AUTHORS: Janica Bronson, Maryam Teimouri, Henrique Gaspar, Icaro Fonseca, Karolina Bierkowska, Filip Ginter, Herbert Koelman

TITLE: A RAG-based LLM Approach for Data Validation and Harmonization in Ship Design

YEAR: 2025

DOI: 10.5281/zenodo.17223682

VERSION: Publisher's PDF

CITATION: Bronson, J., Teimouri, M., Gaspar, H., Fonseca, I., Bierkowska, K., Ginter, F., & Koelman, H. (2025). A RAG-based LLM Approach for Data Validation and Harmonization in Ship Design. In V. Bertram (Ed.), *23rd Conference on Computer and IT Applications in the Maritime Industries: COMPIT'25* (pp. 171–181). International Conference on Computer and IT Applications in the Maritime Industries. https://doi.org/10.5281/zenodo.17223682

A RAG-based LLM Approach for Data Validation and Harmonization in Ship Design

Janica Bronson, NTNU, Ålesund/Norway, Janica.a.bronson@ntnu.no
Maryam Teimouri, UTU, Turku/Finland, mtebad@utu.fi
Henrique Gaspar, NTNU, Ålesund/Norway, henrique.gaspar@ntnu.no
Icaro Fonseca, NTNU, Ålesund/Norway, icaro.fonseca@ntnu.no
Karolina Bierkowska, NTNU, Ålesund/Norway, karolina.bierkowska@ntnu.no
Filip Ginter, UTU, Turku/Finland, figint@utu.fi
Herbert Koelman, NHL Stenden, Leeuwarden/The Netherlands, herbert.koelman@nhlstenden.com

Abstract

Validating ship design data across systems is challenging due to fragmented information from multiple sources, file types, and formats – from 2D drawings, 3D models, and specifications, often found in unstructured text files. While unified 3D models aim to serve as a single source of truth, ensuring accuracy and consistency across all representations remains a complex task. This paper presents a retrieval-augmented generation (RAG) solution for extracting and comparing design parameters from diverse files and formats. The approach aims to detect inconsistencies between documents and versions, helping designers maintain data integrity and reduce manual effort throughout the ship design process.

1. Introduction and Cost of Errors in Early Ship Design

The early concept design stage in naval architecture represents a critical phase where extensive data generation occurs under severe time constraints. This stage is characterized by intensive multidisciplinary collaboration and competitive bidding processes that require simultaneous development of numerous design documents across specialized domains, *Andrews (2018)*. Despite the inherent uncertainty and reliance on preliminary estimations, concept designs must rapidly converge to meet stringent bid requirements and project timelines. This phase exhibits unique constraints: (1) compressed timeframes with intense pressure, highly collaborative workflows requiring specialized expertise, *Le Poole et al. (2023)*, continuous validation of design parameters against performance thresholds, *Brathaug et al. (2008)*, and substantial uncertainty in design assumptions and calculations, *Jorge et al. (2018).*

The convergence of these factors creates an environment highly susceptible to errors that can propagate through subsequent design phases, with correction costs escalating exponentially as detail levels increase, as shown in Fig.1, *DeNucci and Hopman (2021).* Research indicates that early-stage design errors can result in cost overruns when discovered during detailed design or construction phases, and sometimes irreparable errors leading to high repercussions, *Andrews (2021), Rigterink (2014).* This sensitivity necessitates robust validation mechanisms to ensure parameter consistency and accuracy throughout the iterative design process.

Current industry practice relies heavily on traditional design spiral methodologies and concept variation methods (CVM) that involve multiple manual review cycles and version synchronization processes, *Papanikolaou (2018)*. However, these approaches face fundamental limitations in modern ship design environments, characterized by the fragmented nature of design information that spans 2D drawings, 3D models, specifications, and unstructured text files, *Bronson et al. (2024)*. While unified 3D and collaborative environments are increasingly promoted as single sources of truth, ensuring accuracy and consistency across all design representations remains a complex challenge, *Koelman et al. (2024)*. Studies reveal that engineers spend approximately 14% of their time locating information and verifying accuracy, representing a significant inefficiency in time-critical design phases, *Chui et al. (2023)*. Existing version control and change tracking mechanisms prove inadequate for managing the rapid iteration cycles characteristic of concept design. Moreover, current validation approaches require extensive manual synchronization between different systems and file formats.

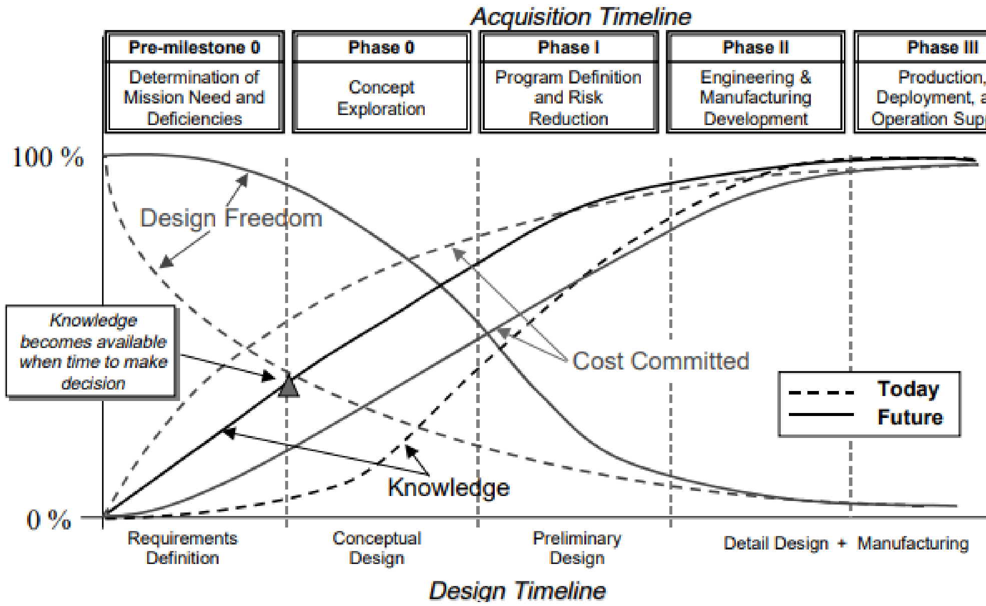

Fig.1: Relationship of committed costs and design freedom (adopted from *Mavris and DeLaurentis (2000)*)

This paper presents a novel approach involving retrieval-augmented generation (RAG) that addresses these fundamental challenges by automatically extracting and comparing design parameters from diverse file types and formats to reveal data inaccuracies and potential errors. Unlike traditional synchronization-based solutions that require comprehensive workflow restructuring and lengthy implementation periods, this method focuses on reviewing inconsistencies across documents and versions while preserving existing design processes. The key advantage of this approach is its non-disruptive integration with the current workflow; designers can continue using their preferred tools and established practices, while the solution is available to designers to validate parameter consistency.

2. Current Practice and Challenges

Despite the proliferation of advanced digital design environments, early-stage ship design validation remains heavily reliant on manual cross-referencing of heterogeneous data sources. Designers are often required to consult and compare design parameters from CAD models and hydrostatic calculations to spreadsheets and regulatory documents. This process is not only labor-intensive and error-prone but also constrained by tool interoperability and limited access to proprietary software platforms.

A core challenge lies in the fragmentation of design data across multiple formats and systems. Critical parameters - including principal dimensions, form coefficients, stability margins (e.g., GM), and design coefficient - are distributed across lines plans, general arrangements (GA), structural models, weight estimates, and machinery specifications. These parameters exhibit strong interdependencies: for example, hull form characteristics influence hydrostatic stability; structural arrangements affect weight distribution; propulsion requirements impact hull resistance and fuel consumption. Ensuring coherence across these dimensions necessitates continuous cross-document validation, which current workflows do not adequately support.

Moreover, regulatory compliance further complicates validation. Requirements elucidation is a core task that involves the synthesis of multiple regulations and guidelines from class. Design proposals must align with diverse and evolving standards, including SOLAS, MARPOL environmental regulations, Energy Efficiency Design Index (EEDI) thresholds, and classification society rules. These overlapping requirements generate a multi-objective validation landscape in which inconsistencies can propagate unnoticed, particularly when validation relies on manual inspection.

While modern software such as CADMATIC and AVEVA Marine supports model-based approvals, *AVEVA (2020), Yllikäinen (2019)*, they are primarily optimized for detailed design and approval stages - not early-stage concept design. Most tools validate geometry and compliance but overlook consistency across functional parameters. These and other validation techniques are discussed below:

1. Manual or in-house Validation Tools (Isolated) - Designers must manually extract and compare parameters from technical documents (e.g., line plans, hydrostatic reports, spreadsheets). This task is not only time-consuming but also highly susceptible to human error, especially as design iterations increase. Isolated scripts or digital checklists may help automate the validation of specific parameters (e.g., GM, LCG/LCB). While helpful, these solutions may not scale due to interoperability limitations.
2. Quality Assurance (QA) Procedures (Peer/External Review) - In many firms, validation is deferred to QA reviews. These reviews require cross-functional teams to manually synthesize inputs across disciplines, increasing the cognitive load. Although there is new research in this domain, there is a need for critical company buy-in for these QA processes and require dedicated personnel to carry through, *Hmeshah et al. (2015)*.
3. Model-Based Validation (OCX and Similar Standards) - OCX-based workflows and 3D model-centric platforms are designed to encapsulate validation within a unified geometric model. However, these models typically support only those parameters that can be directly visualized or geometrically mapped (e.g., structural members, arrangement boundaries). Alphanumeric parameters such as stability margins, performance coefficients, or other important design data currently still remains outside the scope of these models and must be validated separately, *Astrup (2022)*.
4. Software Tools – Class is also leading the development of new tools for validation. For example, the development of Nauticus Hull's Rules Check allows users to run their finite element analysis (FEA) against relevant cargo holding rules and thresholds, *DNV GL (2018)*. These tools, apart from DNV Nauticus Hull, include AMBS Eagle UDM, ClassNK PrimeShip-Hull, Lloyd's Register's RulesCalc, Korean Register's SeaTrust-HullScan, among others. However, these are mainly focused on structural validation.
5. Novel approaches – New solutions are being proposed by persons such as Soman (2015), who aim to improve the Smart Ship Design (S3D) environment by addressing the current lack of capability in evaluating design against engineering guidelines. The proposed solution uses Natural Language Processing (NLP) to extract design guidelines efficiently. However, the solution stops at the extraction level, *Soman et al. (2015)*, a gap this paper hopes to address.

3. Large Language Models (LLMs) and RAG

Large Language Models (LLMs), when combined with RAG, open new possibilities for assessing inconsistencies across technical documents. While LLMs provide context-aware reasoning over complex language, RAG enhances this capability by incorporating fresh, external data into the model's responses. By embedding and indexing technical documents, the system can instantly cross-reference them - allowing ship designers to ask questions such as, "Is the GM value consistent across all reports?" or "Does this hull design meet SOLAS and EEDI standards?"

Large pretrained language models are highly effective at retaining knowledge and retrieving factual information from their parameters. However, their effectiveness tends to decrease on downstream tasks that require expanding or updating their knowledge. Hybrid approaches that combine parametric memory with non-parametric memories can help address these limitations, as they allow knowledge to be revised and expanded more easily and quickly, *Lewis et al. (2020)*. *Siddharth and Luo (2024)* introduce a retrieval-augmented generation (RAG) framework specifically designed for ship design patents. It focuses on extracting named entities and their relational structures from patent texts to construct a structured, domain-specific knowledge base that supports more accurate and context-rich information retrieval.

Fig.2 illustrates the enhancement supported by RAG in the 3-step process of prompt or question answering, covering indexing of documents, retrieval of relevant documents based on semantic similarity, and input into the LLM for the generation of final answers. Existing research, such as *Soman (2015)*, is limited to filtering, highlighting, and extracting relevant rules from technical standards. By incorporating the generative reasoning of LLMs together with RAG's ability to retrieve and integrate external knowledge, these capabilities can be significantly extended. This enables ship designers not only to identify applicable rules but also to cross-validate them against new documents and evolving designs.

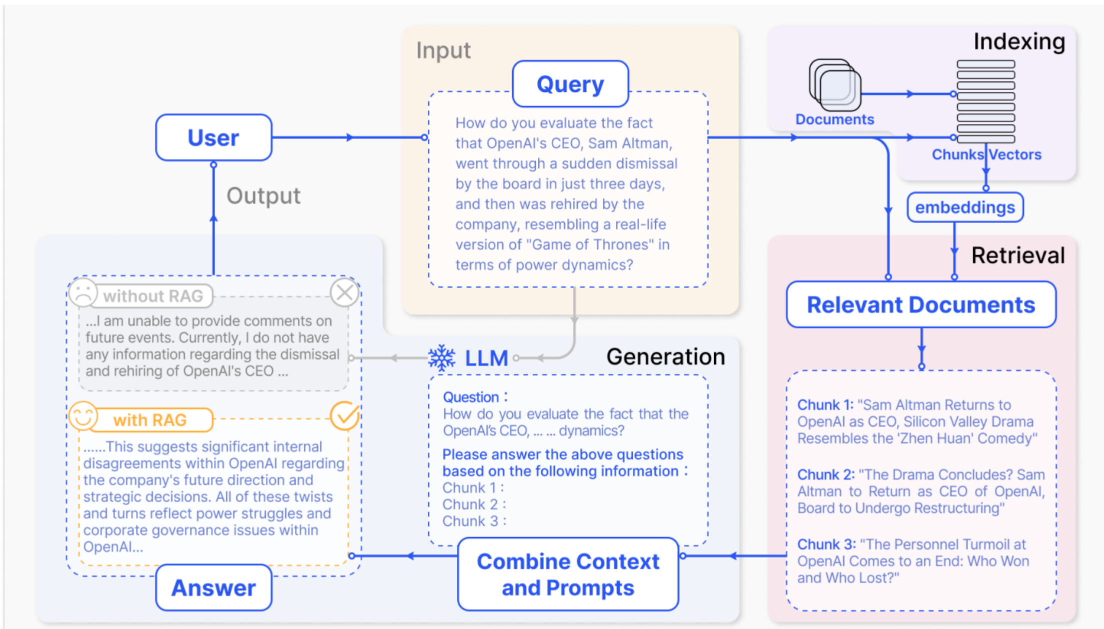

Fig.2: Representation of the RAG process enhancing LLM for prompting, *Gao et al. (2023)*

4. Case Study

To explore the potential of LLM-RAG validation approaches in early-stage ship design, we focus on a case study centered on the research vessel 'RV Gunnerus'. This case examines whether inconsistencies in design parameters can be effectively identified across multiple document types and subsequently reviewed against established design rules.

In this pipeline, we consider the usability of LLMs and RAG mainly in:

- Extracting parameters from unstructured PDFs and text files
- Cross-validating across internal design documents
- Highlighting inconsistent and regulatory requirements,

4.1. Parameters and Dataset

The study focuses on a core set of interrelated parameters commonly found in specification sheets such as principal dimensions (e.g., Length Overall, Beam, Draft), capacities, machinery data, equipment, and mission-specific facilities. Hence, the dataset used involved the 'RV Gunnerus' specification sheets, general arrangement (GA) drawings, hydrostatics inputs used for preliminary hydrodynamics tests, the 3D model, and equipment data. This original dataset contains: (1) the 3D model, (2) technical 2D drawings, and (3) text data in PDFs. The dataset was provided in part through research within the SEUS Project, which enabled access to NTNU ShipLab. For this case study, the data was used in its original form without any pre-processing.

In the next section, we discuss the key technical aspects of developing the AI assistant, including data preparation, system development, and model selection.

4.2. Methods

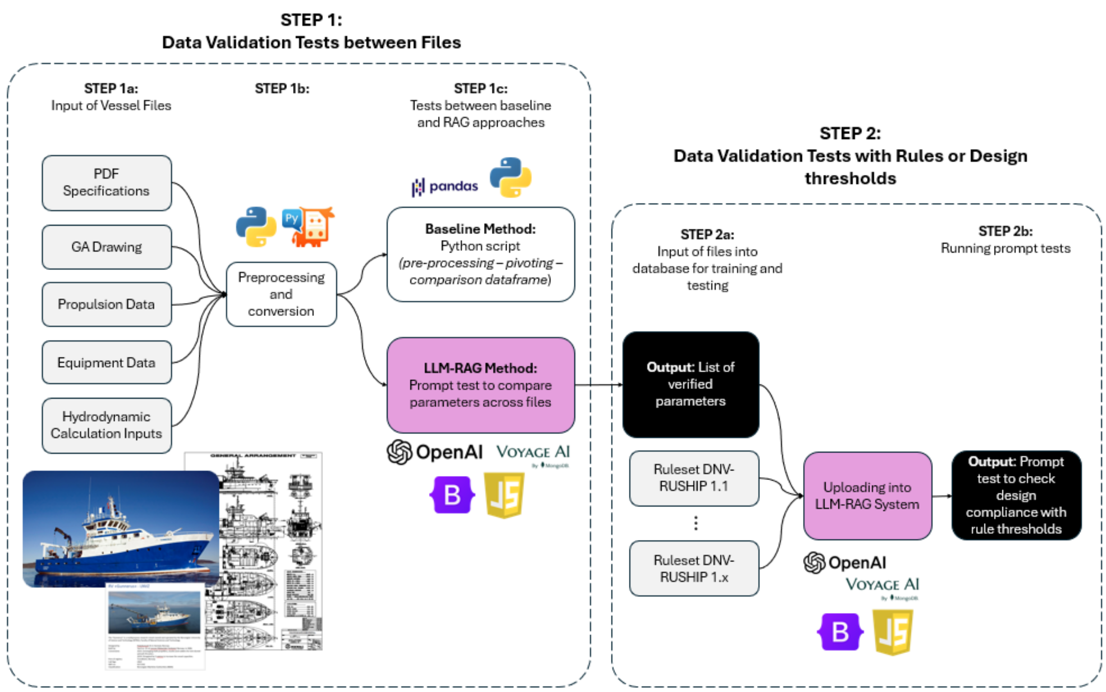

Fig.3: Methods for 2-step Validation of Parameters

The methods for this study employ a 2-step approach, as visualized in Fig.3, to cover the aforementioned goals:

1. <u>Data Validation Tests between Files:</u> For this step, the contents of the various design files are compared with each other. The files range from 2D, text, to 3D drawings and are pre-processed with the help of an LLM and further fed back into the model for prompt testing. In order to understand the gains in using the LLM-RAG model, we compare the results of this step with a Python script that automates the comparison of parameters between files.
2. <u>Compliance and Validation of design parameters against rules</u>: For this step, the vetted parameters are then compared to rulesets – an additional standard protocol for compliance and design validation. Given that *RV Gunnerus* has known notations, additional guidelines are fed into the model and used to assess whether the parameters are potentially compliant or not.

4.2.1. Data Processing

For data processing, the goal was to develop a scalable pipeline suitable for companies and organizations handling large volumes of data. Some of the data was already in text format, but a significant portion first had to be converted into images and then extracted as text, since the available metadata was not useful. To address this, we employed automated approaches using Python, with tools such as docx, pytesseract, PIL's Image module, and PyMuPDF (fitz).

As part of the data processing pipeline, we briefly investigated the integration of 3D model data into our system. The dataset provided contained primarily. prt and .x_t files, which are proprietary formats typically created with licensed software such as Siemens NX. These formats could not be processed directly using open-source Python libraries like pythonOCC. However, we identified converting these files into more accessible formats such as. STEP (.stp), STL, or. IGES would enable further processing and analysis. Since Siemens NX supports scripting for large-scale batch conversions, this conversion step can be incorporated into an automated workflow without compromising the scalability of the pipeline. To fully integrate the 3D models into the RAG framework, we need to define the relevant keywords or structural features to extract from the 3D data to support meaningful retrieval and knowledge augmentation. Achieving this would also require developing additional scripts for calculating dimensional properties. However, to keep the scope of this paper focused and to avoid potential errors from miscalculations, we limit our work here to 2D PDFs and text data.

4.2.2. Data Comparisons

For comparing the data of the files, we developed a Python script to complement the RAG method and compare the effectiveness of the RAG system. The Python script collects the pre-processed data, appends them into a data frame, and pivots the data frame such that only unique parameters are indexed and the different sources are concatenated into columns. These values are then compared in order to determine if they are consistent or inconsistent. The output results reveal a summary report in text file format.

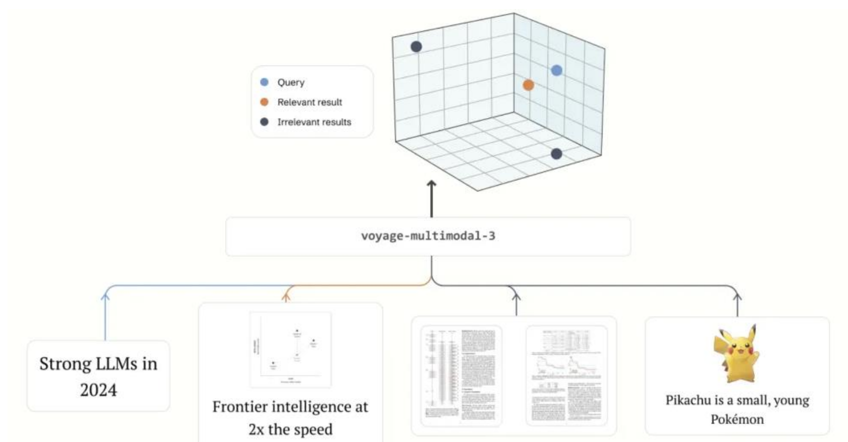

Fig.4: Latest Voyage AI Multimodal Framework, *VoyageAI (2024)*

4.2.3. RAG System

In designing the RAG system, we identified the need for two distinct machine learning models: one for embedding the data into a vector space for efficient storage and retrieval, and another as an LLM to generate coherent textual outputs. Given the multilingual nature of the dataset, it was essential to ensure that the system retrieves semantically relevant content based on context rather than language similarity. Through preliminary evaluations, the Voyage Multilingual model demonstrated the best performance in retrieving contextually accurate information across languages, *VoyageAI (2024)*. For text generation, we chose GPT-4o due to its state-of-the-art multilingual capabilities, low latency, and cost-effectiveness. These decisions build on the foundations laid by the original GPT-4 architecture, which demonstrated robust multilingual reasoning and generation across tasks, *Open AI et al. (2023)*.

4.2.4. Web Interface

The web interface was developed using Python and FAISS for the backend, and Bootstrap and JavaScript for the frontend. The system is designed to provide a chat-based environment where domain experts can pose questions and receive responses generated from contextually relevant documents. Key components of the system are fully configurable, allowing users to select the number of top retrieved documents (n-top), the similarity metric, the embedding model, and the language model (LLM) used for generation. A document viewer is integrated into the right-hand panel, enabling users to validate the generated answers by reviewing the source documents. These documents can also be downloaded for further inspection.

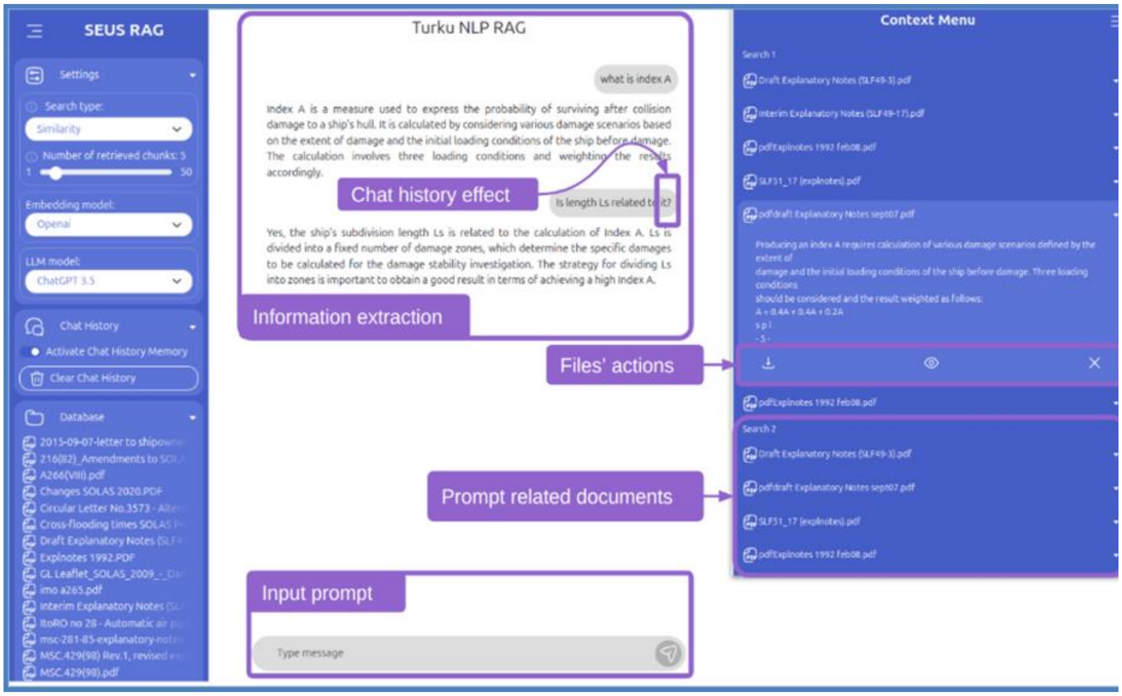

Fig.5: Web Interface for the RAG Solution

4.3. Results

Tests for the first step reveal that about 173 unique parameters were found. In the Python baseline approach, about 168 were known consistent parameters and about 5 parameters were deemed inconsistent. These parameters include moulded breadth, cargo hold volume, fuel oil volume, tonnage, and depth data. The output summary file is shown in Fig.6. However, there are limitations to this method in that semantically equivalent parameters were not compared. For instance, ‘loa’ was not compared to length overall.

In comparison, the LLM model claimed to detect about 16 inaccuracies from the post-processed 173 unique parameters and was able to review semantic similarities, given the prompt: ‘what inconsistencies do you find’. Fig.7 shows these parameters. It caught the same inconsistencies flagged by the Python script but also went further by identifying semantically similar terms and comparing them. For example, it correctly recognized that LOA and length overall refer to the same measure, as with LWL and length at waterline. It highlighted the differences in these values in the specification sheet and GA drawing. This discrepancy is likely due to the GA representing a different (lengthened) version of the vessel compared to the specification. The model was also able to distinguish between different types of water capacity, such as technical water versus potable water, and compare their values across documents - revealing, for instance, a 0.4 m^3 difference between the specification sheet and the GA.

However, the model also produced some less meaningful comparisons. In several cases, it compared terms against themselves (e.g., draught underside keel, net tonnage, trim, and water ballast volume), which resulted in misleading outputs. It also compared draught normative with draught at max load (conceptually different measures), and inconsistencies in the level of detail when describing the same propulsion equipment.

Overall, we find the RAG solution can determine more equivalent parameters, but we also observe that it tends to over-correct, showing high sensitivity to anomalies and often attempting to infer more differences than are actually present. Compared with the existing pipeline, we expect the RAG function to streamline the process by removing the need for manual execution of a Python script, offering greater convenience through the developed interface. The user-friendly interface also facilitates easier scrutiny of results, unlike the Python script where comparisons are hardcoded and require users to review the code directly to ensure nothing was overlooked.

```
============================================================
VESSEL SPECIFICATIONS ANALYSIS SUMMARY
============================================================
Generated on: 2025-08-19 at 13:55:20

OVERALL STATISTICS:
--------------------
Total parameters analyzed: 173
Consistent parameters: 168 (97.1%)
Inconsistent parameters: 5 (2.9%)

SOURCE FILES ANALYZED:
-------------------------
1. 165821B6 GA.xlsx
2. DI16-070M-625hk.xlsx
3. Input data for hydrodynamic calculations - RV Gunnerus - June 2021 - Detailed.xlsx
4. Input data for hydrodynamic calculations - RV Gunnerus - June 2021.xlsx
5. Spec.xlsx
6. gunnerus_propulsion.xlsx

Total source files: 6

⚠  INCONSISTENCY ALERT!
-------------------------
Found 5 parameters with inconsistent values across source files.

DETAILED INCONSISTENCY REPORT:
-----------------------------------

1. Parameter: breadth moulded
   ---------------------------
   Different values found:
   • 165821B6 GA.xlsx: 9.80 M
   • Spec.xlsx: 9.60 m
```

Fig.5: Results from Python Script

On top of the extraction piece, as per the tests for Step 2, we are also hoping to test how the RAG can be used to evaluate compliance of the given parameters against the rules. The vessel has the following rule notations: DNV + 1A1 + Ice C + E0 + R2 Cargo ship. The goal was to double-check how much the RAG can support design compliance, given the known and validated parameters against the rules. The model was populated with DNV rules DNVGL-RU-SHIP Parts 1 to 6 and the following prompt was ran: 'Does the RV Gunnerus specification comply with the attached DNV rules for a research vessel, given that it has the following rule notations <<DNV + 1A + Ice C + E0 + R2 Cargo ship>>?'

With the aid of breaking the prompt further down, the RAG-LLM was able to ascertain first what the notations means: 'The notation "1A + Ice C + E0 + R2" indicates that the cargo ship is classified with the following specifications: "1A" signifies a high ice class for vessels operating in ice-infested waters, "Ice C" denotes the ship's capability to navigate in light ice conditions, "E0" indicates the vessel has no restrictions on the use of electrical propulsion, and "R2" refers to the ship's compliance with specific requirements for reliability and redundancy in its systems.' However, for the specifications for each notation, the model advises consulting the latest DNV classification for the latest guidance, showing that while it can ascertain the rules, it is not able to sift through the parameters where thresholds were obvious. It is important to note that, at this stage, we do not expect the LLM to have the ability to perform more complex math and calculations in order to infer compliance.

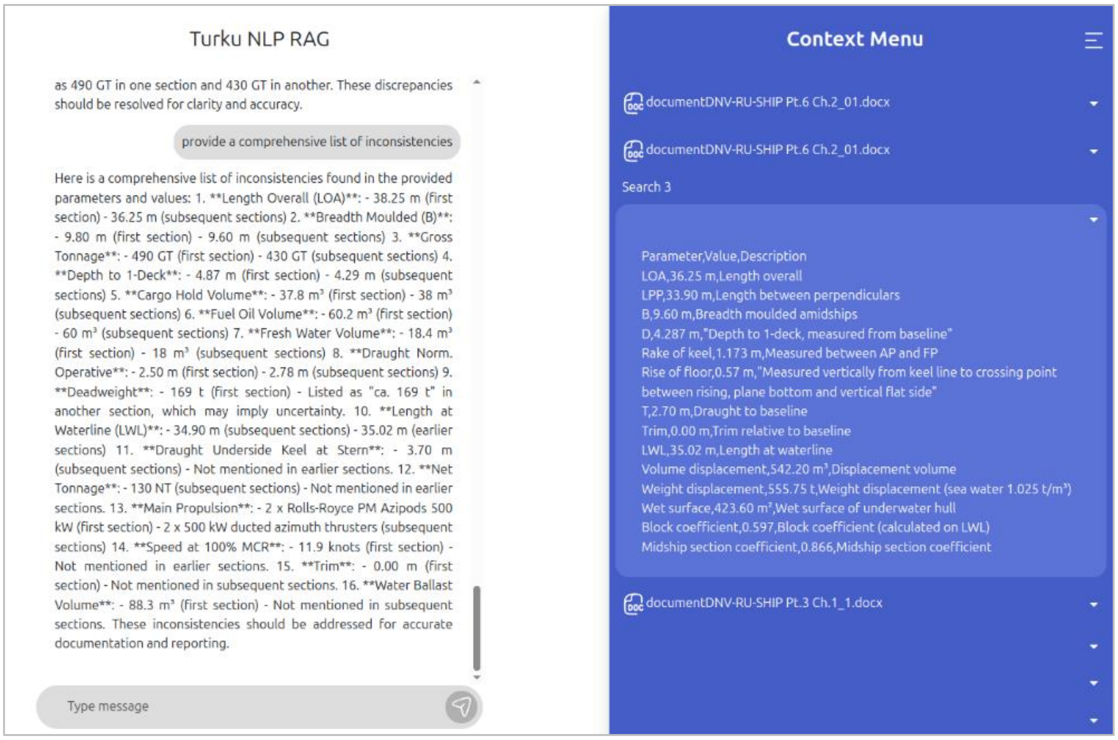

Fig.6: Results of STEP 1 and corresponding rulesets in the context sidebar

Fig.7 displays the interface starting with the prompts around design parameter inconsistencies, as introduced. The Context Menu demonstrates how these prompts are connected to the rule sets provided to the model. As additional rules and design data are incorporated, the interface has the potential to serve as a platform where designers can not only detect data inconsistencies but also evaluate them against the corresponding rules in the Context Menu – making the two validation steps not only possible but performed in parallel.

4.4. Discussions and next steps

There are various limitations and learnings from the case study that are subject to future improvements both for the model and the interface:

1. <u>Metadata</u>: Currently, the model only reads text and context data for retrieval. This makes it challenging to prompt the LLM to review more specific files and documents. The metadata of the files, including file name, version, and date of generation, is not currently considered, but would be helpful for future reference, allowing designers to easily point to specific documents.
2. <u>Uploading interface</u>: Alongside this development, a more user-friendly interface for uploading documents and retrieving them would be handy. To make testing and exploration easier, it would be helpful to add a drop box for uploading files that automatically updates the database.
3. <u>Secure database for corporate data</u>: Leveraging corporate data and deploying local instances of the model can tailor the LLM to a company's specific needs. By training it on corporate templates, designs, and terminology, the model becomes better aligned with organizational practices, making prompts for vessel types, project numbers, and other domain-specific inputs more intuitive and customized for the design team.

Noting these potential improvements, the addition of more and more data for training the LLM can increase the overall sensitivity and accuracy of the model. Further training is expected for the model so that it can become increasingly aware of maritime and ship design-specific semantics.

5. Conclusion

This paper has demonstrated the potential of an LLM-RAG solution for addressing one of the most persistent challenges in early-stage ship design: validating and harmonizing heterogeneous data. By comparing traditional scripted validation, we showed that the RAG system can capture not only explicit inconsistencies but also semantically equivalent parameters that are often overlooked.

While the current system shows sensitivity to anomalies and occasional over-correction, its ability to unify fragmented sources and provide designers with a transparent, user-friendly interface points to strong practical value. The approach does not disrupt existing workflows, reducing reliance on manual synchronization and enabling parallel checks for both parameter consistency and regulatory compliance.

Future work will focus on enriching the model with metadata, improving the upload interface, and tailoring models to organizational data for enhanced reliability. With continued refinement, the proposed solution can evolve into a scalable validation assistant, reducing design risks, lowering costs associated with late-stage errors, and ultimately accelerating the path toward more integrated digital ship design environments.

Acknowledgements

This paper is the result of the research project funded by the Smart European Shipbuilding (SEUS) Project (grant ID 101096224). SEUS is dedicated to advancing computational tools to enhance ship design and construction efficiency while also raising awareness of the current state of the art in maritime digitalization. Further insights and findings can be accessed at https://www.ntnu.edu/seus.

References

ANDREWS, D. (2018), *The Sophistication of Early Stage Design for Complex Vessels*, Int. J. Maritime Eng. 160(SE 18)

ANDREWS, D. (2021), *Design Errors in Ship Design*, J. Marine Science and Eng. 9(1)

ASTRUP, O.C. (2022), *Moving Towards Model Based Approval – The Open Class 3D Exchange (OCX) Standard*, Int. J. Maritime Eng.

AVEVA (2020), *AVEVA E3D Design*, AVEVA Inc.

BRATHAUG, T.; HOLAN, J.; ERIKSTAD, S. (2008), *Representing Design Knowledge in Configuration-Based Conceptual Ship Design*, 7th COMPIT Conf., Liege

BRONSON, J.A.; FONSECA, Í. ; & GASPAR, H.M. (2024), *Challenges Towards an Integrated Digital Twin Platform for Maritime Systems: Tackling Shifts in Data Ownership*, Offshore Technology Vol.1

CHUI, M.; ISSLER, M.; ROBERTS, R.; YEE, L. (2023), *Technology Trends Outlook*, McKinsey & Co

DeNUCCI, T.; HOPMAN, J. (2012), *Capturing Configuration Rationale in Complex Ship Design*, SWZ Maritime 10(133), pp.31–37

DNV GL (2018), *Digital Solutions Brochure: Nauticus Hull*, DNV GL, Hovik

GAO, Y.; XIONG, Y.; GAO, X.; JIA, K.; PAN, J.; BI, Y.; …; WANG, H. (2023), *Retrieval-Augmented Generation for Large Language Models: A Survey*, ArXiv, abs/2312.10997

HMESHAH, K.; ERBACH, N.; BRONSART, R. (2015), *Impact of production requirements on high-quality ship product data models*, J. Eng. for the Maritime Environment, 230(3), pp.508-522

JORGE, J.; AGIS, G.; EBRAHIMI, A.; KEANE, A. (2018), *Quantifying the effects of uncertainty in vessel design performance - A case study on factory stern trawlers*, IMDC Conf., Helsinki https://www.researchgate.net/publication/325797359

KOELMAN, H.J.; VEELO, B.N.; SEPPÄLÄ, L.; FILIUS, P. (2024), *Closing the gap between early and detailed ship design models*, IMDC Conf.

LE POOLE, J.; DUCHATEAU, E.; HOPMAN, H.; KANA, A.A. (2023), *Development and experimental testing of a collaborative design rationale method for early-stage ship layout design*, Int. J. Naval Architecture and Ocean Eng. 15

LEWIS, P.; PEREZ, E.; PIKTUS, A.; PETRONI, F.; KARPUKHIN, V.; GOYAL, N.; … KIELA, D. (2020), Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, https://doi.org/10.48550/arXiv.2005.11401

MAVRIS, D.; DeLAURENTIS, D. (2000), *Methodology for Examining the Simultaneous Impact of Requirements, Vehicle Characteristics, and Technologies on Military Aircraft Design,* 22nd Congress of the Int. Council on the Aeronautical Sciences (ICAS), Harrogate

OPENAI; ACHIAM, J.; ADLER, S.; AGARWAL, S.; AHMAD, L.; AKKAYA, I.; … ZOPH, B. (2023), GPT-4 Technical Report, https://doi.org/10.48550/arXiv.2303.08774

PAPANIKOLAOU, A. (2018), *A holistic approach to ship design: Optimisation of ship design and operation for life cycle (Vol. 1)*, Springer

RIGTERINK, D. (2014), *Methods for Analyzing Early Stage Naval Distributed Systems Designs, Employing Simplex, Multislice, and Multiplex Networks*, University of Michigan

SIDDHARTH, L.; LUO, J. (2024), *Retrieval Augmented Generation using Engineering Design Knowledge*, In S. U. o. T. a. Design (Ed.)

SOMAN, R.R.; ANDRUS, M.; BOSWORTH, M.; LEONARD, I.; STEURER, M. (2015), *Approach to develop ship design evaluation rule-base*, 2015 IEEE Electric Ship Technologies Symp. (ESTS)

VoyageAI (2024), *voyage-multimodal-3: all-in-one embedding model for interleaved text, images, and screenshots*, VoyageAI

YLLIKÄINEN, M. (2019), *Effective utilization of digital design assets in post-design phases*, https://www.cadmatic.com/en/resources/articles/effective-utilization-of-digital-design-assets-in-post-design-phases/#:~:text=More%20controlled%20verification,the%20information%2C%20thereby%20improving%20traceability