DESIGN OF NEW SPACE MISSIONS WITH THE SUPPORT OF A DIGITAL ASSISTANT FOCUSED ON SYSTEMS ENGINEERING ACTIVITIES DURING THE EARLY DESIGN PHASES

Paloma Maestro Redondo[(1)], Gérald Garcia[(2)], Annalisa Riccardi[(3)], Gwendolyn Kolfschoten[(1)]

[(1)] Starion Nederland B.V
Schuttersveld 2
2316ZA Leiden, Netherlands
Email: info@stariongroup.eu

[(2)] Thales Alenia Space
26 avenue J.F. Champollion
31037 Toulouse Cedex 1 – France
Email: gerald.garcia@thalesaleniaspace.com

[(3)] University of Strathclyde
Aerospace Centre of Excellence
G1 1XJ Glasgow, UK
Email: annalisa.riccardi@strath.ac.uk

INTRODUCTION

The design of complex engineering systems requires an exhaustive and continuous verification and validation process at each stage of the mission lifecycle. While at the early phases of a project the system design might not yet include a high level of detail, it is still necessary to ensure that it meets the mission requirements. Even if the design data is recorded using Model-Based System Engineering (MBSE) tools, these do not usually offer integrated verification capabilities to cross check the design against the requirements.

Due to the fact that requirements are expressed in natural language, there have been recent efforts to investigate the applicability of Natural Language Processing (NLP) techniques to support requirements management tasks. Although requirements can be managed in a structured manner within the context of MBSE, they are still fully expressed in natural language. The availability of well-structured, normalised, and connected data is an advantage when applying artificial intelligence (AI) technologies, and this need can be met by utilising MBSE models to train the algorithms. The existing data generated from conducted studies using an MBSE approach provides a significant opportunity for systematic reusability.

The analysis of the design data using state-of-the-art NLP methods can facilitate the design process of new missions by providing engineers with check and recommendation mechanisms. The early design phases of space engineering projects can greatly benefit from this technology, providing support to both domain experts and systems engineers in the definition of new missions, reducing design time by learning from previous projects and improving the quality and the traceability between the design artefacts. By using structured ontologies, digital assistants can be also coupled to the tools used during the conceptual studies, like CDP4-COMET, to ensure integration across domains and with the later stages of a project.

To evaluate the potential of the latest AI techniques and their capability to assist systems engineers in their tasks we conducted a research project to identify the most common user needs that can be supported by AI-enhanced tools and to develop a prototype application that works using MBSE data. This effort was part of a Technology Development Element (TDE) program activity, involving a collaboration between Starion Group, Thales Alenia Space France, and the University of Strathclyde. In this context, we designed and developed a digital assistant (known as AIDA in reference to the name of the activity) for space system engineering, capable of learning from previous structured data modelled using MBSE approaches and analysing the requirements and system models defined in ongoing projects. The assistant identifies common concepts from past missions and proposes suggestions for new designs, accelerating the mission definition phase and preventing the repetition of previous mistakes.

This paper presents the proposed solution for the digital assistant, which consists of a web application integrated with the MBSE Hub and provides conceptual interoperability with the Space System Ontology (SSO). The paper outlines the application cases and demonstrates how the NLP techniques, fine-tuned for the space domain, allow users to analyse and determine relationships between requirements and MBSE artefacts.

METHODOLOGY

The latest research in the application of NLP for requirement engineering has been focused on developing solutions for requirements classification [1], detection of anomalies and similarities [2], requirements tracing [3], and big data approaches to mining user feedback for the purpose of developing more user-centric requirements [4]. Nevertheless, the more advanced and recent NLP techniques, based on contextualised word embeddings, still require further investigation for their potential applications in the field of requirements engineering and domain-specific contexts. As part of this project, we have developed an assistant for the space domain that uses state-of-the-art NLP methods to determine semantic similarities between text inputs, to fine-tune language models to extract domain-specific concepts, and to classify text input into specific categories.

Besides the implementation of NLP techniques which have been adapted to the space domain, the methodology also involved identifying the key potential use cases, collecting relevant datasets from past mission studies and designing the architecture for the assistant, including a simple user interface to facilitate users' interaction with the tool. All these points are explained in the following subsections.

Identification of Application Cases

The study commenced with a workshop conducted with experts from the European Space Agency (ESA), who are typically involved in concurrent design studies. The aim of this workshop was to identify and collect ideas for potential application scenarios of the digital assistant. Initially, a set of more than 80 use cases was compiled, which were subsequently processed and prioritised based on three criteria: the added value for the Agency, the technical feasibility for implementation, and the availability of data to train the models for these specific use cases.

The initial set of application scenarios was classified into four distinct categories: semantic search, recommendation, check/validation, and generative design. Additional information on the use cases identification and selection process can be consulted on the paper presented at SECESA 2022 [5]. Three use cases were selected for implementation and were further described by decomposing each one into several user stories, which respond to potential user needs. In addition, a validation process was defined based on these user stories to facilitate the testing of the tool. The three selected use cases can be summarized as follows:

1. Link low- and high-level requirements, check their traceability, propose additional requirements, and suggest or check the verification method.
2. Complete or update the design with components and their respective properties.
3. Verify that the specifications of the artefacts do not conflict with the requirements and inform the user if the model data does not fulfil the requirements.

Selection and Processing of Datasets

While investigating the latest NLP techniques that would best meet the identified needs, the consortium also gathered relevant datasets for the training of the algorithms and the validation of the model. The datasets were pre-processed to ensure consistency and compatibility with the AI techniques to be applied, thus facilitating the efficient use of the data. This entailed cleaning and normalising the textual data, as well as structuring the information into a knowledge graph. The dataset used included:

- A large set of specifications and requirements from an ESA project including their traceability links.
- Several CDP4-COMET models from past Phase 0/A missions.
- A Capella model that was linked to the CDP4-COMET models to form a structured set of data representing the design artifacts and their interconnections.

The datasets used are proprietary, so to avoid the risk of reverse engineering we published the results using a reduced set of this data together with open-source data.

Integration with the MBSE Hub and the Space System Ontology

The deployment of AI algorithms which need to be trained and used on large sets of engineering data is usually a challenge due to the unavailability of enough quality and structured data. ESA's OSMoSE (Overall Semantic Modelling for System Engineering) initiative aims to achieve semantic interoperability by defining a global conceptual data model, known as the Space System Ontology (SSO), and developing tools that support consistent model-based exchanges. Therefore, the

SSO has been used to provide the necessary structure and traceability to the data models, facilitating the application of the NLP algorithms.

The software architecture design aimed to accommodate three main objectives: enabling NLP techniques for addressing specific use cases, retrieving knowledge from SSO structured data models, and integrating the assistant with the MBSE Hub. Once the development of the assistant for the three use cases was completed, we created a web application that exposes the tool's functionalities and allows the connection to the Hub.

The high-level data exchanges used for this project are shown in Fig. 1. The datasets used by the AI assistant as input for the training of the models are formalised into a knowledge graph structured by the SSO ontologies. This knowledge is stored in the Data Hub, which in this case corresponds to the MBSE Hub developed by Starion as part of the Model Based Engineering Hub project [6]. A significant feature of the Hub's design is its high degree of code generation directly from the ORM model (in this case an SSO model extended with the project needs) using the Kalliope library developed and open-sourced by Starion [7]. The final prototype consists then of a web application integrated with the MBSE Hub and that provides conceptual interoperability with the SSO.

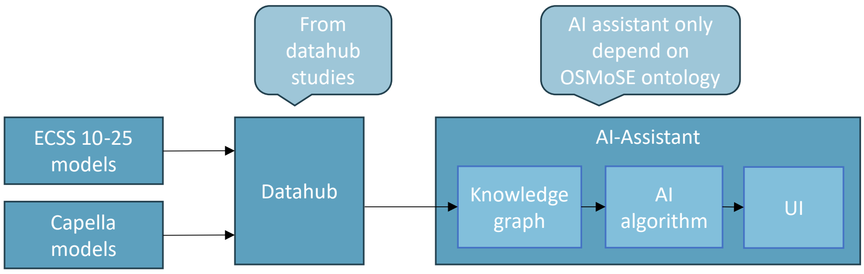

Fig. 1. Digital assistant structure and data exchanges

AI Techniques and Algorithms

In this project we propose to use NLP methodologies and a graph database for the analysis of data obtained from previous missions, with the objective of facilitating the design process of new missions. The development included techniques for analysing a database of past mission requirements, as well as the physical and functional architectures of past missions. The objective of these methods is to provide engineers with enhanced capabilities to define the requirements and the design for a new spacecraft in a more efficient manner. The methodology has been presented in [8], with the description of the techniques and the used algorithms: comparison of semantic similarity, filtered-breadth-first search, Large Language Models, and graph theory.

For the analysis of past mission requirements, we firstly developed techniques to identify semantic similarities between past mission requirements and a given new requirement. Furthermore, a language model is fine-tuned in order to analyse the logical traceability between two requirements. An additional functionality allows to propose or check the verification method for the requirements being analysed.

The physical and functional architecture of the systems defined using MBSE models can be also investigated, as well as the dependencies between the engineering model items, to be later proposed for new missions. Based on an input for a new design, a graph database of past-mission data can be queried for similar design choices and functionalities by again leveraging the abilities of semantic similarity and a specialised filtered breadth-first-search graph traversal algorithm. With this process, the architecture design of new missions can be expanded and completed by incorporating components, functions, and physical parts derived from past designs.

An example of this design analysis process in shown in Fig. 2. The given textual input is firstly encoded using a sentence embedding model. This could either be the name of a component or a description of a function as in the example, where it is shown in the blue box at the top of the figure. The next step consists of identifying the most semantic similar entity name in our graph database of past missions and use the breadth-first-search algorithm to traverse the graph initiating from this entity. When a node representing the concept of interest is found the search stops and all nodes traversed are captured and returned to the user. In this example the target nodes correspond to the components linked to the function used as starting point for the search. This process can support engineers to define and enhance the architecture design for new missions.

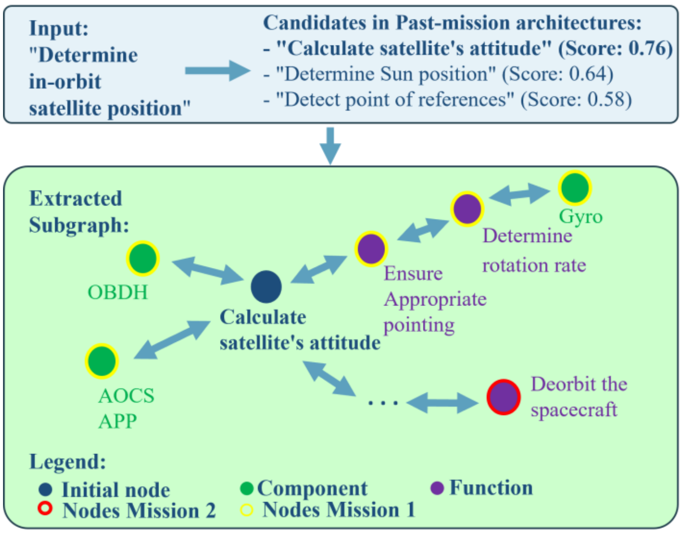

Fig. 2. Design analyses process consisting of identifying and extracting relevant information based on past designs

With the implementation for the last use case, we show how both the requirement and design analyses could be combined in order to automatically verify if the information stated in the requirements is reflected in the physical architecture. The overall process followed by the assistant is summarised in Fig. 3. A language model is used in this analysis to extract core concepts from a requirement, and in a second step, the concepts from the requirement are mapped to similar components present in the mission design graph. The neighbour components and the links to the root entity are also extracted. When the input consists of a requirement classified as being of the type "Performance", the attributes associated with the component are also extracted. To complete the last step, which is the verification, this information is used as input for a Large Language Model (LLM), which is prompted to reason if the requirement is fulfilled or not.

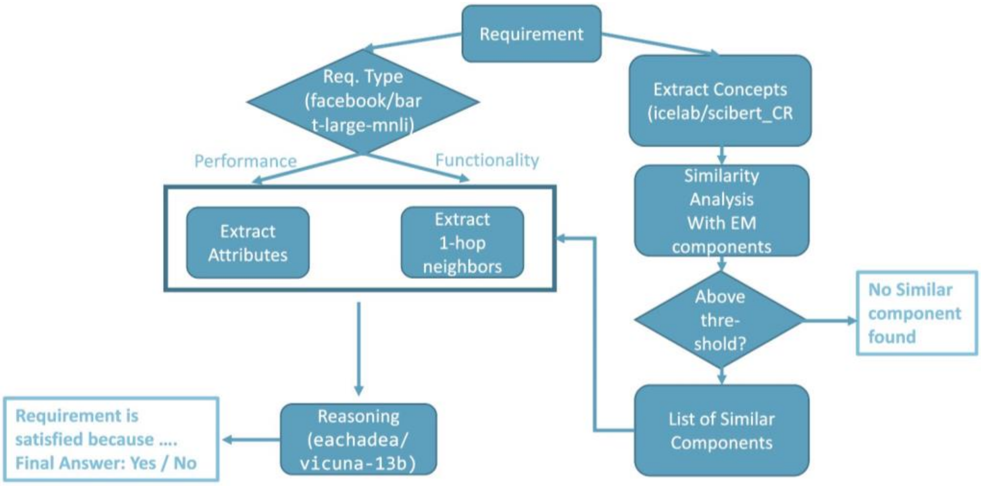

Fig. 3. Requirement verification process diagram

The prompt used for the LLM was:

> "I want you to act as an Spacecraft Engineer. I will provide an extract of an Engineering System in triple format as well as a requirement, and it will be your job to analyse if the requirement is satisfied by the engineering system or not. First try to extract the information (entities and relations) from the System that is potentially useful in your analysis. Next try to establish if that information either implicitly or explicitly fulfils the requirement. This could involve performing a gap analysis between the engineering system and the requirement. The Final step is to clearly state if the requirement is satisfied or not by stating FINAL ANSWER: YES or FINAL ANSWER: NO. My first engineering system and requirement are:"

The LLM used for this project is open-source (Vicuna-13b [9]), but as demonstrated in [8] more advanced and larger models would give more accurate results. The proof of principle and its effectiveness have been already demonstrated, but further quantitative testing should be conducted to improve the methodology and better assess its limitations.

APPLICATION TO CONCURRENT ENGINEERING

The methodology and techniques described above could significantly improve the results of concurrent design studies. By leveraging NLP and graph search techniques, this approach can make the design processes for new space systems more efficient and effective reusing information from past designs. In this way, engineers can take advantage of previous work and the data generated after each study to reuse the information in a systematic way, and reduce not just the time and resources needed to define the system, but also the errors in the design. A specific field of application could be in the context of New Space, where the added value would be in accelerating the verification of new systems and consequently shortening the time to market. Based on the current available functionalities of the assistant, the most common application scenarios identified in the context of concurrent engineering are outlined below:

- Identification of new requirements to be proposed for ongoing mission designs based on the similarity with the requirements included in the mission or system specifications.
- Detection of missing traces between requirements of the same or different specifications.
- Proposal of new design elements based on historical data and current project needs. This could be seen as a generative design capability that helps engineers explore innovative solutions while maintaining consistency with the project requirements.
- Suggestion of updated or new component properties to complete the model based on past designs.
- Verification of the model data to ensure that the design is compliant with the specified requirements.
- Identification of inconsistencies between requirements and the system design.

The recommended approach to integrate the assistant capabilities with the concurrent engineering processes would be to first train the potential users on the use of the tool. Even if its usage is intuitive and the algorithms are run in the background, it is necessary to understand the meaning of the outputs to be able to evaluate the design improvements suggested by the assistant. The second step would be to further train the digital assistant with more relevant sets of data, potentially MBSE models produced during past studies.

As the web application is integrated with the MBSE Hub, the experts or the systems engineers taking part in a study can directly analyse the data contained in their MBSE models stored in the Hub. As an example, if the CD team is using CDP4-COMET, the engineering model data can be retrieved by the digital assistant through the Hub for its evaluation. Users could then check and analyse specific parts of their model, like requirements, functions or other elements in the product tree, running the analysis available in the assistant, and decide whether to implement or not the recommendations provided as outputs. The current solution does not allow to propagate the model changes to the authoring tools linked to the Hub, but this should be investigated for future developments in this direction.

It is important to recall at this point that the datasets used as input for the assistant are formalised into a knowledge graph structured according to the SSO ontologies. It is therefore important to note that the same version of the SSO should be used by the assistant's graph database, the models used for training, and the MBSE Hub instance used for retrieving the model data which is to be analysed. Mapping between different versions of the SSO is not straightforward, so it is thus recommended to regenerate the software components based on a consolidated release of the SSO before deploying this methodology for new concurrent design studies.

THE DIGITAL ASSISTANT

The web application developed enables end-users to interact with the Digital Assistant (AIDA) by selecting the desired analysis to be executed on their data, together with the additional inputs and settings that may be required. Once connected to the MBSE Hub, users are able to retrieve information, such as requirements, from the Hub databases in order to perform an analysis directly on this data. The software solution stores the completed analysis and displays the outputs in the results page (Fig. 4). A dashboard page (Fig. 5) provides a summary of the project related data, the types of analyses that can be run, and statistics on previously completed analyses.

Furthermore, users receive a notification when an analysis is completed, allowing them to explore the results. The dedicated view for inspecting the analysis information (Fig. 4) includes the start date, time to completion, the type of analysis, inputs and outputs (with multiple display options), any errors that were encountered, the status, and the user's agreement on the provided recommendations. Users can give feedback to indicate if the suggestion from the assistant is accepted. This may in the future enhance additional machine learning based on the user feedback.

Fig. 4. Results page showing all the analysis related information

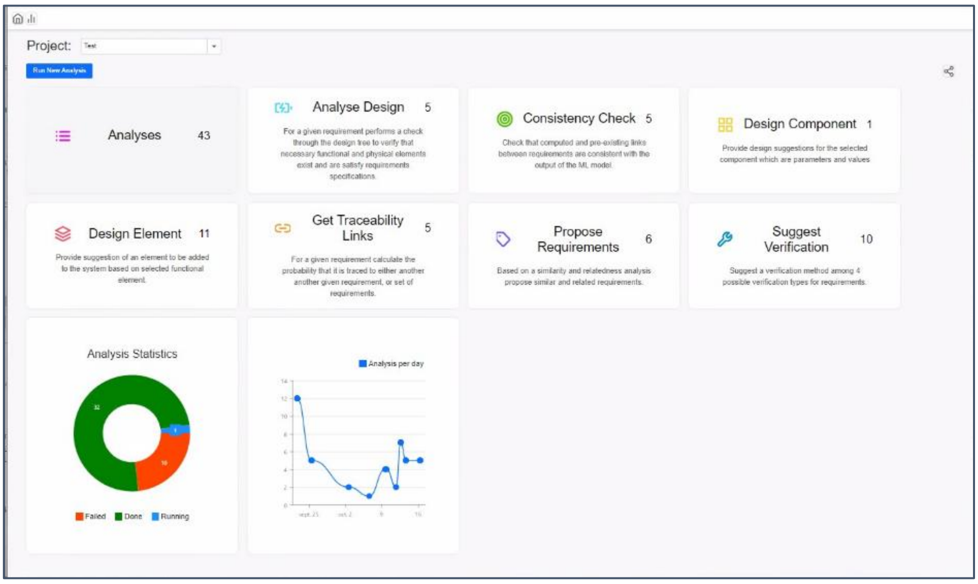

Fig. 5. Dashboard of the Digital Assistant

Case Studies Implemented

The three use cases selected for the development of the prototype for a digital assistant have been implemented on the tool using the AI algorithms and NLP techniques described in the methodology section of this paper. Fig. 5 shows the process to select the desired analysis and define the settings that need to be specified for each case. This process consists of three steps: users should first select one of the analysis kinds, which for the prototype represent the implemented use cases, then they can select the specific action to be performed on their data, and finally, they specify additional information and settings that are taken as input for the analysis.

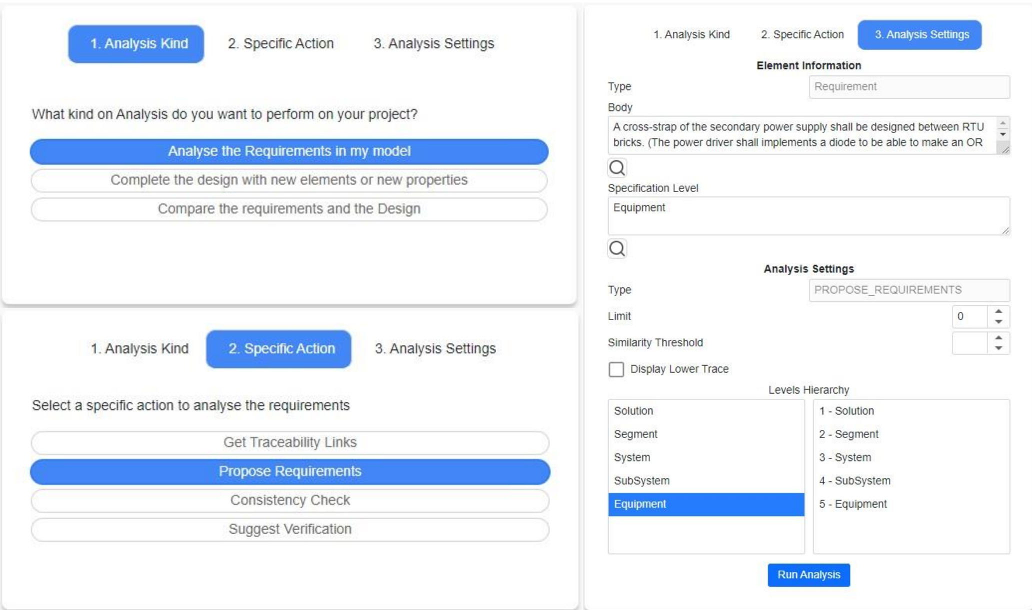

Fig. 6. Selection of the Analysis and Settings

A typical context in which the developed application can be used would begin with system engineers defining requirement specifications, which can be linked to other MBSE artefacts such as components or functions in the model authoring tools. This model data can be stored and transferred through the MBSE Hub to be used by other domain specific tools, like the assistant developed in this project. After being trained on past mission data, the assistant offers users recommendations for ongoing designs based on previous models. The user starts by selecting a project and connecting to the Hub, and then selects the kind of analysis they want to run on their data.

The screenshots in Fig. 6 show the example for the ‘Propose Requirements’ analysis type, where the input requirement can be selected from the actual project data sourced from the Hub. Users would get recommendations on similar or related requirements based on similarity metrics. Other analysis options related to requirements implemented in the context of the activity allow users to get traceability links with the related probability from a given requirement or specification to another specification (or container), to check the consistency of existing traces and to update or suggest the verification method of each requirement.

The second analysis kind can be used to provide recommendations for completing the model design with either new elements or new properties based on a selected input function. The last analysis uses an LLM to obtain a statement on model consistency based on the results from the two previous analysis types. The LLM performs a check through the design model to verify that necessary functional and physical elements exist and satisfy the requirements. An example of the results for this analysis is shown in Fig. 7, where the target requirement is compared with the design data and the LLM output provides a final statement on the verification result.

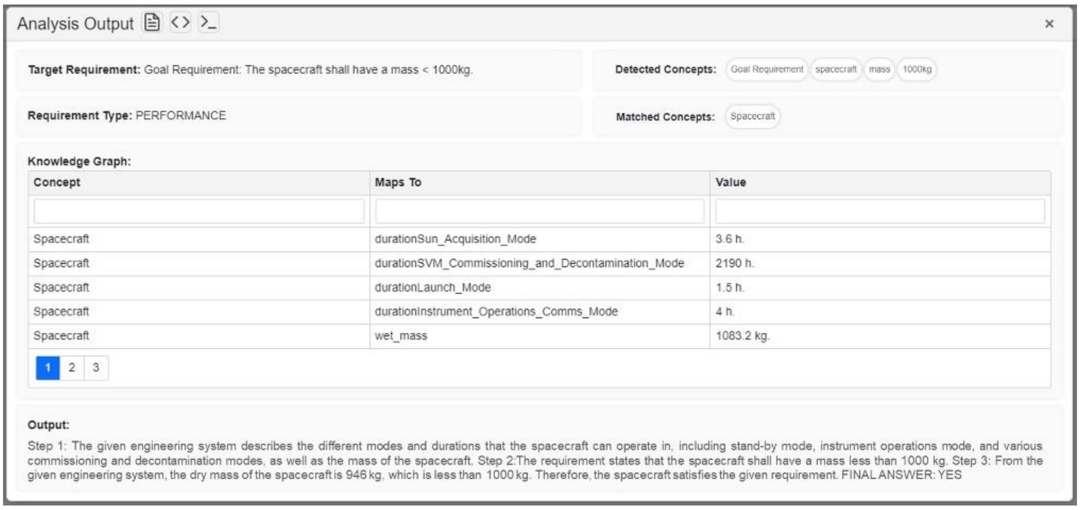

Fig. 7. Analysis results window displaying the LLM output

Testing and Validation

The digital assistant was subjected to a continuous testing and validation process to ensure its effectiveness and assess its reliability. This involved internal testing during the development and user testing following the validation plan to evaluate the assistant in real-world scenarios. Moreover, the performance of the AI language models and algorithms was evaluated in terms of accuracy, quality, and efficiency.

The results of the validation process indicate a positive outcome, although the reliability of the results heavily depends on the size and quality of the available data sources for training. Another key challenge is to improve performance. This would allow to process larger volumes of data for the analyses, including the ability to check complete models. It should be noted that, even when the application is connected to the MBSE Hub, the results cannot yet be used in the authoring tools. It would be beneficial to have the option to save results in the Hub, so that the recommendations provided by the assistant can be translated into the original model and displayed as revision points. This would improve the usability of the results, particularly in the context of a semi-automated review cycle.

CONCLUSION AND FUTURE WORK

The results of this study highlight the potential of leveraging NLP techniques to explore past-mission data and assist engineers in the early design phases for the definition of new space missions. We have designed and developed the digital assistant "AIDA" based on a web application solution, which includes three different analysis modes to check and improve the requirements and designs for new projects. Furthermore, the integration of the tool with the MBSE Hub shows how it can be an enabler for analysis using AI technologies, and how the SSO provides a consistent and structured data model to support tool interoperability. In this paper, we have also presented how the technology could be integrated with the method and tools used for concurrent engineering, summarising the main potential scenarios.

We illustrated through qualitative evidence that a fine-tuned language model can trace logical connections between requirements and predict verification methods at an acceptable level of accuracy. The explanation on the precision of the used algorithms has been described in [8]. This facilitates a detailed analysis of the requirement space for new space missions, helping system engineers to identify similar requirements together with all their dependent lower-level traced requirements. With the second analysis mode, the assistant offers the capability to analyse and explore the mission design of past projects with the objective of completing the definition of new systems. This is achieved by proposing updates to the design with new parts and properties for the components. For this purpose, the mission design information was formalised as a graph database and a filtered breadth-first search algorithm was deployed to query the graph based on a starting node for multiple different entity types.

The automatic requirement verification analysis showed how LLM reasoning capabilities can perform complex tasks and that prompting them with the correct contextual information can aid engineers in the verification process. For this analysis, the assistant first identifies a relevant section of the mission design graph based on the input requirement and subsequently prompts a LLM to reason whether the input requirement is fulfilled by the design or not. The results show that modern NLP techniques are capable of generalising effectively to a domain such as spacecraft engineering, even with minimal in-domain training data.

After evaluating the results, we can state that the use of digital assistants will enhance MBSE practices in the future, especially supporting projects where consistency between model artefacts is critical, and efficiently helping users to identify inconsistencies and errors. The improvement of the performance and accuracy of the algorithms directly depends on the availability of high-quality datasets needed for training and testing the models. As in many MBSE projects, proprietary data is a challenge. Therefore, finding a way for the assistant to deal with proprietary data would improve its usability. Moreover, there's concern about the Technology Readiness Level (TRL) of using Large Language Models (LLMs) for reasoning tasks. The assessment of the trustworthiness and resilience of these applications is still an active area of research.

Future studies could build upon the developed analysis modes or conduct test studies to evaluate the efficiency gains of using a digital assistant in the spacecraft design process. With respect to the methodology, the applicability of LLMs to the first two analysis modes could be investigated, as well as the capabilities of LLMs to support design model generation. The developed methodologies should also be tested to scale by including a larger set of missions and designs. Additionally, providing an end-user feedback mechanism that could participate to continuously update the AI models would greatly improve the efficiency and usability of the tool, and enhance the self-learning capability of the assistant integrating human feedback in the model. Other potential improvements have been identified in the results of the validation process, such as allowing users to systematically check the entire model for each analysis or integrating functionalities both in AIDA and in the MBSE Hub to enable the assistant results to become "usable" in the model authoring tools.

This activity is part of a broader set of roadmaps for technologies like AI and MBSE, aiming at reducing by 30% the development time and cost of ESA missions, in accordance with the Agenda 2025 [10]. It is expected that, thanks to the rapid advancements in the fields of AI and LLMs, the NLP techniques used in this activity will reach a higher TRL within few years and be used broadly for space mission design. The results of the activity, including the source and application code together with the supporting project documentation, can be accessed in the European Space Software Repository (ESSR) [11] under an ESA Community License.

REFERENCES

[1] E. Parra, C. Dimou, J. Llorens, V. Moreno, A. Fraga, "A methodology for the classification of quality of requirements using machine learning techniques", *Information and Software Technology*, vol. 67, pp. 180-195 November 2015.
[2] A. Ferrari, A. Esuli, "An NLP approach for cross-domain ambiguity detection in requirements engineering", *Automated Software Engineering,* vol. 26, pp. 559–598, June 2019.
[3] D. Falessi, G. Cantone, G. Canfora, "A comprehensive characterization of NLP techniques for identifying equivalent requirements", *IEEE International Symposium on Empirical Software Engineering and Measurement*, September 2010.
[4] O. S. Dawood, A.-E.-K. Sahraoui, "Toward Requirements and Design Traceability Using Natural Language Processing", *European Journal of Engineering and Technology Research,* vol. 3, no. 7, pp. 42-49, July 2018.
[5] P. Maestro Redondo, A Riccardi, G. Garcia, G. Kolfschoten, "Development of an AI-Powered Digital Assistant for Space System Engineering: Use Case Definition and Applications to Support Concurrent MBSE", *10th International Systems & Concurrent Engineering for Space Applications Conference (SECESA 2022)*, October 2022.
[6] A. Vorobiev, K. Tiensuu, S. Gerené, S. Jahnke, L. Bitetti, HP. De Koning, "Model Based System Engineering Hub", *MBSE 2022 Workshop*, 2022.
[7] Kalliope; https://github.com/STARIONGROUP/Kalliope.
[8] P. Darm, F. Marchetti, G. Garcia, P. Maestro Redondo, A. Riccardi, A. González Fernández, "Leveraging Language Models Semantic Similarity Capabilities to Facilitate Information Reuse in System Engineering", *International Astronautical Congress*, October 2023.
[9] Vicuna 13b Model, https://huggingface.co/lmsys/vicuna-13b-v1.5, last accessed 2024/07/03.
[10] ESA Agenda 2025, https://esamultimedia.esa.int/docs/ESA_Agenda_2025_final.pdf, last accessed 2024/07/03.
[11] European Space Software Repository (ESSR) - AI-Powered Digital Assistant (AIDA), https://essr.esa.int/project/ai-powered-digital-assistant-aida, last accessed 2024/07/03.