Transforming Systems Engineering in Nuclear Projects with Generative AI: A Path to Efficiency and Compliance

Jérémy Bourdon[a], Julien Rodriguez, Quentin Lesigne, Pauline Suchet, Berenger Fister, Loic Montagne, Olivier Malhomme, Lies Benmiloud-Bechet and Robert Plana[b]
Assystem Energy and Operation Services (AEOS), Tour Egée, Courbevoie 92400, France

Keywords: Systems-Engineering, LLM, SysML, Requirements Extraction, Requirements Allocation, Engineering Automation.

Abstract: This article explores the integration of generative artificial intelligence (AI) into nuclear systems engineering to improve efficiency and compliance. The Generative Systems Engineering (GenSE) project is transforming traditional systems engineering processes by leveraging AI across the entire plant lifecycle. Key challenges addressed include the extraction and reformulation of requirements, their allocation within the Product Breakdown Structure (PBS), and integration with existing engineering tools. To meet these challenges, a specialized Large Language Model (LLM) tailored for Nuclear Engineering, named "CurieLM", has been developed through fine-tuning. A workflow has been developed, using CurieLM, to automate requirements extraction, ensure quality assurance according to INCOSE guidelines, and facilitate allocation while maintaining compliance with ISO 15288 and ISO 24641 and integrating with SysML tools. The case study on a MOX fuel fabrication plant shows significant time reductions: 88% in requirements extraction, 87% in reformulation, and 66% in allocation to PBS. These improvements are accompanied by a gain in quality, based on feedback from requirements engineers. However, human verification remains essential to interpret and validate the results. In conclusion, the article highlights the potential of AI to transform systems engineering, while highlighting the need for collaboration between humans and AI to guarantee the quality of decisions.

1 INTRODUCTION

Over the past two years, Energy Transition policies have undergone significant shifts, with nuclear energy emerging as a cornerstone for achieving CO2 reduction targets. This shift has led to the initiation of numerous nuclear programs, bringing new challenges related to the timely and cost-effective delivery of these projects. Similar issues were encountered in the aerospace and aeronautics sectors years ago and were effectively addressed through the adoption of Systems Engineering.

However, in the nuclear sector, engineering practices remain predominantly document-centric, posing a significant barrier to improving project delivery timelines and overall performance. This underscores the urgent need for research aimed at automating, simplifying, and accelerating the adoption of system modeling approaches in nuclear infrastructure projects. The Generative Systems Engineering (GenSE) project aims to redefine systems engineering processes for installations throughout their lifecycle, by integrating artificial intelligence (AI) and more specifically foundation models. In addition, it would support operations, safety considerations and compliance with requirements.

However, integrating AI into nuclear systems engineering raises several challenges. Both technical and practical problematics need to be resolved to ensure effective implementation.

One concern is the accuracy and completeness of the probabilistic results generated by AI, particularly regarding needs and requirements analysis, functional and physical modeling, and system validation. It is essential to use carefully constructed data sets,

[a] https://orcid.org/0000-0001-6475-044X
[b] https://orcid.org/0000-0003-2378-8331

Bourdon, J., Rodriguez, J., Lesigne, Q., Suchet, P., Fister, B., Montagne, L., Malhomme, O., Benmiloud-Bechet, L. and Plana, R.
Transforming Systems Engineering in Nuclear Projects with Generative AI: A Path to Efficiency and Compliance.
DOI: 10.5220/0013443400003896

In *Proceedings of the 13th International Conference on Model-Based Software and Systems Engineering (MODELSWARD 2025)*, pages 395-406
ISBN: 978-989-758-729-0; ISSN: 2184-4348

include deterministic post-treatments and always keep human in the loop to minimize the risk of errors or omissions. In a field as critical as nuclear energy, even small inaccuracies can have significant repercussions on Nuclear Safety throughout the lifecycle.

Another obstacle lies in the integration of AI tools within existing engineering tools. These tools need to fit into established workflows and platforms, to encourage their adoption. They also need to demonstrate their ability to enhance process efficiency without introducing unnecessary complexity. The main objective is to enhance engineers' capabilities while preserving the continuity of existing practices.

User confidence is an essential element. For these tools to be adopted, engineers must perceive them as useful and adapted to their needs. This means involving them during the tool development phases, so that their feedback integrates the proposed solutions. The tools must also bring concrete benefits to operational tasks, reinforcing their status as useful tools rather than mere innovation fads.

By addressing these challenges, the GenSE project aims to modernize nuclear systems engineering practices. It relies on AI to improve decision-making, optimize processes and increase the efficiency and reliability of engineering tasks. At the same time, it offers solutions tailored to the specific needs of the nuclear sector.

2 BACKGROUND

2.1 Context

The GenSE initiative is taking place against a backdrop of increasing demands to improve the safety and regulatory compliance of critical infrastructures, particularly in the nuclear industry. The increasing complexity of the design and management of these facilities stems from stringent legal frameworks and the imperative of safe and efficient operation. These technical and regulatory challenges are further exacerbated by economic constraints to optimize costs, streamline processes and maintain rigorous quality and safety standards.

Economic considerations in the nuclear sector are various. One of the main challenges is the constant need to reduce costs, not only during the construction of new facilities, but also during the operation and eventual decommissioning of existing ones. At the same time, it is essential to accelerate the pace of design processes, as the regulatory constraints and complexity of nuclear projects often lead to long and costly development cycles. In addition, maintaining the reliability and safety of nuclear systems remains a mandatory requirement. The Nuclear Safety requirements of such facilities requires robust engineering practices and traceability to ensure safe and reliable operations.

The GenSE aims to tackle these challenges by developing AI-driven tools to improve systems engineering methodologies. By integrating artificial intelligence into engineering workflows, the project seeks to optimize processes, reduce costs, improve quality and the global efficiency of project execution. This approach also contributes the design phases to be completed on schedule as safety and compliance standards are met. Thanks to these innovations, the GenSE initiative intends to provide the nuclear industry with a framework that enables it to overcome its technical and economic hurdles, while preserving the integrity and reliability of its critical infrastructure.

2.2 State of the Art

This section presents a review of the literature on Text to model approaches to SysML generation, focusing on the nuclear domain. It addresses key issues concerning recent advances, methodologies and challenges in this field.

The use of Optical Character Recognition (OCR) and Natural Language Processing (NLP) for automatic requirements extraction from numeric and printed text documents is an active area of research. (Ben Nasr, 2016; Luttmer et al., 2023; Patel et al., 2024; van Remmen et al., 2023). Language models, pre-trained on large datasets, can be refined for specific technical domains such as nuclear engineering. Rule-based, machine learning and deep learning approaches are used to identify and classify requirements (Akundi et al., 2024; Luttmer et al., 2023; Zhao et al., 2022). The integration of domain-specific knowledge, such as ontologies and nuclear glossaries, can improve the accuracy of requirements extraction (Ben Nasr, 2016; Cocci et al., 2024) For example, CurieLM, a refined language model for the nuclear domain, illustrates the integration of specific knowledge to improve understanding of nuclear texts (Bouhoun et al., 2024).

Despite progress, the automatic generation of SysML models from text still faces a number of limitations (Ahmad et al., 2022; van Remmen et al., 2023). The ambiguity of natural language, the complexity of nuclear systems and the lack of structured training data are major obstacles (Chami et al., 2019; Luttmer et al., 2023; Necula et al., 2024). Interface management, requirements allocation and change impact analysis are particularly difficult to automate (Ahmad et al., 2022; Cocci et al., 2024; McDermott et al., 2020).

Artificial intelligence (AI) and machine learning (ML) offer opportunities to improve the accuracy and completeness of automatically generated models (Cocci et al., 2024; McDermott et al., 2020; Zhang & Yang, 2024). Techniques such as reinforcement learning, transfer learning and deep neural networks can be used to optimize model structure, semantics and consistency (Akundi et al., 2024; Winkler & Vogelsang, 2017). The integration of automatic model validation and human feedback can further refine the generation process. This extract about the Large Language Model (LLM) "The LLM meticulously examines these architectures and evaluates their alignment with the defined requirements. By correlating the intricate details within the physical and functional architectures, the LLM verifies if they meet or deviate from the established requirements." (Cocci et al., 2024) describes how a language model can be used to check the conformity of system architectures with requirements.

The integration of domain-specific knowledge is essential for the generation of accurate and relevant SysML models for nuclear engineering. (Ben Nasr, 2016; Bouhoun et al., 2024; Zhang & Yang, 2024). Nuclear ontologies, technical lexicons and component databases can provide valuable context for modeling algorithms (Alaoui et al., 2023). Knowledge extraction techniques from existing documents and models can also be used.

Compliance verification is crucial to ensure that nuclear systems comply with safety standards and regulations. The integration of compliance rules, constraints and standards into the text-model system enables real-time verification during the modeling process (Cocci et al., 2024). Formal logic and model checking techniques can be used to automate the verification process (Ben Nasr, 2016; Luttmer et al., 2023).

Managing interfaces and allocating requirements in complex systems is a major challenge for automation (Ahmad et al., 2022; McDermott et al., 2020). The complexity of interactions between components, the evolution of requirements and the lack of clear traceability make it difficult to capture these relationships automatically (van Remmen et al., 2023). Approaches based on knowledge graphs, semantic analysis and machine learning are being explored to tackle these challenges (Petnga, 2019).

Change impact analysis is essential for assessing the consequences of design or requirement modifications on the overall system (Mengist et al., 2021). Integrating this analysis into text-model systems makes it possible to track dependencies between model elements and predict the potential impact of changes (Weston et al., 2009). Constraint propagation, dependency analysis and simulation techniques can be used to automate this process (Plehn, 2018).

Automatic SysML (OMG, 2006) and diagrams from documentation is an important objective for improving communication and understanding of systems (Patel et al., 2024)Model transformation, natural language generation and graphical visualization techniques are used to produce clear, concise and informative documents (Akundi et al., 2024)

Evaluating the quality and accuracy of automatically generated models is essential to guarantee their reliability and usefulness (Chapurlat, 2013). Measures such as precision, recall, requirements coverage and semantic consistency can be used to quantify model quality. Techniques such as expert validation, comparison with reference models and simulation testing can complement these measures (Nastov et al., 2015).

The integration of text-model systems for SysML generation in the nuclear domain promises to improve the efficiency, accuracy and traceability of the system design process. Although significant progress has been made, challenges remain in terms of natural language ambiguity, managing system complexity and integrating domain-specific knowledge. Future research should focus on developing more robust NLP techniques, more intelligent AI models and more rigorous evaluation methods to overcome these limitations and realize the full potential of text-model systems for SysML generation in the nuclear domain.

3 METHODOLOGY

The GenSE concept is structured around eleven themes. These themes cover current activities in systems engineering and focus on preliminary design. These themes are derived below :

- **Automated Requirements Extraction:** Automating the extraction of requirements from documented requirements specifications.
- **Requirements Quality:** Reformulate requirements to comply with best practices such as INCOSE.

- **Functional Architectures:** Automate the creation of alternative functional architectures in response to requirements.
- **Work Breakdown Structure:** Automate the creation of a work breakdown structure for a design project.
- **Product Breakdown Structure and Architectures:** Automate the creation of logical architecture alternatives consistent with a functional architecture, and physical architecture alternatives consistent with a logical architecture.
- **Requirement Allocation:** Automate the allocation of requirements to different engineering artifacts.
- **Project Interfaces:** Automate the identification and documentation of interfaces in WBS elements, consistent with the organic architecture.
- **Continuous Verification of Compliance:** Automate the verification of compliance with the requirements of the system as designed, aiming for continuous verification.
- **Layout:** Generate installation layout alternatives with the various zoning zones common in the nuclear sector (e.g. radiological, fire, safety, etc.).
- **Deliverable Production:** Automate the creation of deliverables based on stakeholder expectations and all documentation developed (templates and documents).
- **Change Impact:** Automate the analysis and prediction of the impact on the design of a change (e.g. requirements, functionalities, configurations, etc).

In the remainder of this article, we will detail a workflow that integrates three of these themes: automated extraction, requirements quality and allocation. This workflow is intended to support the stakeholder and systems requirements definition process as described in ISO 15288 (ISO & IEC, 2023a) and ISO 24641 (ISO & IEC, 2023b).

4 ASSIST STAKEHOLDERS AND SYSTEM REQUIREMENTS DEFINITION PROCESSES

The workflow architecture is illustrated in Figure 1, which shows the input data in gray (i.e. a PBS-type specification and decomposition of the system of interest), the data produced by generative AI in purple, and the human interaction in blue. The tools used in this architecture include CurieLM-Mistral-7B-Instruct-v0.2, tools implementing the SysML language and INCOSE rules. In addition, a human-machine interaction (HMI) interface has been developed. The CurieLM-Mistral-7B-Instruct-v0.2 model, hereafter referred to as the CurieLM model, is a fine-tuned model based on the Mistral 7B Instruct v0.2 model, using instructions and data specific to the nuclear sector (Bouhoun et al., 2024).

For this workflow, we will apply the experiments to a case study of a Mixed Oxide Fuel (MOX) fabrication plant. This plant is designed to produce nuclear fuel from mixed fissile materials.

The input data to be collected must be sufficient for the system of interest, the system whose life cycle is under consideration, and must also be public. We based our research on open-access documents describing an existing Mox Fuel Fabrication Facility (MFFF). The choice of this system allows us to obtain technical data close to current design projects for a MOX fabrication facility in France. To describe this system, a list of documents was made available to the workflow:

- Design of the MOX fuel fabrication facility (Johnson & Brabazon, 1993)
- Application for authorization to construct the MOX fuel fabrication facility (DCS, 2006)

To consolidate the technical information retrieved, we relied on the following IAEA documents:

- AIEA, Status and advance in MOX technology (INTERNATIONAL ATOMIC ENERGY AGENCY, 2003)
- AEIA, SSG7 Safety of Uranium and Plutonium Mixed Oxide Fuel Fabrication Facilities (INTERNATIONAL ATOMIC ENERGY AGENCY, 2023a)
- AIEA, SSG6 Safety of Uranium fuel fabrication facilities (INTERNATIONAL ATOMIC ENERGY AGENCY, 2023b)

4.1.1 Step 1: Needs Extraction and Classification

The first Sub-step is to create a design specification document. Figure 2 illustrates the automation process used to produce design specifications for a MOX fuel fabrication plant.

This step is based on the use of an artificial intelligence model, in this case Mistral Large, to process technical documents and generate deliverables. In three successive activities, the system analyzes technical data from IAEA safety standards,

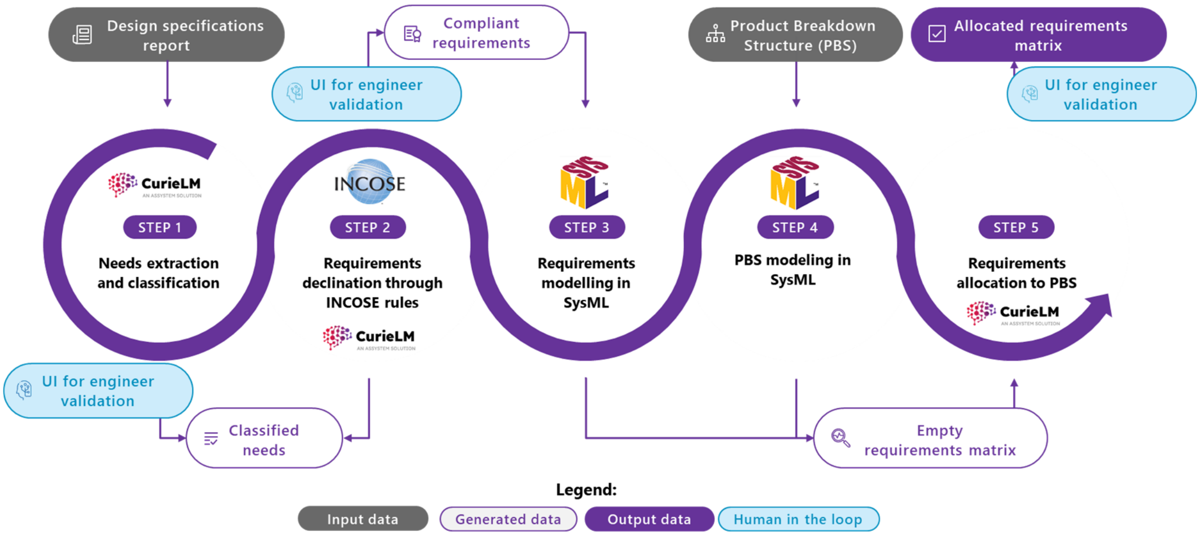

Figure 1: Workflow to assist stakeholder and system requirement process definition.

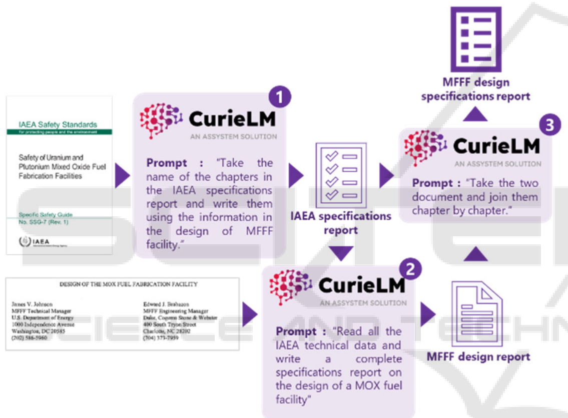

Figure 2: Automated process for creating specification reports for the MOX fuel fabrication plant.

writes a design report adapted to the case under study, then consolidates all the information into a finalized specifications report.

(1) The first is an analysis of Chapter 5 of the IAEA safety standards (Safety of Uranium and Plutonium Mixed Oxide Fuel Fabrication Facilities). The CurieLM is asked to read and interpret the technical data contained in this chapter. The result of this first step is a design specification report for a MOX plant.

(2) The second is the production of a design document based on the previously created element, as well as a technical document describing the design of a plant of the same type.

The CurieLM is once again being asked to create this document.

(3) The final activity is to combine the two previous documents, merging them chapter by chapter. Finally, the output document is a design requirements specification.

The second sub-step uses the document generated in the previous step to generate a list of requirements classified as functional or non-functional.

Figure 3 describes the automated workflow used to extract and classify requirements from a validated MOX Fuel Fabrication Facility (MFFF) design specification report. This step relies on the CurieLM artificial intelligence model to analyze the content of the report, organize the information and produce a ranked list of requirements.

The process begins by converting the validated report into usable textual content. This text is then broken down into distinct segments or "chunks" for easier processing. From these segments, CurieLM performs an initial classification, distinguishing relevant portions containing needs from those that do not. Once the needs have been identified, they are grouped together in an initial structured list.

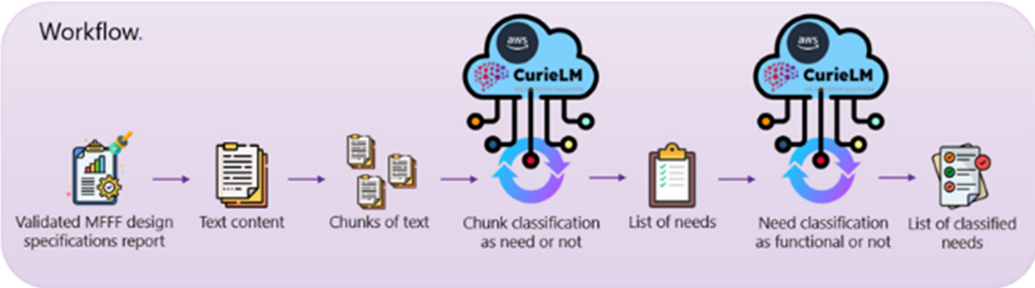

Figure 3: Automated workflow for extracting and classifying requirements.

In a second step, the extracted requirements are further classified to differentiate them into two categories: functional requirements, which define the system's expected capabilities, and non-functional requirements, which specify performance, safety or reliability constraints. The result is a complete, organized list of classified requirements, ready to be used as the basis for subsequent design and analysis stages.

4.1.2 Step 2: Requirements Declination Through INCOSE Rule

We worked with the CurieLM and used Langchain for output formatting and prompt engineering, as it gave good reasoning performance and was able to follow the desired output formats in previous tests on a similar use case.

This step requires interaction with the requirements engineer, so a dedicated interface has been created. An overview of this interface is provided in the Appendix.

The principle for reformulating the extracted requirements is as follows:

- Browse the file, line by line, and confirm or deny the proposal made by the tool.
- Look at the breakdown of each requirement.
- Check that the requirements have been written in accordance with INCOSE good writing practices.

The reformulation stage is carried out using the user interface developed as part of this project. The requirements from the previous step are provided as input data for this step. Two types of documents have been integrated into the interface:

- The list of requirements extracted by CurieLM, broken down by sentence.
- The list of requirements extracted by CurieLM, broken down by paragraph.

First, the requirements engineer checks the accuracy of the requirements. To do this, he accepts or rejects the requirement proposal in the HMI interface. This intermediate sub-step enables complete verification and validation of all stakeholder requirements extracted by the CurieLM from the specifications. More specifically, it is possible to:

- Validate or reject the CurieLM's choice of requirement identification.
- Summarize the title of the selected requirement.
- Select editing rules to be automatically checked by CurieLM.
- Check the rules and propose a new wording for the requirement.
- Accept the CurieLM proposal or accept the requirement with its original title.

Once the lines considered not to be requirements have been discarded, the reformulation stage continues, this time focusing on the quality of the formulation of each requirement.

To carry out this step, the HMI interface integrates INCOSE writing rules. For a selected requirement, the user chooses the rules to be applied and launches the verification. This verification is performed automatically by the CurieLM. The CurieLM indicates whether the requirement conforms to each chosen rule and suggests a conforming reformulation at the end of the check.

During this sub-step, it is possible to assess which rules the CurieLM interprets correctly and which it does not. Once the rules have been applied to the requirement, the requirement engineer chooses between validating the reformulation proposed by the CurieLM or retaining the original formulation. Once this choice has been made, the requirement is approved and added to the final list of requirements to be returned.

Initially, twenty INCOSE rules were implemented to assess formulation quality. The first requirements were checked against the twenty CurieLM rules. However, it soon became apparent that some rules were not correctly considered by the CurieLM. In fact, some rules add too much interpretation to the CurieLM, resulting in a reformulation that is too far from reality that is considered as hallucinations. For the rest of the experiment, these rules were discarded. In perspective, it would be interesting to prompt the CurieLM in such a way as to limit these counter-productive interpretations.

Here are some of the INCOSE rules that have been removed following this phase of experimentation:

- R2: Use the active voice in the main sentence structure of the need or requirement statement, with the responsible entity clearly identified as the subject of the sentence.
- R6: Use appropriate units when stating quantities; units of measurement for all numbers must be explicitly stated.
- R10: Avoid open-ended clauses such as “including but not limited to”, “etc” and “and so on”.
- R23: Avoid parentheses and brackets containing subordinate text.
- R24: Explicitly list sets instead of using a group name to name the set.
- R27: Avoid relying on headings to explain or understand the requirement.
- R30: Explicitly express the propositional nature of a condition for a single action, rather than giving lists of actions for a specific condition.
- R34: Use “each” instead of “all”, “any” or “both” when quantifying universally.

4.1.3 Step 3 & 4: SysML Modelling

Steps 3 and 4 consist in implementing the data in a tool used in MBSE and implementing the SysML language. They will not be detailed in the rest of this article. Stakeholder needs have been integrated as requirements objects and PBS elements as blocks. This implementation can be extended to other languages, such as that used in Capella (Roques, 2017) or other system modelling tools. An overview of data as integrated into a modelling tool is provided in the Appendix.

4.1.4 Step 5: Requirements Allocation to PBS

As for steps 1 and 2, we worked with the CurieLM. One call is made per requirement and per group of subsystems. For each group of subsystems, the CurieLM indicates which systems are affected by the requirement and returns an explanation.

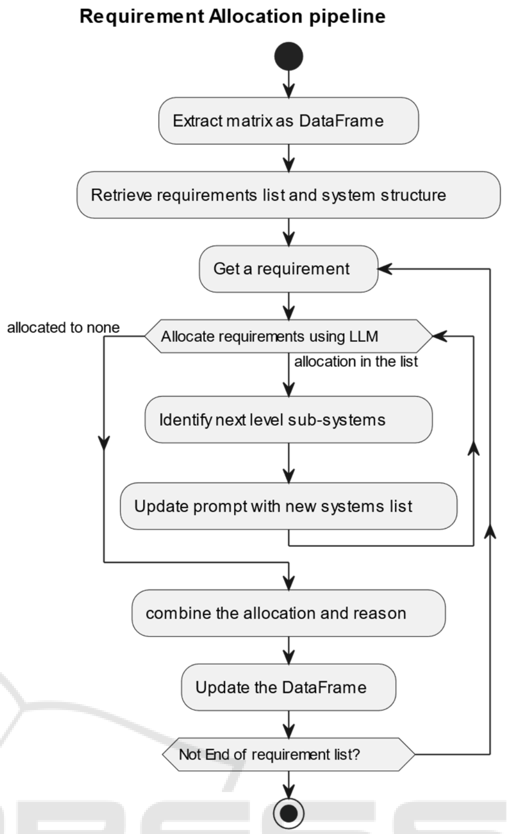

Figure 4: Description of the action to automatically allocate requirements to the PBS.

The tests were carried out at different temperatures (i.e. the LLM parameter that sets the “creativity” in the response from zero to one). Zero temperature (almost zero in our case, since Langchain does not allow zero temperature) seems to show better results and more consistent responses from one call to the next, which seems to generate more reliable outputs that will be easier and quicker for the requirement engineer to check before saving and sending them to the modelling tool.

The requirements allocation is presented in matrix form, where the requirements are the rows, and the PBS elements are the columns. Each positive element in the matrix indicates an allocation between the requirement and the corresponding PBS element. The sub-steps followed to achieve these allocations are illustrated in Figure 4.

From the elements in the model created in the system modeling tool, three elements are retrieved: the list of requirements, the PBS elements and their hierarchical structure (e.g. system X is made up of subsystems X.1, X.2 and X.3). Then, for each requirement, CurieLM allocates the requirement to the corresponding level 1 PBS elements. If these systems contain subsystems, the prompt is updated with the list of level 2 subsystems and CurieLM performs allocations to the second level of the PBS. This process is repeated down to level N, the lowest level in the PBS hierarchy. This process replicates what a systems engineer would do on the same task, i.e. proceed by iteration and recursion.

This step has also been integrated into the user interface described above. It enables the requirement engineer to select a list of requirements revised from the previous steps and a PBS, and to launch the allocation suggestion. He can then view the result in the form of a matrix in which CurieLM justifies his allocation suggestion.

When the engineer has finished reviewing CurieLM results, he can use the export button to export a file containing the list of allocations (in Excel format) and inject it into the system modelling tool of his choice.

In summary, the use of an LLM was able to assist and guide a requirements engineer through the process of defining stakeholder requirements. Starting from technical documentation describing a system of similar interest and the elements to be considered as described by a safety organization (a major stakeholder in a nuclear facility project), he was guided through the process to the creation of a requirements/system allocation matrix. The results and associated gains are presented in the following section.

5 RESULTS

5.1 Comparison Approach

A fully human and an intelligently assisted run were performed in parallel with the development of the CurieLM pipeline. We asked a system engineer to take the same input data (i.e. specifications and PBS) and replicate the steps by hand, without the help of the assistant. The goal is to compare the results and quantify the time savings brought by AI. The exercise given to the system engineer consists of the following steps:

Step 1 - Manual Requirements Extraction: During this initial phase, a 16-page document detailing the MOX Fuel Fabrication Facility Design Specification Report was provided for manual analysis. The task consisted of a thorough review of the document to identify and extract all discernible requirements. The extracted requirements were then classified according to their functional or non-functional nature. All duplicate requirements were identified and eliminated from the extracted requirements set.

Step 2 - Requirements Rewriting The second phase focused on rewriting the extracted requirements (excluding identified duplicates) to ensure adherence to the pre-selected INCOSE guidelines. These guidelines include a comprehensive set of rules designed to improve clarity, conciseness, completeness, and accurate quantification of the requirements. The rewritten requirements were then classified as either fully compliant with the selected INCOSE rules or considered to be already well-written in their original form.

The selected INCOSE rules were as follows:

- R1: Use the definite article "the" instead of the indefinite article "a".
- R7: Avoid using vague terms such as "some", "all", "allowable", "several", "many", "some", "almost always", "very near", "nearly", "about", "close to", "almost", and "approximate".
- R9: Avoid escape clauses such as "as far as possible", "as little as possible", "if possible", "if necessary", "to the extent necessary", "as appropriate", "as required", "as far as possible", and "if possible".
- R12: Use a separate clause for each condition or qualification.
- R26: Avoid using double-meaning pronouns and verbs: Avoid using indefinite pronouns and pronouns.
- R37: Explicitly define temporal dependencies: Explicitly define temporal dependencies instead of using temporal keywords such as "eventually", "until", "before", "after", "as", "once", "at the earliest", "at the latest", "instantaneous", "simultaneous", "finally".

Step 3,4 and 5 - Allocation of requirements in a modeling tool: the rewritten requirements were then loaded into the modeling tool. Each individual requirement was assigned to the most relevant elements of the PBS in the tool interface.

5.2 Operation Duration

This step showed a significant reduction in the time required, from 8.5 hours to 1 hour. Then, the extracted requirements were declined according to the INCOSE (International Council of Systems Engineering) rules, reducing the processing time from 16 hours to 2 hours thanks to automation. The requirements were then modeled in an engineering platform, integrating the identified needs and the declined requirements with a significant time saving. The allocation of requirements to the Product Breakdown Structure (PBS) was also optimized, a time saving from 7.5 hours to 2.5 hours. These results are summarized in Table 1.

Table 1: Duration comparison between manual and intelligent assisted operation.

	Conventional	CurieLM Assisted
Step 1	8,5 h	1 h
Step 2	16 h	2 h
Step 3-4-5	7,5 h	2,5 h
Total	31,5 h	5,5 h

5.3 Quality

The extraction file obtained with the extraction tool gives us 159 identified requirements. Most of the major requirements of the specifications were identified by it. After analysis and verification, it appears that the number of requirements is lower (around 73). This is potentially due to the way the specifications were broken down by the tool to extract the requirements. Some of the requirements selected by the tool are indeed too vague or imprecise, so it is wise to remove them from the list.

For example, here is an extracted requirement that should have been removed from the final list: "*223 - The program must also include provisions for record keeping and reporting to support continuous improvement*". The program is too vague here, it lacks a context that the specifications do not provide a priori. In addition, the relevant provisions are not explained, which does not allow us to understand how this requirement must be tested and verified.

Another requirement that was confirmed after analysis was "*103 - The MFFF shall use proven European technology and shall be adapted to comply with U.S. requirements*". This is still a very high level requirement, but it is consistent with most good drafting practices and is not easy to control and manage.

Unlike automated extraction, extraction with CurieLM gives much less output requirements. The splitting at the time of extraction was apparently not the same as for the extraction tool, hence a list of requirements different from the first list provided.

The identified requirements often have longer titles, with several sentences and even several topics. This last point makes the understanding of the requirement more difficult. For example: "*For cases where misidentification of containers could pose a hazard, provisions for easy identification of the content shall be used, such as unique colors, shapes, and valves. Additionally, technical provisions for inspection and maintenance of containers classified as items important to safety shall be available. The MFFF shall receive PuO2 from the Pit Disposition and Conversion Facility (PDCF) in containers that meet DOE standards. The PuO2 shall meet specification requirements of DCS and shall be shipped by methods that ensure the security of the material, under DOE authority until inside the MFFF secured area. Both the PDCF and the MFFF shall provide for sufficient storage of PuO2 to ensure a continuous flow to the MFFF to satisfy the demand curve for the material.*". This requirement is abstracted from CurieLM, but it contains different topics and too much information to be considered a single requirement.

Metrics for the comparison are synthetized in the Table 2.

Table 2: Comparison between automated operation and CurieLM assisted extraction results.

	Conventional	CurieLM Assisted
Extracted requirements	159	94
Approved requirements	73	56
Reworked Requirements	31	26
Discarded Requirements	86	38

5.4 Results Discussions

Time saving is the element that stands out the most from the comparison between the two approaches (i.e. engineer with or without CurieLM). When carried out, activities with CurieLM make it possible to carry out the same volume of work as that of the systems engineer in 2 or 3 times less time. Nevertheless, the work of the systems engineer remains very efficient, particularly in terms of decision-making and arbitration on the identification, allocation, or even formulation of needs. This efficiency is obviously closely related to the experience and expertise of the engineer. On the other hand, the precision and quality of the AI results are generally lower than those of the engineer. The automatic work of the AI is not linear and homogeneous, some tasks were more relevant to be carried out with the CurieLM while other tasks brought nothing (even from the point of view of time saving because the engineer had to rework the results of the LLM in all cases).

The first observation is that it is necessary to find a fair collaboration between the engineer and the AI to save time in this type of task without losing quality. The use of CurieLM allows engineers to save considerable time in the early design phases. In this context, the organization is often not yet fixed, and the multiplicity of roles and responsibilities can accumulate. Having a tool that reduces the time required to structure these roles, such as the construction of a technical specification, a PBS or verification activities.

The systems engineer provides results that are applicable to any project. Conversely, even if the AI results are consistent, they must necessarily be verified, or even reworked, to be usable in a project. The activity of the systems engineer therefore remains essential and his volume of activity must not be reduced compared to the LLM. These activities must include considering and collaborating with the LLM in the role of systems engineer.

6 PERSPECTIVES AND CONCLUSIONS

The first perspective concerns the improvement of workflows assisted by artificial intelligence, in order to achieve more precise and efficient extraction and classification of requirements. To achieve this, it will be necessary to develop new algorithms and integrate advanced machine learning techniques. In addition, the development of the user interface intended for engineering teams will play a key role. By integrating their feedback, the tool will be able to gradually evolve to adapt to the concrete needs of users. This approach will promote smooth adoption and optimized daily use.

The expansion of the automation workflow to other system engineering themes identified in the project represents an area of development. This will make it possible to integrate other key activities such as the automatic generation of architectures, the analysis of interfaces or the allocation of requirements to the subsystems concerned. By systematizing these approaches, the different stages of the project life cycle can be optimized.

The integration of text-model systems for SysML generation in the nuclear domain offers significant improvement prospects in terms of efficiency, accuracy and traceability of the system design process. Although significant progress has been made, challenges remain, particularly regarding the ambiguity of natural language, the complexity of system management and the integration of nuclear domain-specific knowledge.

REFERENCES

Ahmad, K., Abdelrazek, M., Arora, C., Bano, M., & Grundy, J. (2022). Requirements Engineering for Artificial Intelligence Systems: A Systematic Mapping Study. http://arxiv.org/abs/2212.10693

Akundi, A., Ontiveros, J., & Luna, S. (2024). Text-to-Model Transformation: Natural Language-Based Model Generation Framework. Systems, 12(9), 369. https://doi.org/10.3390/systems12090369

Alaoui, M. El, Chapurlat, V., Rabah, S., Richet, V., & Plana, R. (2023). An approach for ontology-based research and recommendation on systems engineering projects. Procedia Computer Science, 225, 1350–1359. https://doi.org/10.1016/j.procs.2023.10.123

Ben Nasr, S. (2016). Mining and Modeling Variability from Natural Language Documents: Two Case Studies. https://inria.hal.science/tel-01388392v1

Bouhoun, Z., Allali, A., Cocci, R., Assaad, M. A., Plancon, A., Godest, F., Kondratenko, K., Rodriguez, J., Vitillo, F., Malhomme, O., Bechet, L. B., & Plana, R. (2024). CurieLM: Enhancing Large Language Models for Nuclear Domain Applications. EPJ Web of Conferences, 302, 17006. https://doi.org/10.1051/epjconf/202430217006

Chami, M., Zoghbi, C., & Bruel, J.-M. (2019). A First Step towards AI for MBSE: Generating a Part of SysML Models from Text Using AI (pp. 123–136).

Chapurlat, V. (2013). UPSL-SE: A model verification framework for Systems Engineering. Computers in Industry, 64(5), 581–597. https://doi.org/10.1016/j.compind.2013.03.002

Cocci, R., Huang, S., Lesigne, Q., Suchet, P., Montagne, L., Roumili, E., Ali, A., Assaad, M. A., Vitillo, F., Benmiloud-Bechet, L., & Plana, R. (2024). Requirements Management Analytics Enabled by Model Based System Engineering and Artificial Intelligence for Decommissioning and Dismantling Projects.

DCS. (2006). License Application Package Overview MOX Fuel Fabrication Facility.

INTERNATIONAL ATOMIC ENERGY AGENCY. (2003). Status and Advances in MOX Fuel Technology, Technical Reports Series No. 415 (Issue 415). IAEA. https://www.iaea.org/publications/6562/status-and-advances-in-mox-fuel-technology

INTERNATIONAL ATOMIC ENERGY AGENCY. (2023a). Safety of Uranium and Plutonium Mixed Oxide Fuel Fabrication Facilities, IAEA Safety Standards Series No. SSG-7 (Rev. 1) (Issue SSG-7 (Rev. 1)). IAEA. https://www.iaea.org/publications/15079/safety-of-uranium-and-plutonium-mixed-oxide-fuel-fabrication-facilities

INTERNATIONAL ATOMIC ENERGY AGENCY. (2023b). Safety of Uranium Fuel Fabrication Facilities, IAEA Safety Standards Series No. SSG-6 (Rev. 1) (Issue SSG-6 (Rev. 1)). IAEA. https://www.iaea.org/publications/15078/safety-of-uranium-fuel-fabrication-facilities

ISO, & IEC. (2023a). ISO/IEC/IEEE ISO/IEC/IEEE 15288:2023 Systems and software engineering — Software life cycle processes. International Organization for Standardization, 1.

ISO, & IEC. (2023b). ISO/IEC/IEEE ISO/IEC/IEEE 24641:2023 Systems and software engineering —

Methods and tools for model-based systems and software engineering. International Organization for Standardization, 1.

Johnson, J. V, & Brabazon, E. J. (1993). Design of the Mox Fuel Fabrication Facility.

Luttmer, J., Prihodko, V., Ehring, D., & Nagarajah, A. (2023). Requirements extraction from engineering standards - systematic evaluation of extraction techniques. Procedia CIRP, 119, 794–799. https://doi.org/10.1016/j.procir.2023.03.125

McDermott, T., DeLaurentis, D., Beling, P., Blackburn, M., & Bone, M. (2020). AI4SE and SE4AI: A Research Roadmap. INSIGHT, 23(1), 8–14. https://doi.org/10.1002/inst.12278

Mengist, A., Buffoni, L., & Pop, A. (2021). An integrated framework for traceability and impact analysis in requirements verification of cyber–physical systems. Electronics (Switzerland), 10(8). https://doi.org/10.3390/electronics10080983

Nastov, B., Chapurlat, V., Dony, C., & Pfister, F. (2015). A verification approach from MDE applied to model based systems engineering: XeFFBD dynamic semantics. Complex Systems Design and Management - Proceedings of the 5th International Conference on Complex Systems Design and Management, CSD and M 2014. https://doi.org/10.1007/978-3-319-11617-4_16

Necula, S. C., Dumitriu, F., & Greavu-Șerban, V. (2024). A Systematic Literature Review on Using Natural Language Processing in Software Requirements Engineering. In Electronics (Switzerland) (Vol. 13, Issue 11). Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/electronics13112055

OMG. (2006). OMG SysML Specification. OMG Systems Modeling Language.

Patel, A., Maheshwaran, Y., & Santhya, P. (2024). Easing Adoption of Model Based System Engineering With Application of Generative AI. 2024 IEEE Space, Aerospace and Defence Conference (SPACE), 871–874. https://doi.org/10.1109/SPACE63117.2024.10667868

Petnga, L. (2019). Graph-based Assessment and Analysis of System Architecture Models. INCOSE International Symposium, 29(1). https://doi.org/10.1002/j.2334-5837.2019.00644.x

Plehn, C. (2018). A method for analyzing the impact of changes and their propagation in manufacturing systems (Vol. 333). Herbert Utz Verlag.

Roques, P. (2017). Systems Architecture Modeling with the Arcadia Method: A Practical Guide to Capella. In Systems Architecture Modeling with the Arcadia Method: A Practical Guide to Capella. https://doi.org/10.1016/C2016-0-00854-9

van Remmen, J. S., Horber, D., Lungu, A., Chang, F., van Putten, S., Goetz, S., & Wartzack, S. (2023). Natural language processing in requirements engineering and its challenges for requirements modelling in the engineering design domain. Proceedings of the Design Society, 3, 2765–2774. https://doi.org/10.1017/pds.2023.277

Weston, N., Chitchyan, R., & Rashid, A. (2009). A framework for constructing semantically composable feature models from natural language requirements. Proceedings of the 13th International Software Product Line Conference.

Winkler, J., & Vogelsang, A. (2017). Automatic classification of requirements based on convolutional neural networks. Proceedings - 2016 IEEE 24th International Requirements Engineering Conference Workshops, REW 2016. https://doi.org/10.1109/REW.2016.16

Zhang, J., & Yang, S. (2024). Recommendations for the Model-Based Systems Engineering Modeling Process Based on the SysML Model and Domain Knowledge. Applied Sciences (Switzerland), 14(10). https://doi.org/10.3390/app14104010

Zhao, L., Alhoshan, W., Ferrari, A., & Letsholo, K. J. (2022). Classification of Natural Language Processing Techniques for Requirements Engineering. http://arxiv.org/abs/2204.04282.

APPENDIX

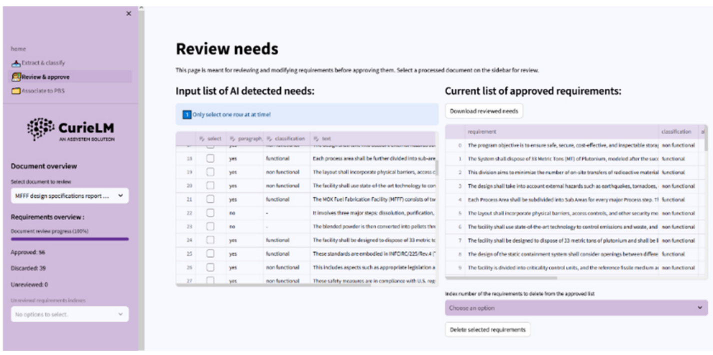

Figure 5: View of the requirements approval page.

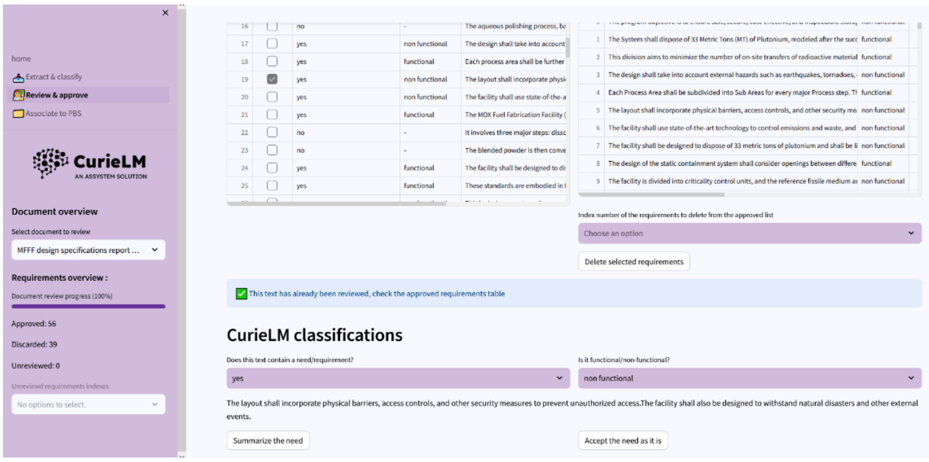

Figure 6: Classification and summary view.

Figure 7: INCOSE Auto Check View.

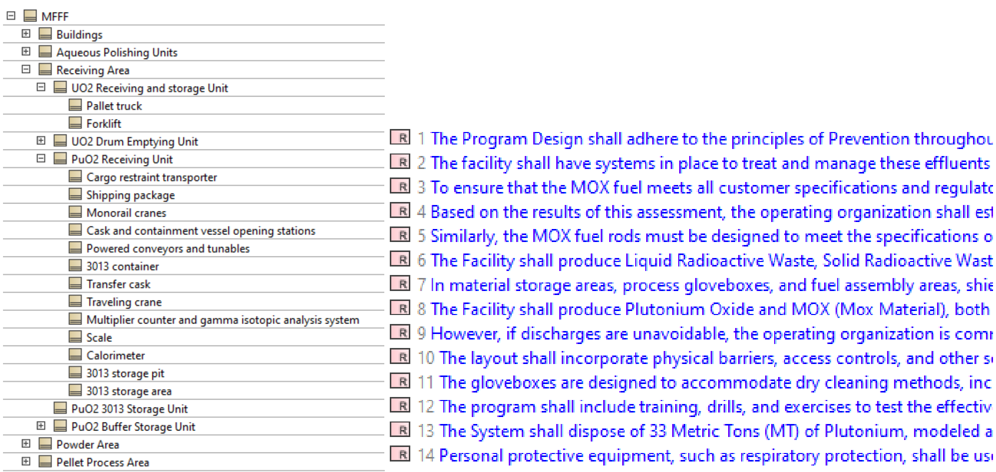

Figure 8: PBS and Requirements modelled in a system modelling tool.