies focus on isolated use cases, use only one LLM or one prompt configuration, and provide limited evidence on how prompt strategies influence the quality of generated model artefacts. SysML-v2 is also not consistently addressed, although its textual notation is particularly relevant for LLM-based model generation. To investigate this gap, a systematic literature analysis was conducted. The initial search identified 1207 studies across multiple scientific databases. After duplicate removal, title and abstract screening, full-text assessment, and snowballing, eight primary studies remained for detailed analysis. These studies were further assessed using a Harvey-Balls-based evaluation scheme covering methodological clarity, application relevance, LLM integration, MBSE relevance, SysML-v2 support, prompt engineering, tool/API integration, and reusability. The assessment shows that existing work provides valuable initial evidence, but lacks a reproducible comparison of multiple LLMs, prompt strategies, and SysML-v2 artefact types under comparable experimental conditions. Consequently, it remains unclear how prompt strategy selection affects SysML-v2 artefact quality, how different LLMs perform across requirements, structural, and behavioural modelling tasks, and which prompt–model configurations are most suitable for specific modelling objectives.

1.3. Aim and Contributions

Based on this gap, the paper investigates the following research questions:

RQ1: How does prompt strategy selection affect the quality of generated SysML-v2 artefacts?

RQ2: How do different LLMs perform across requirements, structural, and behavioural modelling tasks?

RQ3: Which prompt–model configurations achieved the highest observed scores for the investigated SysML-v2 artefact types in the traction-battery case study?

These research questions focus the study on the interaction between prompt strategy, model selection, and artefact type. This enables a task-specific evaluation of LLM-assisted SysML-v2 generation instead of treating MBSE automation as a single generic problem.

This paper makes three contributions. First, it proposes a five-phase method for prompt-strategy-driven SysML-v2 artefact generation. The method covers input preparation, prompt strategy selection, LLM selection, artefact generation, syntax validation, and quality evaluation. Second, it provides a comparative evaluation of four LLMs and

five prompt strategies across three representative MBSE tasks: requirements generation, block definition modelling, and state-machine modelling. Third, it combines syntax validation, F1-score analysis, and LLM-as-a-Judge assessment to evaluate both element-level correctness and semantic model quality. This design comprises 20 core model-prompt configurations for each modelling task, supplemented by two platform-native RAG Upload configurations for requirements generation, resulting in 62 configurations overall. It therefore provides a broader controlled comparison than studies evaluating only one model or prompt configuration. The present study focuses on Model-Based Systems Engineering; data-driven Systems Engineering is considered a complementary research direction requiring different data sources, learning methods, and validation criteria. Within the investigated case study, prompt-strategy selection was associated with task-dependent differences in artefact scores, and no single strategy achieved the highest observed score across all modelling tasks. This indicates that LLMs can support early-stage SysML-v2 modelling as human-in-the-loop assistants, but expert validation remains necessary to ensure syntactic correctness, semantic consistency, and domain validity.

2. Related Work and Research Gap

2.1. LLMs for Model-Based Engineering

Large Language Models are increasingly investigated as assistants for engineering tasks that require the transformation of natural-language information into structured representations. In Model-Based Systems Engineering, this capability is particularly relevant because many early system descriptions, stakeholder needs, and requirements are initially expressed in textual form, whereas later development activities require formalized artefacts such as requirements models, structural decompositions, interfaces, and behavioural descriptions. Prior work has shown that LLMs can support selected MBSE activities, including requirements analysis, model querying, model manipulation, and preliminary SysML-v2 model generation [7–9]. However, the use of LLMs in model-based engineering differs from general text generation. A generated artefact must not only be linguistically plausible, but also syntactically valid, semantically consistent, and compatible with the modelling language and tool environment. This is especially relevant for SysML-v2, whose textual notation makes it more accessible to LLM-based generation than purely graphical modelling languages, but also imposes formal constraints on element definitions, relationships, and behavioural constructs [3]. Consequently, LLMs should be understood as human-in-the-loop modelling assistants rather than autonomous modelling systems. Despite their potential, LLMs remain subject to well-known reliability limitations. They may generate plausible but incorrect content, omit relevant elements, introduce unsupported relations, or produce outputs that are syntactically close to the target language but not executable in a modelling tool [10]. These limitations are critical in MBSE, where model artefacts are expected to support traceability, verification, and decision-making. Therefore, LLM-assisted SysML-v2 generation requires structured prompting, explicit validation, and task-specific evaluation.

2.2. Prompt Engineering for SysML-v2 Artefact Generation

Prompt engineering provides a mechanism for controlling LLM behaviour by specifying the task, context, constraints, examples, reasoning steps, and expected output format. In the context of SysML-v2 artefact generation, prompt design is particularly important because the output must combine domain content with formal modelling syntax. General prompts may be sufficient for exploratory text generation, but formal engineering artefacts require more constrained prompting to reduce ambiguity and improve reproducibility. The prompt strategies considered in this study represent different levels of

guidance. Zero-Shot prompting serves as a baseline in which the model receives only the task description. Few-Shot prompting adds examples to guide the structure and style of the output. Chain-of-Thought prompting encourages stepwise reasoning before generating the final artefact. Chain-of-Verification extends this idea by asking the model to review and correct its own output. Retrieval-Augmented Generation adds domain-specific context to the prompt and is therefore expected to improve completeness and terminology alignment [5,11]. For SysML-v2 generation, these strategies address different modelling challenges. Requirements generation benefits from contextual completeness and precise formulation. Structural modelling requires correct decomposition, relation types, and hierarchy. Behavioural modelling requires consistent states, transitions, triggers, and conditions. Therefore, the effectiveness of a prompt strategy is expected to depend on the target artefact type rather than being universally transferable across all MBSE tasks.

2.3. Systematic Literature Search

To position the proposed method within the current research field, a systematic literature search was conducted. The search focused on studies addressing the intersection of LLMs, artificial intelligence, MBSE, SysML, SysML-v2, and methodical model generation. The databases included Scopus, ScienceDirect, IEEE Xplore, ACM Digital Library, and INCOSE. The search strategy combined terms related to Large Language Models, artificial intelligence, Model-Based Systems Engineering, SysML, SysML-v2, methods, frameworks, and processes. The initial search returned 1207 records. After merging the results, 157 duplicates were removed, leaving 1050 unique studies for screening. Title screening excluded 779 studies, and abstract screening excluded a further 221 studies. This resulted in 50 publications for full-text assessment. After applying the inclusion and exclusion criteria, 44 publications were excluded due to missing access, language restrictions, or insufficient relevance to the research questions. Backward and forward snowballing identified two additional relevant studies. In total, eight primary studies were selected for detailed analysis. The selected studies confirm that LLMs can support MBSE-related tasks, but they also show that the field is still at an early stage. Existing work frequently addresses individual use cases, such as natural-language interaction with models, requirements processing, or first model skeleton generation. However, only limited evidence is available on how different LLMs and prompt strategies perform under comparable conditions. Furthermore, SysML-v2 support, syntax validation, and reproducible evaluation procedures are not treated consistently. The results of the PRISMA process are shown in Figure 1 and Table 1.

Table 1. Primary studies selected through the systematic literature search.

No.	Title	Author(s)	Year	Source
S1	Accelerating Model-Based Systems Engineering by Harnessing Generative AI	Crabb, Erin Smith; Jones, Matthew T.	2024	[12]
S2	An Agent-Based Approach for the Automatic Generation of Valid SysML v2 Models in Industrial Contexts	Cibrián et al.	2025	[13]
S3	Easing Adoption of Model-Based System Engineering With Application of Generative AI	Patel et al.	2024	[14]
S4	Generating SysML Behavior Models via Large Language Models: An Empirical Study	Wang et al.	2025	[15]
S5	Leveraging Large Language Models for Direct Interaction with SysML v2	John K. DeHart	2024	[16]

Table 1. Cont.

No.	Title	Author(s)	Year	Source
S6	Enhancing Model-Based Systems Engineering with Large Language Models	Rafique et al.	2025	[7]
S7	LLM-Assisted Semantic Alignment and Integration in Collaborative Model-Based Systems Engineering Using SysML v2	Li et al.	2025	[17]
S8	SysML v2 Based Modelling Guidelines for Mechanical System Elements	Boelsen et al.	2025	[18]

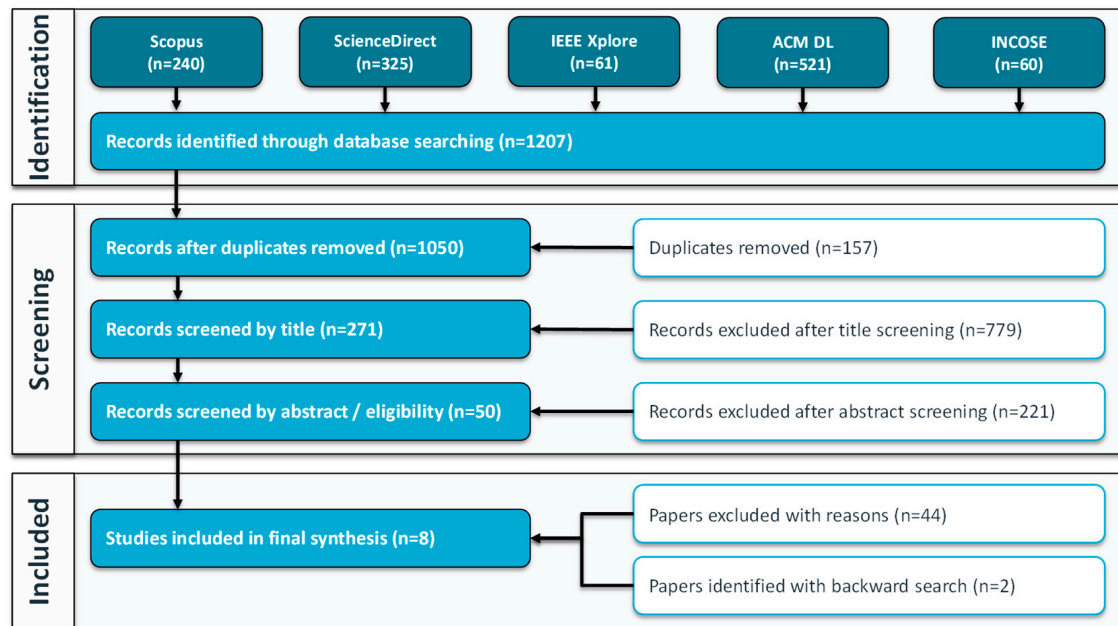


Figure 1. PRISMA-based selection process from 1207 database records to eight primary studies included in the final literature synthesis.

2.4. Harvey-Balls-Based Assessment of Existing Approaches

The selected studies were assessed using a Harvey-Balls-based evaluation scheme shown in Figure 2. The criteria covered methodological clarity, understandability, application accuracy, LLM relevance, MBSE relevance, SysML-v2 relevance, prompt-engineering support, tool/API integration, and reusability. This evaluation was used to identify not only whether existing approaches address LLM-assisted MBSE, but also how complete and transferable they are for formal SysML-v2 artefact generation. The assessment shows that most studies provide a clear motivation and understandable methodological descriptions. Several works demonstrate that LLMs can support the transformation of natural-language engineering descriptions into model-related structures or assist in interacting with existing MBSE models [7–9]. Nevertheless, the level of formal modelling support varies considerably. In particular, SysML-v2 is not consistently addressed, and prompt engineering is often treated as an implementation detail rather than as an experimental variable. Tool integration and reproducibility also remain limited, especially where generated outputs are not validated against formal syntax or evaluated against explicit reference artefacts. These findings indicate that existing research provides promising building blocks but does not yet offer a systematic framework for comparing LLMs, prompt strategies, and SysML-v2 artefact types. This gap motivates the method and experimental design introduced in the following sections.

Publication	Method	Clarity	Application accuracy	Large Language Model	Model-Based Systems Engineering	SysML v2	Prompt Engineering	Tool/API	Reusability
	K1	K2	K3	K4	K5	K6	K7	K8	K9
S1									
S2									
S3									
S4									
S5									
S6									
S7									
S8									
Legend: = completely fulfilled = well fulfilled = moderately fulfilled = poorly fulfilled = not fulfilled									

Figure 2. Harvey-Balls comparison of eight primary studies across nine methodological, LLM, MBSE, SysML-v2, integration, and reusability criteria.

2.5. Derived Research Gap

The literature analysis and Harvey-Balls assessment reveal four central gaps. First, existing studies rarely compare multiple LLMs under identical modelling conditions. Second, prompt strategies are not systematically evaluated across different SysML-v2 artefact types. Third, generated model artefacts are often assessed qualitatively, while quantitative and semantic evaluation are rarely combined. Fourth, reproducible workflows that include input preparation, generation, syntax validation, and task-specific quality assessment are still missing. This paper addresses these gaps by proposing a prompt-strategy-driven method for LLM-assisted SysML-v2 artefact generation. The method is evaluated across requirements generation, block definition modelling, and state-machine modelling, thereby covering textual, structural, and behavioural MBSE artefacts. By comparing four LLMs and five prompt strategies using syntax validation, F1-score analysis, and LLM-as-a-Judge evaluation, the study provides a structured basis for identifying which prompt-model configurations are suitable for different SysML-v2 modelling tasks.

3. Methods

3.1. Methodological Design and Objectives

The proposed method was developed to support the controlled generation of SysML-v2 artefacts from natural-language engineering information. Its purpose is not

to replace model-based systems engineers, but to provide a structured human-in-the-loop workflow in which Large Language Models (LLMs) assist the early creation of formal model artefacts. The method addresses the research gap identified in Section 2 by enabling a reproducible comparison of different prompt strategies, LLMs, and SysML-v2 artefact types under comparable conditions. The method is based on four design objectives. First, the generation process must be artefact-specific, because requirements, structural models, and behavioural models impose different syntactic and semantic constraints. Second, the method must support prompt comparability, allowing different prompt strategies to be evaluated using the same task input. Third, it must support model comparability, so that differences in the generated artefacts can be attributed to the selected LLMs and not to varying input conditions. Fourth, the method must include validation and evaluation, since SysML-v2 artefacts are only useful in MBSE if they are syntactically processable, semantically meaningful, and traceable to the intended engineering content. These objectives lead to a workflow that separates input preparation, prompt strategy selection, LLM selection, artefact generation, and quality evaluation. This separation is necessary because the quality of the final artefact depends not only on the model, but also on the prompt formulation, the available context, the target artefact type, and the validation procedure.

3.2. Five-Phase Workflow

The method consists of five consecutive phases, as shown in Figure 3. The phases define a controlled transition from informal engineering descriptions to evaluated SysML-v2 artefacts. Each phase produces an explicit intermediate result, which improves transparency and supports reproducibility. The same LLM is used during one iteration of the whole workflow. The concrete inputs, outputs, recorded information, and decision rules for each phase are summarized in Appendix A, Table A2.

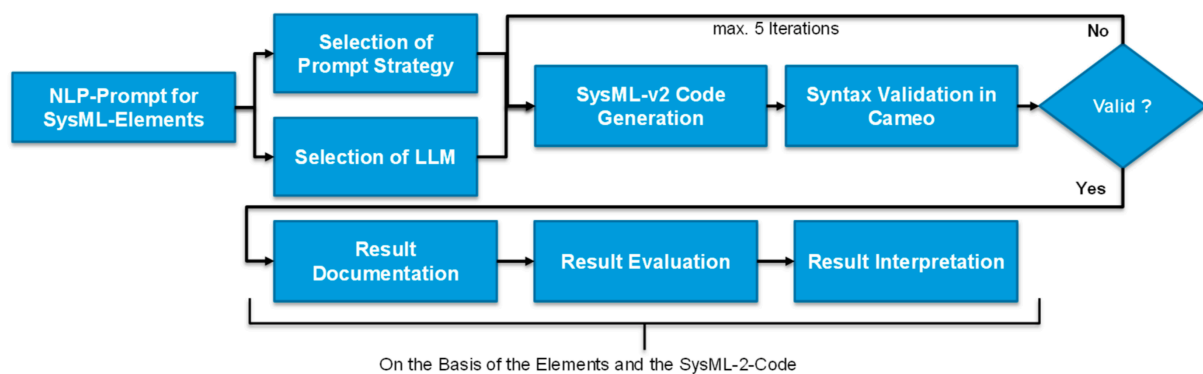


Figure 3. Five-phase workflow covering input preparation, prompt and model selection, SysML-v2 generation, syntax correction, evaluation, and interpretation.

In Phase 1, the modelling task and the input information are defined. This includes the selection of the System of Interest, the relevant engineering context, and the target SysML-v2 artefact type. In this study, the artefact types are requirements, block definition models, and state-machine models. The input is formulated in natural language and contains the system information that should be transformed into formal model elements. In Phase 2, the prompt strategy is selected. The method considers five prompt strategies: Zero-Shot, Few-Shot, Chain-of-Thought, Chain-of-Verification, and Retrieval-Augmented Generation. These strategies differ in the amount and type of guidance provided to the LLM. The purpose of this phase is to define how the model is instructed, what contextual information is provided, and which output format is required. In Phase 3, the LLM is selected and configured. The selected models are used under comparable conditions and receive equivalent task inputs. This ensures that differences in output quality can

be analysed with respect to both prompt strategy and model behaviour. The method distinguishes between model selection and artefact generation so that the experimental setup remains transparent and reproducible. In Phase 4, the SysML-v2 artefact is generated. The natural-language input, selected prompt strategy, and selected LLM are combined to produce SysML-v2 code. The generated output is then checked for syntactic validity and compatibility with the modelling environment. If syntax errors occur, correction iterations are performed while preserving the intended model content. Each model–prompt–task configuration was generated once. Where syntax errors occurred, up to five correction iterations were performed using the validation feedback; these iterations were part of the syntax-repair procedure and did not constitute independent repeated experimental runs. This phase also includes systematic documentation of prompts, model outputs, correction steps, and final artefacts. In Phase 5, the generated artefacts are evaluated and interpreted. The evaluation depends on the artefact type. Requirements and structural elements can be compared against reference artefacts designed for this use case on the basis of the same data the LLMs are given, using precision, recall, and F1-score. Behavioural models, such as state-machine models, may allow multiple valid solutions and therefore require semantic assessment in addition to element-level comparison. For this reason, the method combines quantitative metrics with LLM-as-a-Judge evaluation. The judge model, evaluation procedure, scoring scale, number of judging runs, and relationship to the evaluated generation models are documented in Appendix A, Table A4.

Artificial intelligence was used in this study as the object of investigation, as a controlled modelling assistant for generating candidate SysML-v2 artefacts from natural-language engineering input, and as an evaluation instrument. The four experimental generation models were ChatGPT-5.4 Pro, Gemini 3.1 Pro, Llama 4 Maverick, and DeepSeek-v3. Claude Opus 4.6 was used separately for the LLM-as-a-Judge evaluation and was not one of the models compared for artefact generation. ChatGPT-5.5 was used only for linguistic revision, formulation support, and translation of the manuscript. All LLM-generated outputs were documented, syntactically validated, and evaluated against reference artefacts or semantic assessment criteria, while expert review remained necessary to ensure modelling correctness, consistency, and domain validity.

3.3. Prompt Strategies

The selected prompt strategies represent different levels of model guidance. Zero-Shot prompting is used as a baseline and provides only the task description and expected output format. This strategy tests whether an LLM can generate a SysML-v2 artefact without examples or additional reasoning support. Few-Shot prompting adds representative examples to the prompt. The examples guide the model regarding syntax, structure, and level of detail. This strategy is particularly relevant when the expected output follows a recurring pattern, such as a requirement list or a block hierarchy. Chain-of-Thought prompting encourages the model to reason stepwise before generating the final artefact. In SysML-v2 generation, this can help the model identify relevant system elements, relationships, states, or transitions before formalizing them. This strategy is expected to be particularly useful for behavioural modelling, where the correctness of the result depends on sequential and conditional logic. For the Chain-of-Thought configuration, the model was instructed to perform a structured analysis before producing the final artefact. Only the final generated artefact and any explicitly requested concise intermediate output were stored and evaluated; hidden provider-internal reasoning was neither accessible nor assessed. Chain-of-Verification prompting extends the generation process by requiring the model to review and correct its own output. The aim is to reduce omissions, inconsistencies, and unsupported relations before the artefact is evaluated. This strategy is relevant for structural modelling, where

over-modelling and missing relations can both reduce model quality. Retrieval-Augmented Generation adds external or task-specific context to the prompt. This context can include system descriptions, domain information, or modelling constraints. The purpose is to improve completeness and terminology alignment, especially for requirements generation, where missing implicit information can reduce recall.

3.4. Artefact Generation and Syntax Validation

SysML-v2 artefact generation is treated as a controlled transformation task. The LLM receives the same modelling input across comparable experimental conditions, while only the prompt strategy and model are varied. The output is expected to follow the textual SysML-v2 notation and to contain the model elements required by the target task. The generated artefact is first assessed syntactically. For this purpose, the generated SysML-v2 code was imported into Cameo (Magic Systems of Systems Architect/Cameo Enterprise Architecture 2026x). An artefact was considered syntactically valid when the import was completed successfully and no syntax errors preventing parsing were reported by the tool. A valid output must be processable in the modelling environment and must not contain syntax errors that prevent import or parsing. If the generated code is invalid, the error information is used for iterative correction. The correction process is restricted to syntactic and structural repair; the intended semantic content of the generated artefact is not manually improved during this phase. This distinction is important because the evaluation should reflect the LLM-generated modelling quality rather than expert reconstruction of the artefact. In addition to syntax validation, the method documents each generated artefact together with the corresponding LLM, prompt strategy, input task, and correction history. This documentation enables later comparison of prompt–model configurations and supports reproducibility.

3.5. Evaluation Logic

The evaluation logic combines formal correctness and semantic quality. For artefact types with an explicit ground truth, the generated elements are compared with reference elements using precision, recall, and F1-score. Precision indicates how many generated elements are relevant, recall indicates how many reference elements were recovered, and the F1-score provides a balanced measure of both. This is suitable for requirements generation and selected structural model elements. However, not all SysML-v2 artefacts can be evaluated adequately through exact element matching. Structural and behavioural models may differ in naming, decomposition depth, or modelling style while still being semantically valid. Therefore, the method also includes LLM-as-a-Judge evaluation to assess criteria such as completeness, consistency, syntactic plausibility, semantic correctness, and suitability for the intended MBSE task. This is particularly relevant for block definition and state-machine modelling. The combined evaluation approach allows the method to capture both measurable element-level correctness and broader model quality. As a result, the method provides a basis for analysing how prompt strategy and model choice influence SysML-v2 artefact generation across different modelling tasks.

4. Experimental Setup

4.1. Case Study System

The proposed method was evaluated using a traction battery system for battery electric vehicles as the case study. This system was selected because it represents a safety-relevant, interdisciplinary engineering system that combines electrical, thermal, mechanical, and control-related functions. It therefore provides a suitable application context for evaluating LLM-assisted SysML-v2 artefact generation across requirements, structural,

and behavioural modelling tasks. The traction battery is treated as the System of Interest. Its relevant engineering context includes energy storage, high-voltage operation, thermal regulation, battery management, housing, electrical interfaces, and operational safety. The case study is not intended to evaluate battery design itself, but to provide a realistic and sufficiently complex engineering scenario for testing whether LLMs can transform natural-language system information into formal SysML-v2 artefacts. This allows the experiment to examine not only textual generation quality, but also whether the generated artefacts contain meaningful system elements, consistent relationships, and behaviourally plausible model structures.

4.2. SysML-v2 Generation Tasks

The experimental setup contains three modelling tasks. These tasks were selected to represent the three central artefact categories required in early-stage MBSE: requirements, structure, and behaviour.

4.2.1. Task 1 Requirements Generation

The first task evaluates whether LLMs can derive structured SysML-v2 requirements from natural-language descriptions of the traction battery system. The generated artefacts are expected to include relevant functional and non-functional requirements and to express them in a form suitable for model-based processing. This task provides the semantic basis for subsequent modelling activities because requirements define what the system must achieve and constrain later architectural and behavioural descriptions. The output of this task is compared against a reference set of validated requirements. The evaluation therefore focuses on whether the LLM identifies the relevant requirement content, avoids unsupported additions, and produces requirements that are sufficiently clear and complete for further MBSE use.

4.2.2. Task 2 Block Definition Modelling

The second task evaluates structural model generation. The LLMs are instructed to identify the relevant system, subsystem, and component elements of the traction battery and to represent them as a SysML-v2 block definition model. The expected output includes a hierarchical decomposition of the traction battery and relevant structural relationships between its elements. This task is more complex than requirements generation because the LLM must not only extract individual elements, but also organize them into a coherent system architecture. Errors may occur through missing components, incorrect hierarchy, over-modelling, or unsupported relations. For this reason, the evaluation considers both element-level correctness and semantic modelling quality.

4.2.3. Task 3 State-Machine Modelling

The third task evaluates behavioural model generation. The LLMs are asked to derive a SysML-v2 state-machine model for selected battery-related behaviour. The generated artefact is expected to include relevant states, transitions, triggers, and conditions. This task is used to examine whether LLMs can move beyond static decomposition and generate behaviourally consistent model structures. State-machine modelling is the most interpretive task in the experiment because there may be several valid ways to represent the same behaviour. Therefore, exact comparison against one reference solution is less appropriate than for requirements or structural elements. The evaluation focuses on behavioural plausibility, consistency of transitions, completeness of states, and suitability for the intended MBSE context.

4.3. LLMs and Prompt-Strategy Matrix

Four LLMs were selected to cover both closed-source and open-source model categories. The closed-source models were ChatGPT-5.4 Pro by OpenAI and Gemini 3.1 Pro by Google. The open-source models were Llama 4 Maverick by Meta AI and DeepSeek-v3 by DeepSeek. This selection was used to compare proprietary models, which typically provide strong language understanding and robust generation, with open models that may offer advantages regarding transparency, local deployment, and data-control-sensitive MBSE applications. The study uses these four models specifically to represent both commercial and open LLM landscapes under comparable conditions. Detailed information on the displayed model identifiers, access methods, access dates, and the specific role of each model is provided in Appendix A, Table A1.

Each model was combined with five core prompt strategies: Zero-Shot, Few-Shot, Chain-of-Thought, Chain-of-Verification, and Retrieval-Augmented Generation. This resulted in 20 core model–prompt configurations for each modelling task. For requirements generation, an additional platform-native RAG Upload condition was evaluated for ChatGPT and Gemini, for which direct document-upload functionality was available. Consequently, the requirements task comprised 22 configurations, while the block-definition and state-machine tasks each comprised 20 configurations, resulting in 62 configurations overall. The RAG Upload condition was not fully crossed across all four models and relied on platform-specific document-ingestion and context-processing capabilities. It therefore represents a platform–model–prompt condition and does not isolate the effect of retrieval augmentation from platform-specific document handling. All models otherwise received the same task inputs and corresponding core prompt-strategy variants.

4.4. Generation Procedure

Each experimental run follows the same procedure. First, the task-specific natural-language input is prepared. Second, the corresponding prompt strategy is applied to the input. Third, the prompt is submitted to the selected LLM. Fourth, the generated SysML-v2 artefact is documented together with the model, prompt strategy, task, and configuration. Fifth, the generated artefact is checked for syntactic validity and modelling-tool compatibility. If syntax errors occur, correction iterations are performed. Corrections were LLM-driven rather than manually reconstructed. The validation message and the previous artefact were returned to the same model with a correction prompt, after which the revised artefact was validated again. Human intervention was limited to transferring the artefact and validation feedback between the model and the modelling tool. These iterations are limited to repairing syntax or structural incompatibilities and are not used to manually improve the semantic content of the generated model. This distinction is important because the experiment aims to evaluate the LLM-generated artefact quality rather than the quality of expert-corrected models. The final syntactically valid artefact, or the last generated artefact if validity cannot be achieved, is then passed to the evaluation stage.

4.5. Evaluation Metrics

The evaluation combines quantitative and qualitative assessment. For Task 1 and selected aspects of Task 2, a ground-truth-based evaluation is used. Generated elements are compared with reference elements and classified as true positives, false positives, or false negatives. Precision measures the share of generated elements that are relevant, recall measures the share of reference elements recovered by the model, and the F1-score provides a balanced measure of precision and recall. For Task 2 and Task 3, LLM-as-a-Judge evaluation is additionally used because structural and behavioural models can be valid even when they differ from a single reference solution. The judge evaluation considers criteria

such as completeness, structural consistency, syntactic plausibility, semantic correctness, and suitability for the modelling task. This is especially important for state-machine models, where several state decompositions and transition structures may be acceptable. The combined evaluation design allows the experiment to compare prompt strategies and LLMs across different artefact types. It also prevents the assessment from relying exclusively on exact element matching, which would be insufficient for complex SysML-v2 structures. The LLM-as-a-Judge scores are used as rubric-based comparative indicators among the evaluated artefacts. They do not independently establish semantic correctness, engineering validity, or suitability for industrial use, which require assessment by qualified systems-engineering experts.

4.6. Experimental Scope and Reproducibility

The experimental scope is limited to early-stage SysML-v2 artefact generation. The study does not claim to generate a complete industrial MBSE model or to replace expert modelling. Instead, it evaluates how different LLMs and prompt strategies perform when generating first-draft SysML-v2 artefacts from natural-language engineering information. To support reproducibility, each run is defined by the same four parameters: modelling task, input description, prompt strategy, and LLM. The generated outputs, correction attempts, and evaluation results are documented for later comparison. Accordingly, the reported results represent one documented generation sequence per configuration rather than repeated sampling of each model's output distribution. Consequently, the reported differences are descriptive and do not establish statistical superiority or general optimality of a prompt strategy. Based on this experimental design, Section 5 presents and discusses the results for the three modelling tasks in the same sequence.

5. Results

5.1. Overview of the Evaluation

The results are presented according to the three SysML-v2 generation tasks introduced in Section 4. The evaluation focuses on the effect of prompt strategy and model choice on artefact quality. Requirements generation was evaluated primarily using F1-score analysis against a reference set. The task-specific comparison units and the operational definitions of true positives, false positives, and false negatives are provided in Appendix A, Table A3. Block definition modelling was assessed using both F1-score and LLM-as-a-Judge evaluation, since structurally valid models may differ in naming and level of decomposition. State-machine modelling was evaluated using LLM-as-a-Judge assessment because multiple valid behavioural representations are possible for the same system behaviour.

Across the documented generation sequences, the observed artefact scores differed across prompt strategies and modelling tasks. Within the investigated case study, context-enriched prompting achieved the highest observed scores for requirements generation, verification-oriented prompting achieved the highest combined observed performance for structural modelling, and reasoning-oriented prompting achieved the highest average judge score for behavioural modelling.

5.2. Requirements Generation

Requirements generation shows the clearest improvement with increasing prompt structure and contextual support. Zero-Shot and Few-Shot prompting provide moderate baseline results, while Chain-of-Thought and Chain-of-Verification improve the quality of generated requirements by guiding the models toward more complete and consistent outputs. The highest observed scores in the requirements task were obtained when Retrieval-Augmented Generation with document upload was used. In this platform-specific condi-

tion, ChatGPT and Gemini each achieved an overall F1-score of 0.80, representing the highest observed values in this task. This result reflects the complete platform–model–prompt configuration and should not be attributed to retrieval augmentation alone, because document ingestion and context processing were platform-specific. Within the investigated case study, the result is nevertheless consistent with task-specific documentation supporting the identification and formulation of relevant requirements. Figure 4 shows the observed scores for the individual model–prompt configurations. Because RAG Upload was unavailable for DeepSeek and Llama, the figure does not support a general comparison between proprietary and open-model categories for this condition. ChatGPT improves from 0.47 in the Zero-Shot setting to 0.80 with RAG Upload, while Gemini improves from 0.45 to 0.80. DeepSeek also benefits from structured prompting, particularly from CoT and CoVe, but does not reach the level of the RAG Upload configurations because direct document upload was not available. Llama shows weaker and more variable performance, indicating a stronger dependency on prompt formulation.

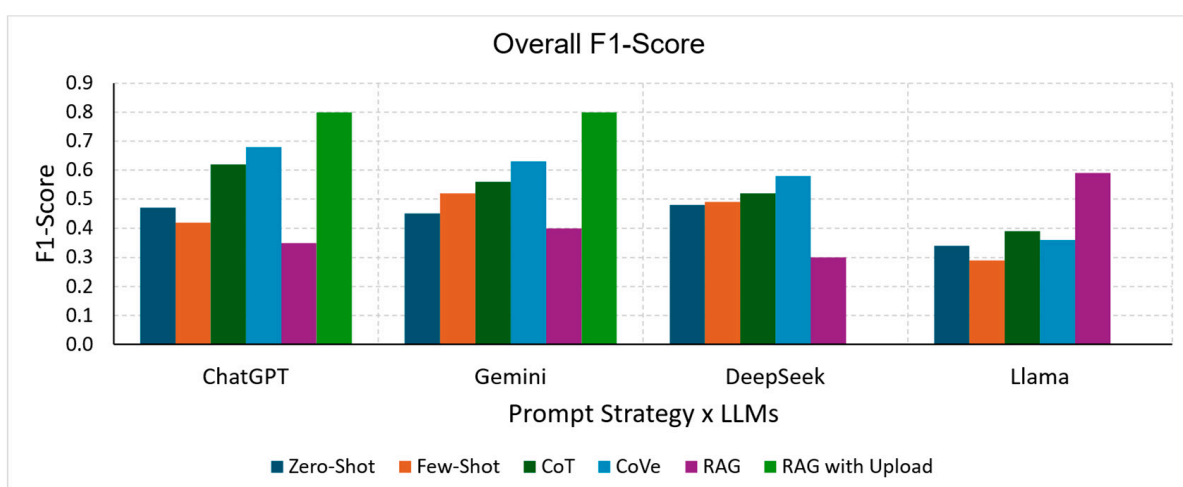


Figure 4. Overall F1-scores for requirements generation across LLMs and prompt strategies (DeepSeek and Llama did not provide RAG-Upload functionality).

The aggregated results show that RAG Upload achieved the highest observed average score in the requirements task, followed by CoVe and CoT. In contrast, the core RAG condition without direct document upload produced lower observed scores. Because the RAG Upload condition was available only through specific platforms, the difference reflects both access to task-specific documentation and platform-specific document ingestion and context processing. It should therefore not be interpreted as the isolated effect of retrieval augmentation. The corresponding aggregated requirement-generation results by prompt strategy and by LLM are provided in Appendix A, Figure A1.

5.3. Block Definition Modelling

The block definition modelling task reveals a more complex performance pattern than requirements generation. F1-score analysis alone suggests that Gemini and Llama produce relatively stable structural outputs, while ChatGPT and DeepSeek show stronger improvements when more structured prompt strategies are used. However, the F1-score is sensitive to naming differences, modelling granularity, and the level of decomposition. As a result, models that generate semantically rich but differently structured BDDs may be penalized despite producing useful artefacts. The aggregated BDD F1-score results by prompt strategy and by LLM are provided in Appendix A, Figure A2. The

detailed BDD F1-score comparison across all LLM-prompt configurations is shown in Appendix A, Figure A3.

For this reason, the F1-score evaluation was complemented by LLM-as-a-Judge assessment. The criterion-level ratings were summed and divided by the maximum attainable total score. The resulting normalized score ranges from 0 to 1, with higher values indicating more favourable ratings under the predefined LLM-as-a-Judge rubric. Under the predefined judge rubric, ChatGPT received some of the highest ratings for completeness and semantic detail, even though its F1-score was low in several configurations due to over-modelling and terminology mismatch. CoVe achieved the highest combined observed performance for this task in the investigated case study, considering both the F1-based and judge-based assessments. This observed pattern is consistent with the interpretation that explicit self-verification can help reduce missing components and support structural completeness.

Figure 5 shows that CoT and CoVe received the highest average normalized judge scores, while RAG received lower scores in the BDD task. These scores are comparative ratings under the predefined judge rubric and do not independently establish engineering validity. Within this rubric, the observed pattern is consistent with structural modelling benefiting from prompts that explicitly organize components and relations rather than only provide domain information. The result also shows that exact element matching is insufficient as the sole evaluation method for structural SysML-v2 artefacts. A valid BDD may differ from the reference solution while still representing a coherent and useful system structure. The corresponding aggregated normalized LLM-as-a-Judge results for the BDD task are summarized in Appendix A, Figure A4.

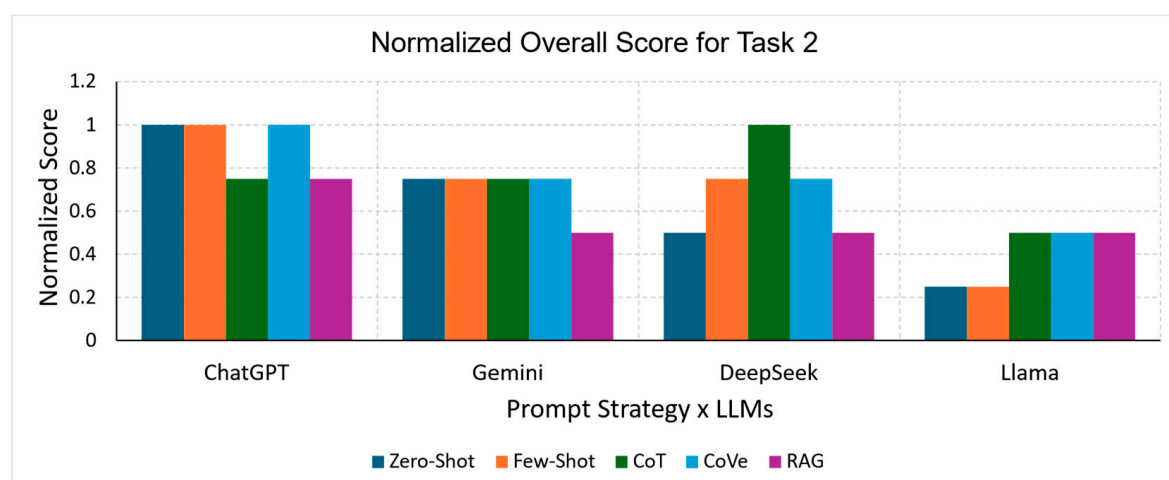


Figure 5. Normalized LLM-as-a-Judge scores for block definition modelling across LLMs and prompt strategies.

5.4. State-Machine Modelling

State-machine modelling is the most behaviourally demanding task in the experiment. The generated artefacts must contain plausible states, transitions, triggers, and conditions. Unlike requirements and structural decomposition, there is no single correct state-machine representation. Therefore, the evaluation focuses on semantic plausibility, behavioural consistency, and suitability for the intended SysML-v2 modelling context. More specifically, behavioural plausibility was assessed based on state completeness, logical transition structure, consistency of triggers and conditions, and alignment with the traction-battery operating description.

Within the investigated case study, Chain-of-Thought prompting achieved the highest average normalized judge score for this task. CoT reaches the highest average normalized prompt-strategy score of 0.88 and is the only strategy for which all four models achieve at least 0.75. This observed pattern is consistent with the interpretation that explicit stepwise reasoning can support behavioural modelling by helping the model identify relevant states and transition logic before producing formal SysML-v2 output.

Figure 6 shows model-dependent differences in the normalized LLM-as-a-Judge scores. ChatGPT received the highest average normalized judge score of 0.95 and the maximum rubric score in four of the five prompt strategies. DeepSeek received an average judge score of 0.85, Gemini received 0.80, and Llama received 0.55. These scores provide comparative ratings of the documented artefacts under the selected judge rubric. They should not be interpreted as definitive measures of engineering validity or as independently validated estimates of model capability. The aggregated state-machine LLM-as-a-Judge scores by prompt strategy and by LLM are summarized in Appendix A, Figure A5.

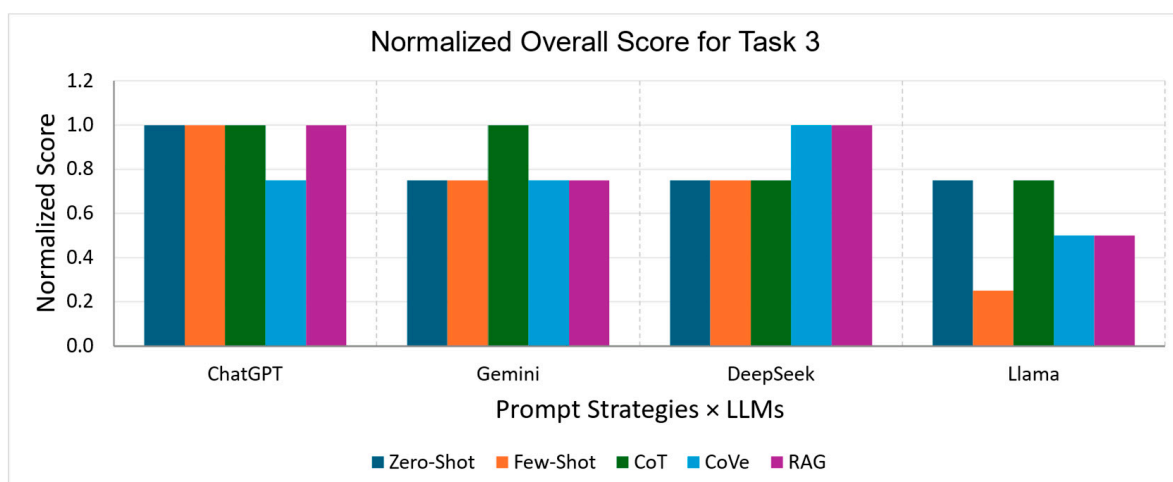


Figure 6. LLM-as-a-Judge scores for state-machine modelling across LLMs and prompt strategies.

5.5. Cross-Task Comparison

Within the investigated case study, no prompt strategy achieved the highest observed score across all modelling tasks. RAG with document upload achieved the highest observed scores for requirements generation, CoVe achieved the highest combined observed performance for block definition modelling, and CoT achieved the highest average judge score for state-machine modelling. Because each configuration was generated once, these findings describe the investigated generation sequences and should not be interpreted as general rankings of prompt strategies.

The model comparison also reveals task-dependent differences. Under the predefined LLM-as-a-Judge rubric, ChatGPT received comparatively high ratings in the BDD and STM tasks. Gemini received scores in the upper observed range across several investigated configurations, particularly for requirements generation and state-machine modelling. DeepSeek received higher observed scores under several structured prompt conditions with strategies that provide explicit reasoning or verification steps. The judge-based comparisons are rubric-dependent indicators and do not constitute definitive assessments of engineering validity.

Overall, the results support three conclusions. First, LLM-assisted SysML-v2 generation is feasible for early-stage modelling, but the generated artefacts require validation before they can be used in engineering workflows. Second, prompt-strategy selection was an important experimental factor associated with observed differences in artefact scores

within the investigated case study. Third, evaluation must be adapted to the artefact type: F1-score is useful for requirements and element-level comparisons, whereas structural and behavioural models require additional semantic assessment.

6. Discussion

The results show that LLM-assisted SysML-v2 artefact generation is feasible, but strongly dependent on the interaction between artefact type, prompt strategy, and model selection. Within the investigated case study, the highest observed requirements-generation scores occurred in the platform-specific RAG Upload condition. This pattern is consistent with task-specific documentation supporting requirement identification and formulation; however, the condition also depended on platform-specific document ingestion and context processing and therefore does not isolate the effect of retrieval augmentation. However, context alone is not sufficient if the model cannot reliably structure the retrieved information. The RAG condition also reflects the retrieval and document-processing capabilities available through each evaluated platform. Consequently, differences in this condition may partly result from platform-specific context handling rather than from the prompt strategy alone. For block definition modelling, the results show that structural SysML-v2 artefacts require more than identifying correct components. The generated model must also contain meaningful hierarchies, relationships, and decomposition levels. Therefore, F1-score alone is insufficient for evaluating BDD quality, since semantically useful models may differ from the reference in naming or granularity. The complementary LLM-as-a-Judge evaluation provides a broader assessment of structural quality. However, these scores were not independently calibrated against assessments by multiple MBSE experts and should therefore be interpreted as structured comparative indicators rather than definitive measures of engineering validity. The observed CoVe results are consistent with verification-oriented prompting supporting the identification of missing elements and inconsistent structures in this case study. State-machine modelling is the most complex task because it requires behavioural reasoning about states, transitions, triggers, and conditions. The strong performance of CoT suggests that stepwise reasoning is especially useful when the target artefact depends on logical sequence and dynamic system behaviour. The model comparison further shows that behavioural modelling is sensitive to general reasoning capability and prompt formulation. Overall, prompt engineering should not be treated as a secondary implementation detail. It is a central methodological factor that directly affects artefact quality. Within the investigated case study, no single prompt strategy achieved the highest observed score across all tasks. The observed results therefore indicate that prompt selection should be aligned with the intended SysML-v2 artefact type, while broader conclusions require repeated generations and additional case studies. The findings also confirm that LLMs should be used as human-in-the-loop modelling assistants rather than autonomous model generators. They can support early modelling and reduce initial effort, but expert validation remains necessary to ensure syntactic correctness, semantic consistency, and domain validity. The study is limited by its focus on one engineering case study and a defined set of LLMs and prompt strategies. In addition, F1-score evaluation depends on the selected reference artefacts, while LLM-as-a-Judge assessment may introduce evaluator bias. Prompt wording, model version changes, and available context may also influence the results.

7. Conclusions

This paper proposed and evaluated a prompt-strategy-driven method for LLM-assisted SysML-v2 artefact generation in Model-Based Systems Engineering. The method structures the generation process into input preparation, prompt-strategy selection, LLM selection,

artefact generation, syntax validation, and quality evaluation. It was evaluated in a traction battery system case study across three representative early-stage MBSE tasks: requirements generation, block definition modelling, and state-machine modelling. The study therefore addresses textual, structural, and behavioural artefact generation within a controlled experimental setting. With respect to RQ1, the results indicate that prompt strategy selection has a substantial influence on the quality of generated SysML-v2 artefacts. Prompting should therefore not be treated as a secondary implementation detail, but as a central methodological factor in LLM-assisted MBSE. The observed effects were task-dependent: context-enriched prompting was most beneficial for requirements generation, verification-oriented prompting improved structural modelling, and reasoning-oriented prompting supported behavioural modelling. With respect to RQ2, the evaluated LLMs showed task-dependent performance differences. ChatGPT and Gemini achieved high observed scores in context-rich requirements generation, while ChatGPT received particularly high comparative judge ratings for the structural and behavioural artefacts. DeepSeek benefited from structured prompting, especially when reasoning or verification steps were included. Llama showed higher variability, indicating that prompt design is especially important for less stable model configurations. These findings suggest that model selection should be aligned with the intended artefact type and evaluation objective rather than based on general model capability alone.

With respect to RQ3, no single prompt–model configuration achieved the highest observed score across all investigated SysML-v2 artefact types. Within the traction-battery case study, RAG Upload achieved the highest observed scores for requirements generation, Chain-of-Verification achieved the highest combined observed performance for block definition modelling, and Chain-of-Thought achieved the highest average judge score for state-machine modelling. The RAG Upload result is platform-specific because this condition was available only for ChatGPT and Gemini and depended on the native document-upload and context-processing capabilities of their respective platforms. Because each configuration was generated once, these results should be interpreted as descriptive findings for the investigated case study rather than as generally optimal prompt–model configurations.

The main contribution of this study is a structured evaluation framework for comparing LLMs and prompt strategies in formal SysML-v2 artefact generation. The findings indicate that LLMs can support early-stage modelling by accelerating the creation of first-draft artefacts and by assisting engineers in transforming natural-language system information into formal modelling structures. At the same time, the results show that generated artefacts cannot be used without validation. Syntax checking, semantic assessment, and expert review remain necessary to ensure correctness, consistency, and domain validity. The conclusions of this study should be interpreted within the limits of the experimental scope. The evaluation is based on one engineering case study, a selected set of LLMs and prompt strategies, and reference artefacts defined for the investigated tasks. In addition, F1-score evaluation depends on the chosen ground truth, while LLM-as-a-Judge assessment may introduce evaluator bias and provides rubric-based comparative indicators rather than definitive measures of engineering validity. Nevertheless, the results show that prompt-strategy selection and model choice were relevant experimental factors within the investigated case study. LLMs should therefore be understood as human-in-the-loop modelling assistants rather than autonomous MBSE tools. Future work should extend the evaluation to additional engineering domains, further SysML-v2 artefact types, tool-integrated validation workflows, and expert-based assessment to strengthen generalizability and practical applicability.

8. Outlook

Future work should evaluate the proposed method in additional engineering domains to assess its generalizability beyond traction battery systems. Further SysML-v2 artefact types, such as internal block diagrams, parametric models, use cases, sequence models, and interface models, should also be included to determine whether the observed prompt-strategy effects remain stable across broader modelling tasks. Further research should investigate stronger integration with MBSE tools, automated syntax repair, domain-specific retrieval pipelines, and expert-based validation. A further extension could investigate scenario-based SysML-v2 generation, in which data-driven generation and clustering methods provide representative operational or environmental scenarios that are subsequently transformed into requirements and behavioural artefacts [19,20]. In particular, future studies should compare LLM-as-a-Judge results with assessments from experienced systems engineers to reduce evaluator bias. In the long term, LLM-assisted MBSE should evolve toward controlled, traceable, and tool-integrated workflows in which AI supports formal modelling without replacing expert responsibility.

Author Contributions: Conceptualization, A.S., U.V.K., B.K., S.B.-S. and T.V.; methodology, A.S., N.W., B.K. and A.K.T.; software, A.S. and A.K.T.; validation, U.V.K., B.K., S.B.-S., A.D., N.W. and T.V.; formal analysis, A.S., N.W., A.D. and A.K.T.; writing—original draft preparation, A.S. and A.K.T.; writing—review and editing, U.V.K., N.W., B.K., S.B.-S., A.D. and T.V.; visualization, A.S. and A.K.T.; supervision, B.K., S.B.-S. and T.V. All authors have read and agreed to the published version of the manuscript.

Funding: The publication of this paper was funded by the TU Braunschweig Publication Fund.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The dataset available on request from the authors.

Acknowledgments: The research presented in this paper was conducted within the framework of two projects: the “Regional Transformation Network Southeast Lower Saxony” (ReTraSON), funded by the Federal Ministry for Economic Affairs and Energy on the basis of a decision by the German Bundestag, and the project “AI-Based Knowledge Assistance for the Development of Vehicle Electrical/Electronic Architectures” (KIARA), funded by the Lower Saxony Ministry of Science and Culture (MWK) within the Center for Digital Innovations Lower Saxony (ZDIN).. Artificial intelligence was used in this study as the object of investigation and as a controlled modelling assistant for generating candidate SysML-v2 artefacts from natural-language engineering input, as well as formulating and translating the text for this publication. While all LLM-generated outputs were documented, syntactically validated, and evaluated against reference artefacts or semantic assessment criteria, expert review remains necessary to ensure modelling correctness, consistency, and domain validity. The study evaluated four Large Language Models: ChatGPT-5.4 Pro by OpenAI, Gemini 3.1 Pro by Google, Llama 4 Maverick by Meta AI, and DeepSeek-v3 by DeepSeek. ChatGPT-5.5 by OpenAI was used only for linguistic revision, formulation support, and translation of the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

Abbreviation	Definition
AI	Artificial Intelligence
API	Application Programming Interface
BDD	Block Definition Diagram
CoT	Chain-of-Thought

CoVe	Chain-of-Verification
F1-score	Harmonic mean of precision and recall
LLM	Large Language Model
MBSE	Model-Based Systems Engineering
NLP	Natural Language Processing
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
RAG	Retrieval-Augmented Generation
RQ	Research Question
STM	State-Machine Modelling
SysML-v2	Systems Modeling Language version 2

Appendix A

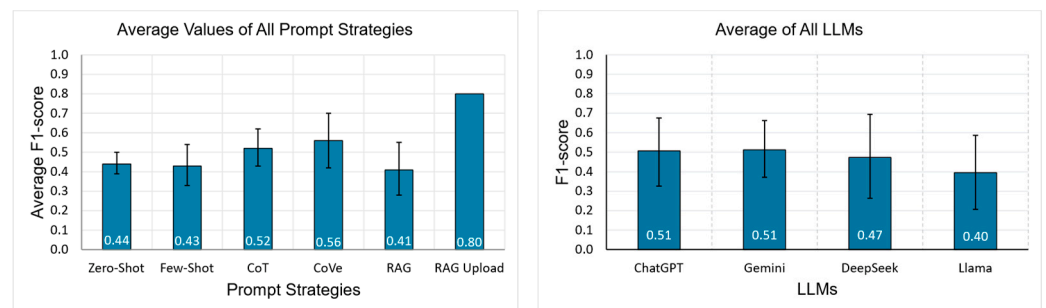


Figure A1. Supplementary results for requirements generation. (Left): Average F1-score by prompt strategy. (Right): Average F1-score by LLM.

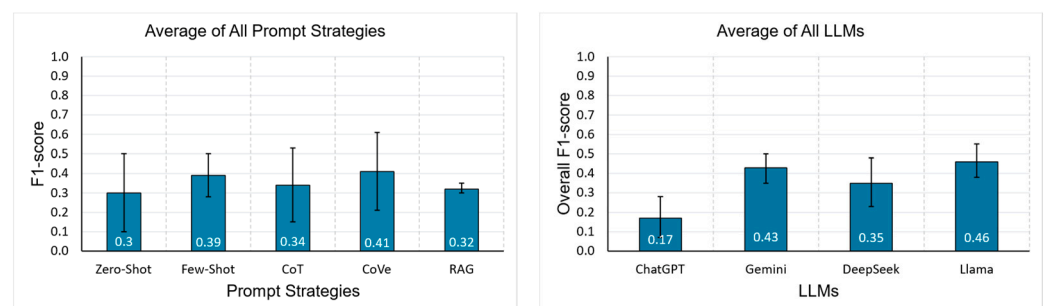


Figure A2. Supplementary F1-score results for block definition modelling. (Left): Average F1-score by prompt strategy. (Right): Average F1-score by LLM.

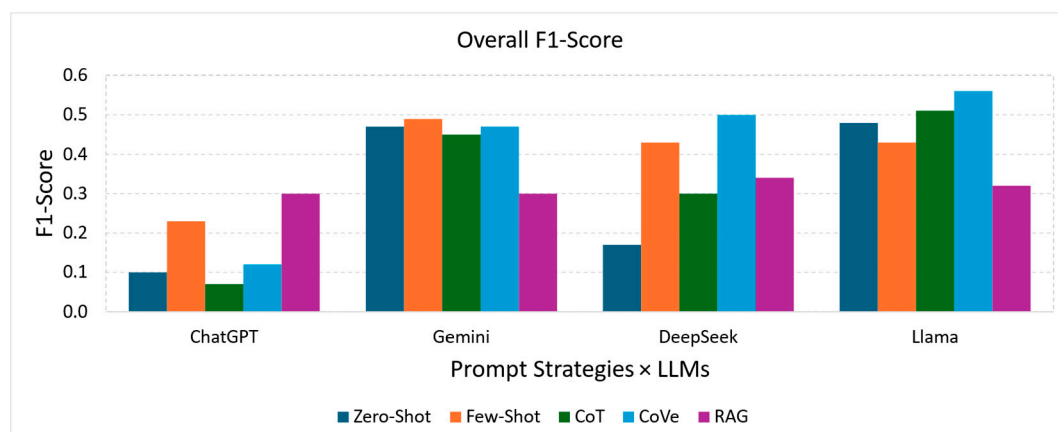


Figure A3. Overall F1-scores for block definition modelling across all LLM and prompt-strategy combinations.

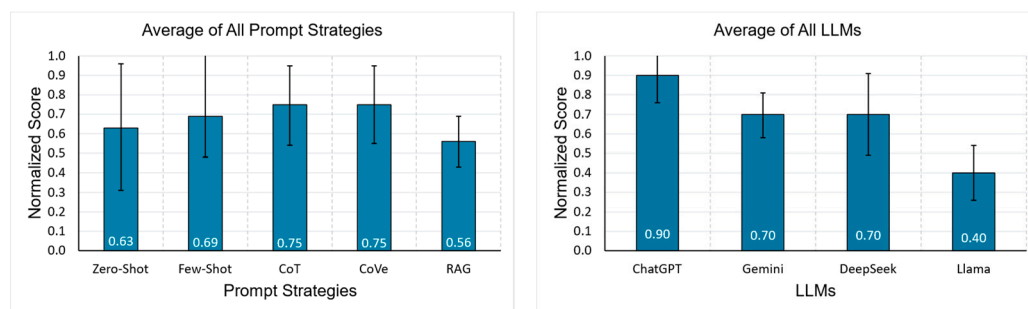


Figure A4. Supplementary normalized LLM-as-a-Judge results for block definition modelling. **(Left):** Average normalized score by prompt strategy. **(Right):** Average normalized score by LLM.

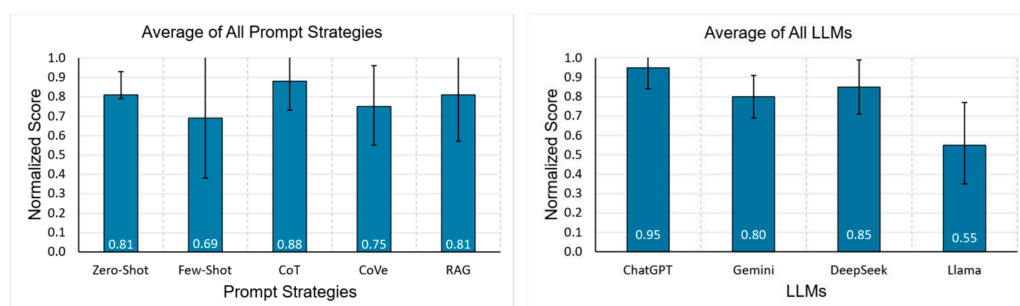


Figure A5. Supplementary LLM-as-a-Judge results for state-machine modelling. **(Left):** Average score by prompt strategy. **(Right):** Average score by LLM.

Table A1. LLM models and execution conditions.

Provider	LLM	Access Method	Last Access Date	Use in the Study
OpenAI	ChatGPT-5.4 Pro	Web Interface	29 April 2026	SysML-v2 requirements, block definition models, and state-machine models
Google	Gemini 3.1 Pro	Web Interface	29 April 2026	SysML-v2 requirements, block definition models, and state-machine models
Meta AI	Llama 4 Maverick	API	29 April 2026	SysML-v2 requirements, block definition models, and state-machine models
DeepSeek	DeepSeek-v3	API	29 April 2026	SysML-v2 requirements, block definition models, and state-machine models

Table A2. Detailed description of the five-phase workflow.

Phase	Input	Main Activity	Output	Decision Rule
Input preparation	System description and task definition	Select and structure relevant information	Task-specific input package	Input must contain sufficient domain and artefact information
Prompt strategy selection	Input package and modelling task	Select Zero-Shot, Few-Shot, CoT, CoVe, or RAG	Complete prompt instance	Strategy selected according to task and available context
LLM selection	Prompt instance	Assign one of the four evaluated models	Model-prompt configuration	Same task information used across comparable configurations

Table A2. *Cont.*

Phase	Input	Main Activity	Output	Decision Rule
Generation and validation	Prompt and selected LLM	Generate SysML-v2 output and validate syntax	Generated or corrected artefact	Maximum defined correction iterations; validation in modelling tool
Evaluation	Validated artefact and reference information	Calculate F1-score or judge-based score	Quantitative and qualitative assessment	Metric selected according to artefact type

Table A3. Operational definitions used for the ground-truth-based F1-score evaluation.

Task	Unit of Comparison	True Positive	False Positive	False Negative	Matching Rule
Requirements generation	One reference requirement	A generated requirement represents a requirement contained in the reference set with the same engineering meaning.	A generated requirement has no corresponding requirement in the reference set.	A requirement contained in the reference set is not represented in the generated output.	Minor wording differences and synonyms were accepted when the intended engineering meaning remained equivalent. Requirement identification and formulation were evaluated against the predefined reference requirements.
Block definition modelling	One reference model element or structural relationship	A generated block or structural relationship corresponds to an element or relationship contained in the reference BDD.	A generated block or relationship has no corresponding element in the reference BDD.	A block or relationship contained in the reference BDD is missing from the generated output.	Matching considered engineering meaning, structural role, and hierarchy. Minor naming differences were accepted when the same system element was represented. Differences in granularity or decomposition were assessed against the defined reference scope.

Table A4. Configuration of the LLM-as-a-Judge evaluation.

Aspect	Description
Judge provider	Anthropic
Judge model	Claude Opus 4.6
Access method	Web Interface
Evaluation approach	Single-answer grading, in which each generated artefact was assessed individually rather than through pairwise comparison
Evaluated tasks	Block definition modelling and state-machine modelling
Evaluation criteria	Correctness, relevance, coherence, completeness, structural consistency, and task-specific functional plausibility
BDD-specific criteria	Completeness and meaningful organization of blocks, properties, relationships, and hierarchy
State-machine-specific criteria	Completeness of states and substates, transition structure, triggers, logical consistency, and functional correctness
Scoring scale	Criterion-level ratings on a five-point scale, aggregated and normalized to a score between 0 and 1
Number of judging runs	One documented judging run per generated artefact
Model overlap	Claude Opus 4.6 was not one of the four LLMs evaluated for SysML-v2 artefact generation

References

- Schmidt, M.M.; Stark, R. Model-Based Systems Engineering (MBSE) as Computer-Supported Approach for Cooperative Systems Development. In *Proceedings of the 18th European Conference on Computer-Supported Cooperative Work (ECSCW 2020): Exploratory Papers*; European Society for Socially Embedded Technologies (EUSSET): Siegen, Germany, 2020.
- Sparx Systems. *Guide to MBSE with SysML, User Guide Series*; Sparx Systems: Victoria, Australia, 2023.
- Bajaj, M.; Friedenthal, S.; Seidewitz, E. Systems Modeling Language (SysML v2) Support for Digital Engineering. *Insight* **2022**, *25*, 19–24. [\[CrossRef\]](#)
- Naveed, H.; Khan, A.U.; Qiu, S.; Saqib, M.; Anwar, S.; Usman, M.; Mian, A. A Comprehensive Overview of Large Language Models. 2023. Available online: <https://arxiv.org/pdf/2307.06435> (accessed on 20 July 2026).
- Vogelsang, A.; Fischbach, J. Using Large Language Models for Natural Language Processing Tasks in Requirements Engineering: A Systematic Guideline. 2024. Available online: <https://arxiv.org/pdf/2402.13823> (accessed on 20 July 2026).
- Stein, A.; Mirzai, A.; Axmann, J.; Vietor, T. Integrating the capabilities offered by large language models into the requirements engineering process. *Digit. Eng.* **2026**, *10*, 100098. [\[CrossRef\]](#)
- Rafique, K.A.; Shah, S.; Dalecke, Š.; Grimm, G. Enhancing Model-Based Systems Engineering with Large Language Models. In *INCOSE International Symposium*; Wiley: Tokyo, Japan, 2025; pp. 1523–1543.
- Chami, M.; Abdoun, N.; Bruel, J.-M. Artificial Intelligence Capabilities for Effective Model-Based Systems Engineering: A Vision Paper. In *INCOSE International Symposium*; Wiley: Tokyo, Japan, 2022; pp. 1160–1174.
- Hanke, F.; Bitá, I.M.; von Heißen, O.; Julian, W.; Aschot, H.; Roman, D. AI-Augmented Systems Engineering: Conceptual Application of Retrieval-Augmented Generation for Model-Based Systems Engineering Graph. In *Proceedings of the Design Society*; Cambridge University Press: Cambridge, UK, 2025; pp. 439–448.
- Bender, E.M.; Gebru, T.; McMillan-Major, A.; Shmitchell, S. On the Dangers of Stochastic Parrots. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*; ACM: New York, NY, USA, 2021.
- Sahoo, P.; Singh, A.K.; Saha, S.; Jain, V.; Mondal, S.; Chadha, A. A Systematic Survey of Prompt Engineering in Large Language Models: Techniques and Applications. 2024. Available online: <https://arxiv.org/pdf/2402.07927> (accessed on 20 July 2026).
- Crabb, E.S.; Jones, M.T. Accelerating Model-Based Systems Engineering by Harnessing Generative AI. In *Proceedings of the 2024 19th Annual System of Systems Engineering Conference (SoSE)*; IEEE: Piscataway, NJ, USA, 2024; pp. 110–115.
- Cibrián, E.; Olivert-Iserte, J.; Llorens, J.; Álvarez-Rodríguez, J.M. An Agent-Based Approach for the Automatic Generation of Valid SysMLv2 Models in Industrial Contexts. *Comput. Ind.* **2025**, *172*, 104350. [\[CrossRef\]](#)
- Patel, A.; Maheshwaran, Y.; Santhya, P. Easing Adoption of Model Based System Engineering with Application of Generative AI. In *Proceedings of the 2024 IEEE Space, Aerospace and Defence Conference (SPACE)*; IEEE: Bangalore, India, 2024; pp. 871–874.
- Wang, Y.; Ge, N.; Liu, J.; Cao, Z.; Chen, Z.; Hu, C. Generating SysML Behavior Models via Large Language Models: An Empirical Study. In *Proceedings of the 16th International Conference on Internetware*; Association for Computing Machinery: New York, NY, USA, 2025; pp. 366–377.
- DeHart, J.K. Leveraging Large Language Models for Direct Interaction with SysML v2. *INCOSE Int. Symp.* **2024**, *34*, 2168–2185. [\[CrossRef\]](#)
- Li, Z.; Husung, S.; Wang, H. LLM-Assisted Semantic Alignment and Integration in Collaborative Model-Based Systems Engineering Using SysML v2. *arXiv* **2025**, arXiv:2508.16181.
- Boelsen, K.; May, M.; Jacobs, G.; Mennicken, M.; Moers, F.; Zerwas, T.; Höpfner, G. SysML v2 Based Modelling Guidelines for Mechanical System Elements. *Forsch. Ingenieurwes.* **2025**, *89*, 60. [\[CrossRef\]](#)
- Hu, J.; Cao, Y.; Tan, G. A dynamic spatiotemporal graph generative adversarial network for scenario generation of renewable energy with nonlinear dependence. *Energy* **2025**, *335*, 138049. [\[CrossRef\]](#)
- Hu, J.; Liu, Y.; Chai, Y.; Tan, G. An adaptive dynamic scenario clustering method for power system optimization control under spatiotemporal correlated uncertainties. *Meas. Sci. Technol.* **2026**, *37*, 236206. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.