Integrating Generative AI for Enhanced Automation in System Design Processes

Jayesh Guntupalli
Services Computing Research Department
Hitachi Ltd.
Tokyo, Japan
jayesh.guntupalli.wv@hitachi.com

Kentarou Watanabe
Services Computing Research Department
Hitachi Ltd.
Tokyo, Japan
kentaro.watanabe.dc@hitachi.com

2024 IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA) | 979-8-3503-6123-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/ETFA61755.2024.10710979

***Abstract*—This work delves into the potential of applying Generative AI techniques to the system design phase, aiming to streamline processes and augment human expertise. Gen AI is used to automate the creation of complex design elements and is emerging as a powerful tool for system engineers. Also, with LLM-generated content, evaluating and verifying the correctness of the responses is a challenge. The SE Assistant designed for system engineers intends to create detailed system design documents quickly and accurately along with its evaluation. At the core of the SE Assistant is a sophisticated system that combines the power of GPT-4 with a Multimodal Retrieval Augmented Generation (RAG) pipeline capable of understanding text, images, and tables to provide valuable context. The evaluation uses the strength of strong LLMs in analyzing content based on design-specific criteria. The SE Assistant prototype demonstrates its ability to streamline the system design process, from initial data gathering to the final design output, making it an invaluable tool for system engineers.**

***Index Terms*—Generative AI , System Engineering, System Design, Large Language Model, Retrieval Augmented Generation.**

I. Introduction

The advent of Generative AI (Gen AI) and Large Language Models (LLMs) has assisted in a transformative era across multiple domains, ranging from natural language processing computer vision to complex decision-making systems [1]. This surge in Gen AI Market, especially in AI-driven innovation, is primarily characterized by the model's ability to generate novel content, interpret complex data, and provide previously unattainable solutions with traditional computational approaches. Among the different kinds of applications, one of the most notable is the integration of Gen AI and LLMs in System Engineering (SE), particularly in System design. This integration promises to enhance the efficiency and effectiveness of design processes and introduces novel methodologies for tackling intricate engineering challenges.

This research aims to answer the question: How can Gen AI improve the efficiency and accuracy of system design in complex systems such as autonomous transportation, traffic networks, and cloud-based software systems?

Gen AI, particularly through LLMs, has revolutionized SE, a field traditionally reliant on human expertise [2]. System Design, a critical phase in SE, involves creating complex blueprints that define system architecture, components, and interconnections. LLMs automate design tasks, generate innovative solutions, and provide insights from vast data, enabling rapid creation and evaluation of multiple design alternatives. This accelerates the design process, reduces costs, and enhances system quality [3].

However, integrating Gen AI into system design poses challenges. The complexity and criticality of system designs in various industries demand that LLM-generated outputs meet high standards of accuracy, reliability, and safety. Rigorous evaluation techniques are essential to verify the technical viability and compliance with industry standards, regulatory requirements, and ethical considerations. This paper introduces methods for system design using Gen AI to generate design information and discusses our proposed evaluation techniques. Our goal is to create a Gen AI-based platform for generating and evaluating Design Document (DesDoc) based on user requirements and system design principles.

II. Gen AI in System Engineering

A. Use of LLM in the System Design Phase

The system design phase is pivotal in determining a system's functionality, performance, and overall success. It involves intricate processes, including requirement analysis, conceptual design, architectural design, and detailed design. The utilization of LLMs in the System Design Phase represents a paradigm shift in how engineers approach these design processes. LLMs can analyze vast amounts of data, extract patterns, and generate complex outputs that aid decision-making and problem-solving. By leveraging these powers, LLMs enable engineers to streamline design tasks, enhance creativity, and optimize system performance.

B. System Design Evaluation

System Design Evaluation is essential in engineering to ensure proposed designs meet specifications, performance criteria, and safety standards. It validates functionality, efficiency, and reliability, reducing risks and costs. Traditionally, this involves analytical methods, simulations, and empirical testing to confirm operational requirements.

However, conventional methods can be resource-intensive, time-consuming, and sometimes impractical for complex systems. Theoretical analyses and simulations may not fully

capture real-world complexities, and empirical testing requires significant investment, delaying the process. Traditional methods may also struggle to keep pace with rapid technological advancements. Gen AI and LLMs offer a promising alternative, enabling rapid analysis, scenario simulation, and failure prediction based on historical data, potentially revolutionizing system design evaluation with more efficient, accurate, and flexible methods.

C. Existing methods for evaluating LLM-generated systen design specific content and their limitations

Existing methods for evaluating LLM-generated content include several metrics like BLEU [4], ROUGE [5], BARTScore [6], BERTScore [7] and GPTScore [8]. BLEU measures word usage precision by comparing machine output with human text, while ROUGE evaluates summarization tasks by assessing content overlap. BARTScore uses the BART model to check semantic coherence and factual correctness, BERTScore focuses on semantic similarity and GPTScore evaluates the likelihood of generating reference text from the candidate text. Fine-tuned LLMs predict content quality based on human-rated datasets, effectively assessing subjective tasks.

However, these methods face challenges in assessing system design-specific documents due to their specialized terminology and complex structures, rendering general metrics like BLEU and ROUGE less effective. They fall short in evaluating engineering principles, innovation, and practical applicability, as tools like BARTScore and GPTScore do not capture these aspects. While MT-Bench [9], Chatbot Arena [9], JudgeLM [10], and PandaLM [11] offer robust frameworks for general performance and scalability, they lack the tailored metrics needed for assessing the technical accuracy, reliability, and compliance of system design outputs, focusing instead on linguistic quality rather than real-world feasibility and innovation in system design.

In SE, evaluating the system DesDocs generated by LLMs like GPT-4 is crucial for ensuring that the final products meet the required quality and functionality standards [12]. Given the limitations of evaluation methods and human-led evaluation processes, mainly due to their time-consuming and costly nature, integrating advanced LLMs as evaluators presents a promising alternative.

III. System Design Generation and Evaluation Methods

A. SE Assistant for DesDoc Generation

The SE Assistant, as illustrated in Fig. 1, represents a tool for automating the creation of DesDocs. This tool leverages Multimodal Retrieval Augmented Generation (M-RAG) to enhance the capabilities of GPT-4 (multimodal) for system design-specific generation. The SE Assistant aims to transform input system requirements and constraints into comprehensive, detailed DesDocs. The SE Assistant operates on a multi-step process to generate DesDocs using the M-RAG architecture for providing context-specific to system design:

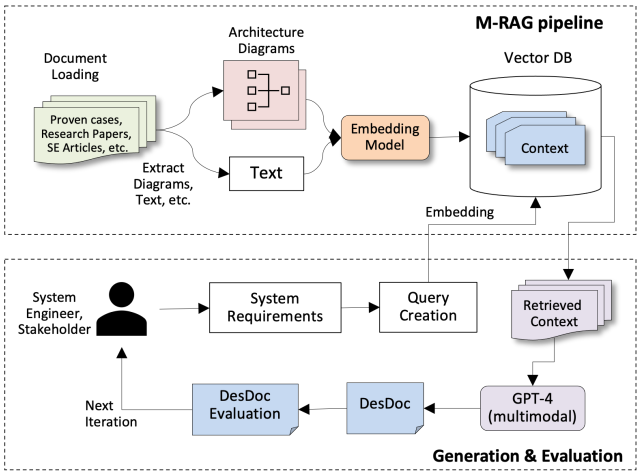

Fig. 1. SE Assistant tool architecture

a) System Requirement: The process starts by inputting system requirements and constraints, which form the foundational data for generating the DesDoc, detailing functional, performance, and regulatory constraints.

b) Query Creation: The SE Assistant crafts queries based on the input data, capturing the essence of the requirements to search for relevant information within the database.

c) Creating Embeddings: These queries are processed through an embedding model, converting natural language into mathematical vectors to match pre-indexed data in a vector database.

d) Vector Database: This database holds indexed vectors representing system design knowledge, design patterns, and previous project archives, serving as a retrieval source for the M-RAG system.

e) Retrieval Context: The embedding model retrieves the most relevant context from the vector database, aligning closely with the system requirements.

f) GPT-4 Augmentation: The retrieved context is fed into GPT-4, which uses it to understand the design requirements and generate informed content.

g) DesDoc Generation: GPT-4 synthesizes the retrieved and generated content to produce a comprehensive DesDoc, including descriptions, design solutions, diagrams, and essential components.

Now, with the system design-specific content generated by the LLMs, the next challenge is to evaluate these responses by GPT-4 to validate the generated content.

B. Generation

The process begins with generating the initial DesDoc using an LLM, which leverages system requirements and constraints as inputs to produce content theoretically aligned with desired outcomes. This phase employs the M-RAG pipeline to ensure the content is innovative, technically relevant, and applicable to the system's needs.

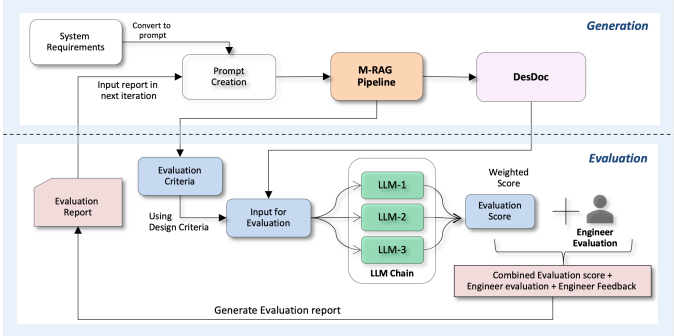

Fig. 2. Generation and Evaluation of DesDoc

C. Evaluation

Upon generating the initial DesDoc, it undergoes evaluation against predefined criteria crucial for system design. These criteria include technical accuracy, compliance with standards, feasibility, innovation, and alignment with system requirements. Fig. 2 illustrates the combined process for Generation and Evaluation of the DesDoc.

D. Generating Evaluation Criteria

The "Evaluation Criteria" are dynamically generated by the LLM, considering specific evaluation metrics. These criteria are uniquely defined for each iteration cycle, informed by the combined score from the Evaluation Score, Engineer Evaluation, system requirements, and established metrics. Special prompts guide the generation of these criteria, ensuring the evaluation process remains focused and aligned with the DesDoc's current state.

E. Inputs for Evaluation

The evaluation process utilizes two primary inputs: the DesDoc generated by the SE Assistant and the dynamically generated Evaluation Criteria. The DesDoc provides the content to be assessed, while the evaluation criteria establish the standards for assessment.

F. Evaluation Using Multiple LLMs

A chain of robust LLMs are employed to evaluate the DesDoc, each offering unique strengths for a comprehensive and unbiased appraisal. Using multiple LLMs mitigates bias, and their outputs are combined into a weighted average score, forming the Evaluation Score.

G. Calculation of the Evaluation Score

The Evaluation Score is calculated as follows:

$$\textit{Evaluation Score} = \frac{\left(\sum_{i=1}^{k} W_i\right)\left(\sum_{j=1}^{n} c_j\right)}{3}$$

Where:
W_i are the weights assigned to each LLM, reflecting their relative importance
c_j are the criteria scores for each evaluation metric, determined by how well the DesDoc meets each criterion defined by the dynamically generated "Evaluation Criteria,"
and k is the number of LLMs in the LLM-chain.

H. Importance of Engineer Evaluation

While LLMs offer systematic, data-driven evaluations, human engineers provide critical insights into practical aspects such as manufacturability, integration, and user experience. Engineers' evaluations ensure results align with human standards and expectations, checking LLM evaluations. Also engineers possess deep knowledge of technical requirements, contextual relevance, feasibility, and implementation, identifying challenges that LLMs might overlook.

I. Combined Evaluation

After calculating the Evaluation Score, the DesDoc is also assessed by an engineer. This dual approach ensures a comprehensive review, with the Evaluation Score providing a data-driven baseline and the engineer's input adding critical human insight.

IV. Implementation Flow

A. Defining Evaluation Criteria

Evaluation Criteria are essential for benchmarking the DesDoc's quality, encompassing technical accuracy, innovation, compliance, and practical implementation. The SE Assistant expands these criteria by integrating industry standards, best practices, and insights from similar projects, ensuring they are comprehensive and reflect current industry trends. The final set of criteria, which can be updated as the project evolves, is documented for transparency and consistency in evaluations, providing a clear benchmark for assessing the DesDoc.

B. Generation Flow

- *Requirement Ingestion:* The SE Assistant ingests specified system requirements and constraints.
- *Prompt Creation:* It formulates prompts that encapsulate these requirements, guiding the generation process.
- *M-RAG Pipeline Activation:* Prompts are processed through the SE Assistant's M-RAG pipeline, consisting of an embedding model and a vector database.
- *Embedding:* The embedding model translates prompts into vector representations.
- *Context Retrieval:* Vectors retrieve relevant context from the database, informing the generated DesDoc.
- *DesDoc Generation:* GPT-4 generates a preliminary DesDoc addressing the requirements and constraints.
- *Output:* The output is a comprehensive DesDoc proposing solutions and architectures to meet the system's needs.

C. Evaluation Flow

- *Input for Evaluation:* The generated DesDoc and established Evaluation Criteria are the inputs.
- *LLM Evaluation:* GPT-4, Claude-2, and Gemini evaluate the document against the criteria, each assigning a score from 1 to 10.

- *Score Aggregation:* Scores from the LLMs are aggregated into a preliminary Evaluation Score by taking the weighted average based on their reliability and domain expertise.
- *Engineer Review:* A human engineer reviews the document, providing a score and detailed comments, adding domain-specific expertise and practical judgment.
- *Combined Score Calculation:* The LLMs' Evaluation Score and the engineer's score are combined into a Combined Score, reflecting the engineer's expertise relative to the LLMs.
- *Improvement:* Combined score is used to improve the Evaluation Criteria and the DesDoc.

This method intertwines generative and evaluative processes, leveraging the strengths of both AI and human intelligence for a more reliable, innovative, and feasible system design. In the procedural flow, LLMs provide initial evaluations, and a system engineer's insights ensure robust, nuanced evaluation, combining AI's breadth of knowledge with human expertise.

V. Discussion

A. Results

The resulting output of this approach was an LLM-generated DesDoc that underwent a rigorous evaluation process. The evaluation summarized the feedback from three different LLMs—GPT-4, Claude-2, and Gemini. We defined the Evaluation criteria for benchmarking and scoring the DesDoc on Performance, Scalability, Reliability, Usability, and Security.

Evaluation Score =

$$\frac{(\sum_{i=1}^{3} W_i) * [P + S + R + U + Se]}{3}$$

Where:
W_i are the weights assigned to GPT-4, Claude-2 and Gemini respectively.
P, S, R, U, Se are the criteria scores for Performance, Scalability, Reliability, Usability and Security respectively.

We generated 20 sets of detailed system requirements for different use cases in system design and tested the DesDoc generation using GPT-4, text-only RAG, and finally with M-RAG. Table I shows the percentage of system requirements met in DesDocs using M-RAG (SE Assistant) in both Single-vendor (e.g., only AWS environment) and Multi-vendor (combination of multiple vendor environments) setups. The results indicate that M-RAG outperformed the other two methods. M-RAG's higher efficiency is also evident in the average evaluation score and combined score obtained when the DesDocs corresponding to each set of system requirements were evaluated using the SE Assistant. We observed that with the improved context provided by M-RAG, compared to GPT-4 and text-only RAG, we achieved the generation of better DesDocs.

TABLE I
Result of Evaluation using SE Assistant

Method	***Requirements met (in %)***		***Evaluation Score Normalized (out of 10)***	***Combined Score Normalized (out of 10)***
	Single-vendor	**Multi-vendor**		
GPT-4	35	32	7.4	7.2
RAG[Text-only]	46	38	8	7.8
M-RAG [SE Assistant]	64	51	8.6	8.5

B. Conclusion and Future Work

This research addressed the question: How can Gen AI improve the efficiency and accuracy of system design documentation in complex systems? The SE Assistant tool, combining GPT-4 with a M-RAG pipeline, significantly enhances the generation and evaluation of DesDocs for complex systems. It excels in integrating textual, visual, and tabular data to produce accurate DesDocs, with robust LLMs ensuring high standards of technical accuracy and compliance. Future work will enhance the SE Assistant's capabilities by incorporating dynamic, context-aware multi-agent mechanisms for integrating human feedback into the AI's learning process, further enhancing the SE capabilities. In conclusion, Gen AI can improves the efficiency and accuracy of system design documentation in complex systems, providing a valuable tool for system engineers.

References

[1] B. Tom et al., "Language Models are Few-Shot Learners," Advances in Neural Information Processing Systems, vol. 33, 2020
[2] Fan, Angela, et al. "Large language models for software engineering: Survey and open problems." arXiv preprint arXiv:2310.03533 (2023).
[3] Hou, Xinyi, et al. "Large Language Models for Software Engineering: A Systematic Literature Review." ArXiv, 2023, /abs/2308.10620.
[4] Papineni, Kishore, et al. "Bleu: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002.
[5] Lin, Chin-Yew. "Rouge: A package for automatic evaluation of summaries." Text summarization branches out. 2004.
[6] Yuan, Weizhe, Graham Neubig, and Pengfei Liu. "Bartscore: Evaluating generated text as text generation." Advances in Neural Information Processing Systems 34 (2021): 27263-27277.
[7] Zhang, Tianyi, et al. "Bertscore: Evaluating text generation with bert." arXiv preprint arXiv:1904.09675 (2019).
[8] Fu, Jinlan, et al. "Gptscore: Evaluate as you desire." arXiv preprint arXiv:2302.04166 (2023).
[9] Zheng, Lianmin, et al. "Judging llm-as-a-judge with mt-bench and chatbot arena." Advances in Neural Information Processing Systems 36 (2024).
[10] Zhu, Lianghui, Xinggang Wang, and Xinlong Wang. "Judgelm: Fine-tuned large language models are scalable judges." arXiv preprint arXiv:2310.17631 (2023).
[11] Wang, Yidong, et al. "Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization." arXiv preprint arXiv:2306.05087 (2023).
[12] Gao, Mingqi, et al. "Llm-based nlg evaluation: Current status and challenges." arXiv preprint arXiv:2402.01383 (2024).