

## Evaluating The Usability Of Chatbot-Based Collaborative Modeling For Class Diagram Development

<sup>1</sup>B. Prashant,<sup>2</sup>Peddisetti Naga Venkata Rajeswari,<sup>3</sup>Thatikonda Durga Bhavani,<sup>4</sup>Katuri Yaraswi,<sup>5</sup>Chintada Harikrishna

<sup>1</sup>Associate Professor & Head Of Department, Department of CSE-Data Science, Eluru College of Engineering and Technology

<sup>2,3,4,5</sup>B. Tech Student, Department of Computer Science & Engineering-DS, Eluru College of Engineering and Technology

### ABSTRACT

Recent advancements in natural language processing have enabled the growing adoption of chatbots in collaborative software engineering activities, including diagram modeling. Families of experiments provide a systematic approach for evaluating tools and processes while reducing common limitations of individual studies, such as small sample sizes and potential bias. This study aims to compare the usability of a chatbot-based collaborative modeling tool (SOCIO) with an online web-based tool (Creately). A family of three experiments was conducted in academic settings to evaluate usability and performance differences between the two approaches. The results show that participants were able to construct class diagrams more quickly using the chatbot, achieving an overall task accuracy of 96.4%, while also reporting higher satisfaction compared to the online collaborative tool. Furthermore, diagrams created with the chatbot tended to be more concise, although slightly less complete. These findings suggest that chatbot-based systems can effectively support class diagram development and provide a foundation for future research on the application of chatbots in collaborative diagram modeling.

**Keywords:** Chatbot-Based Modeling, Collaborative Modeling, Class Diagram Development, Usability Evaluation, Natural Language Processing (NLP), Human-Computer Interaction (HCI).

### I. INTRODUCTION

The rapid evolution of Artificial Intelligence (AI) and Natural Language Processing (NLP) has significantly transformed the way software engineering activities are performed. Traditionally, tasks such as requirement analysis, system modelling, and diagram creation required specialized tools and technical expertise. Among these tasks, Unified Modeling Language (UML) diagram construction plays a vital role in object-oriented system design, particularly in representing system structure through class diagrams. However, conventional modelling tools often demand a strong understanding of both UML notation and the tool's interface, which can create challenges for novice users, especially students in academic environments.

Online collaborative modelling platforms such as Creately allow multiple users to design class diagrams simultaneously. While these tools provide rich graphical features, they often present a steep learning curve. Users must manually create classes, define attributes and methods, and draw relationships between components. In many cases, there is limited real-time guidance, minimal error detection, and no intelligent support to ensure correctness or conciseness. As a result, diagrams may become overly complex, inconsistent, or visually cluttered. These limitations can reduce efficiency, increase task completion time, and negatively impact user satisfaction.

With the advancement of conversational AI, chatbots have emerged as promising assistants in software

development activities. Instead of interacting through graphical menus and toolbars, users can describe system components using natural language. The chatbot interprets the input, extracts modelling elements, and automatically generates structured diagrams. This conversational approach reduces technical barriers and enhances accessibility, especially for beginners. It also opens opportunities for real-time feedback, intelligent suggestions, and guided learning.

This project focuses on the use of the SOCIO chatbot for UML modelling and evaluates its effectiveness through a family of experiments. A family of experiments consists of multiple related experimental studies conducted under varying conditions to improve the reliability and validity of results. Rather than relying on a single experiment with a limited participant group, this approach reduces bias and increases confidence in findings. The objective is to compare the usability of SOCIO with Creately in academic settings, specifically for collaborative class diagram construction.

The SOCIO chatbot enables users to interact using natural language to define classes, attributes, methods, and relationships. It automatically structures diagrams, detects inconsistencies, and provides corrective suggestions in real time. By supporting collaboration and dialogue-based modelling, SOCIO aims to improve efficiency, reduce diagram errors, and enhance overall user satisfaction. Experimental results indicate that students were able to complete modelling tasks faster using SOCIO, produced more concise diagrams, and expressed higher levels of satisfaction compared to traditional tools.

In summary, this study explores the growing role of chatbot-assisted modelling in software engineering education. By investigating usability, performance, and user experience through a family of experiments, the research contributes to understanding how conversational AI can reshape collaborative UML diagramming and potentially redefine future modelling practices.

## II. LITERATURE SURVEY

### 1. Conversational Agents in Software Engineering

**Author(s):** Storey, M. A., Zagalsky, A., Filho, F. F., Singer, L., & German, D. M. (2017)

**Abstract:**

This study investigates the role of conversational agents in supporting software development tasks. The research highlights how chatbots assist developers in activities such as documentation retrieval, task automation, and collaboration. Conversational interfaces reduce context-switching and improve workflow efficiency. The study emphasizes that chatbots enhance productivity by providing real-time responses and simplifying complex tasks. However, limitations include dependency on accurate natural language understanding and integration challenges with existing development environments.

### 2. Automated UML Diagram Generation from Natural Language

**Author(s):** Ambriola, V., & Gervasi, V. (2006)

**Abstract:**

This research explores techniques for transforming natural language requirements into UML diagrams automatically. The framework uses Natural Language Processing (NLP) to extract classes, attributes, and relationships from textual descriptions. Results show that automated modelling tools reduce manual effort and improve consistency in diagram creation. However, challenges include ambiguity in natural language and the difficulty of accurately interpreting complex requirements without human supervision.

### 3. Collaborative Modelling Tools in Education

**Author(s):** Suthers, D. D. (2006)

**Abstract:**

This paper examines collaborative modelling

environments used in educational settings. Online tools such as Creately enable multiple users to co-create UML diagrams in real time. The study finds that collaborative tools enhance teamwork and communication skills among students. However, usability issues such as interface complexity, lack of intelligent guidance, and absence of automated error detection can hinder novice learners and slow task completion.

#### 4. Usability Evaluation through Family of Experiments

**Author(s):** Basili, V. R., Shull, F., & Lanubile, F. (1999)

**Abstract:**

This study introduces the concept of a “family of experiments” in software engineering research. Instead of relying on a single experiment, multiple related experiments are conducted to improve result validity and reduce bias. The findings demonstrate that families of experiments provide stronger empirical evidence when evaluating tools and methodologies. This approach is particularly useful when assessing usability, performance, and user satisfaction in academic contexts.

#### 5. Intelligent Tutoring and Real-Time Feedback Systems

**Author(s):** VanLehn, K. (2011)

**Abstract:**

This research analyzes intelligent tutoring systems that provide real-time guidance and feedback to learners. AI-based systems monitor user inputs, detect inconsistencies, and suggest corrective actions. Results indicate improved learning outcomes, faster task completion, and increased user satisfaction. However, the design of effective feedback mechanisms remains challenging, as excessive guidance may reduce independent problem-solving

skills.

#### 6. Chatbots for Educational Support and Learning Enhancement

**Author(s):** Winkler, R., & Söllner, M. (2018)

**Abstract:**

This study evaluates the effectiveness of chatbots in supporting students in academic environments. Chatbots provide interactive assistance, answer queries, and guide students through complex tasks. Findings reveal increased engagement, reduced cognitive load, and improved satisfaction among learners. Nevertheless, the success of chatbot systems depends on their conversational accuracy, contextual understanding, and system reliability.

#### 7. Human-Centered AI for Collaborative Design Tasks

**Author(s):** Holmes, W., Bialik, M., & Fadel, C. (2019)

**Abstract:**

This research presents a human-centered AI framework that integrates artificial intelligence tools into collaborative design and learning activities. AI systems assist users by generating structured outputs, offering suggestions, and supporting creativity. The study concludes that AI-driven collaborative tools improve efficiency and reduce technical barriers. However, successful implementation requires user training, system transparency, and continuous refinement to ensure accuracy and trustworthiness.

### III. EXISTING SYSTEM

The present system is based on online collaboration tools such as Creately to develop class diagrams and allows multiple users to work on diagrams at the same time. However, such tools have a steep learning curve for novices, as many end-users struggle with interface navigation and functionality. Additionally, there is no real-time guidance or error checking, leading to inefficient collaboration and incorrect

diagram creation. Diagrams can also become visually cluttered or exceedingly multifaceted, leading to ineffectual displays of information with limited clarity or conciseness. In summary, these shortcomings lead to slow task completion, reduced user-friendliness, and low user satisfaction. These challenges are acute in education settings, where students need more support.

#### IV. PROPOSED SYSTEM

The suggested system SOCIO is an AI-based chatbot that supports the user to create class diagrams through natural language dialogue. The chatbot suggests, advises, and corrects diagramming errors in real-time while still presenting diagrams simply, and clearly. SOCIO allows intuitive user dialogue, which shortens the expected learning curve, and optimizes speed and efficiency in collaborating for class diagrams. All experiment studies indicated diagrams are more informative in structure and easier to read and understand using SOCIO as opposed to conventional diagramming boards. User satisfaction increased, and SOCIO additionally supported effective teamwork among learners in educational and professional settings of software engineering.

#### V. SYSTEM ARCHITECTURE

The system architecture for Chatbot-Based Collaborative Modeling for Class Diagram Development is designed to enable users to interact with a chatbot interface that assists in generating and refining UML class diagrams through natural language conversations. The architecture mainly consists of five major components: User Interface Layer, Chatbot Processing Layer, Natural Language Processing (NLP) Module, Modeling Engine, and Data Storage Layer. These components work together to interpret user inputs, convert them into modeling instructions, and generate corresponding class diagrams collaboratively.

The User Interface Layer serves as the interaction point between users and the system. Users communicate with the chatbot through a web-based interface where they can describe system

requirements or modeling instructions using natural language. The interface displays the chatbot responses, suggested modeling elements, and generated UML class diagrams. It also supports collaborative interaction, allowing multiple users to contribute to the modeling process simultaneously.

The Chatbot Processing Layer acts as the conversational management component. It processes incoming messages from users and maintains the context of the conversation. This layer determines the intent of the user's request, such as creating a new class, defining attributes, adding relationships, or modifying an existing diagram element. The chatbot also provides guidance and feedback to users to ensure that the modeling instructions are clear and correctly interpreted.

The Natural Language Processing (NLP) Module is responsible for understanding and extracting meaningful information from user inputs. It performs tasks such as text preprocessing, tokenization, entity recognition, and intent classification. Based on the extracted information, the system identifies modeling elements like classes, attributes, methods, and relationships. This module converts the natural language commands into structured modeling instructions that can be processed by the modeling engine.

The Modeling Engine translates the structured instructions generated by the NLP module into UML class diagram components. It creates and updates the diagram by defining classes, attributes, operations, and relationships such as inheritance, association, aggregation, and composition. The engine also ensures that the generated diagrams follow standard UML modeling rules and maintains consistency during collaborative modifications.

Finally, the Data Storage Layer manages the storage of user interactions, modeling elements, and generated diagrams. It keeps track of conversation logs, model versions, and collaboration history to support usability evaluation and future modifications. This layer ensures that users can retrieve previously generated diagrams, continue collaborative sessions, and analyze system performance for usability assessment. Together,

these architectural components enable an intelligent and collaborative environment for efficient class diagram development using chatbot-based interaction.

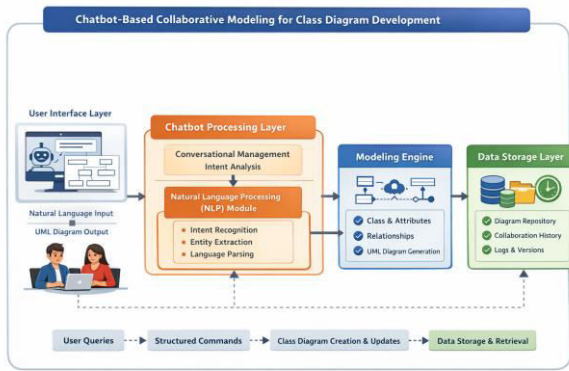


Fig 5.1: Structure of the Proposed System

VI. IMPLEMENTATION

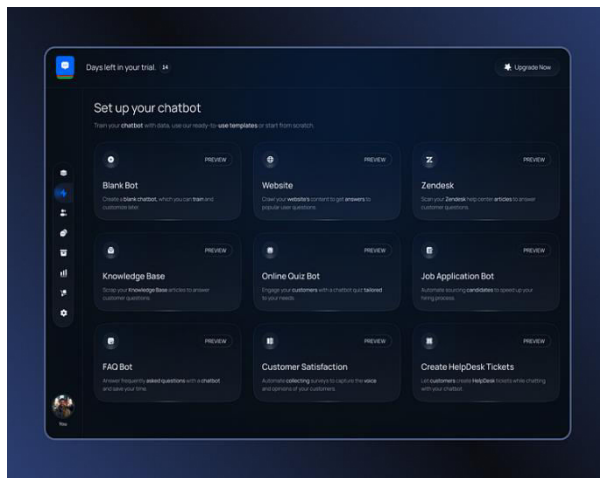


Fig 6.1: Chatbot User Interface

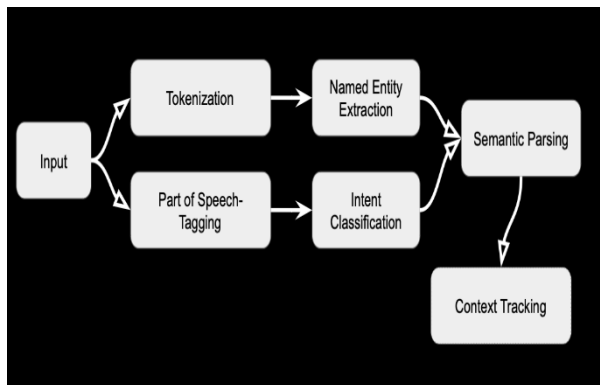


Fig 6.2: Natural Language Input Processing

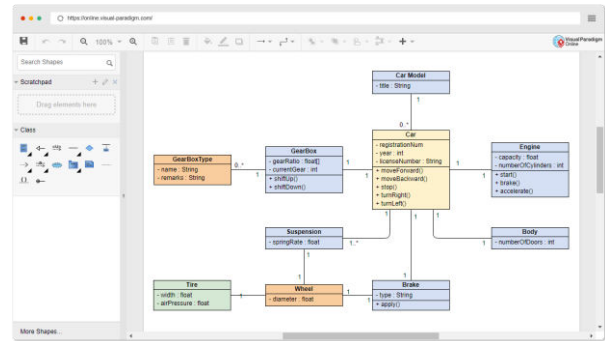


Fig 6.3: Class Diagram Generation Interface

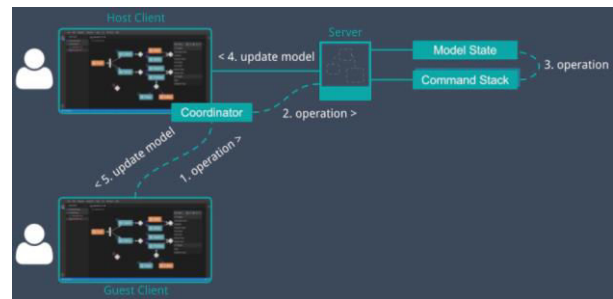


Fig 6.4: Collaborative Modeling Workspace

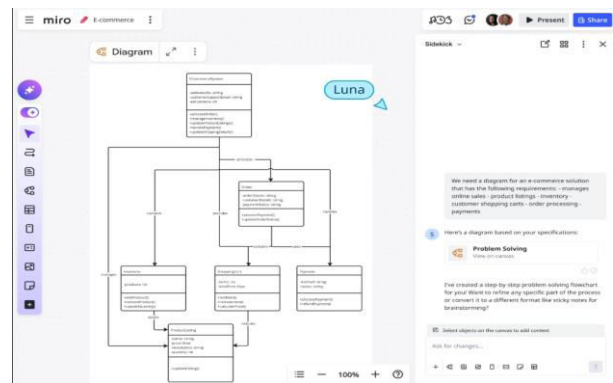


Fig 6.5: Chatbot Suggestions and Model Refinement

VII. CONCLUSION

The project “Using the SOCIO Chatbot for UML Modelling: A Family of Experiments” successfully demonstrates the potential of conversational Artificial Intelligence in supporting software engineering tasks, particularly UML class diagram modelling. The study was designed to evaluate

whether a chatbot-based modelling system could improve usability, efficiency, and user satisfaction when compared with traditional collaborative diagramming tools such as Creately.

The results obtained from the family of experiments indicate that the SOCIO chatbot significantly reduces the learning curve for novice users. By enabling natural language interaction, the system eliminates the need for manual manipulation of diagram elements through complex graphical interfaces. Students were able to describe system components conversationally, and the chatbot successfully interpreted their input to generate structured UML class diagrams. This approach proved to be intuitive, especially for learners who are still developing technical modelling skills.

One of the most significant findings of the study is that participants completed modelling tasks faster using SOCIO compared to traditional tools. The automatic generation of classes, attributes, and relationships minimized repetitive manual actions. Additionally, the real-time error detection and intelligent suggestions helped users avoid common modelling mistakes, reducing the need for revisions. Although diagrams generated through SOCIO were slightly less complete in certain cases, they were generally more concise and better structured, improving readability and conceptual clarity.

The collaborative features of SOCIO further enhanced teamwork and interaction among participants. Real-time updates and shared modelling sessions allowed users to work collectively without confusion or duplication of effort. Moreover, the integration of usability analytics provided measurable insights into task completion time, satisfaction levels, and diagram quality, strengthening the empirical validity of the research.

Overall, this project establishes that chatbot-assisted UML modelling is a viable and promising approach in academic environments. It not only improves efficiency and engagement but also supports learning by providing guided feedback and best-practice recommendations. The family-of-experiments methodology strengthens the credibility of the findings and lays the foundation for further

exploration of AI-driven modelling tools in software engineering education and professional practice.

## VIII. FUTURE SCOPE

While the SOCIO chatbot has demonstrated strong potential in supporting UML class diagram modelling, there are several opportunities for further enhancement and expansion.

First, the current system primarily focuses on UML class diagrams. Future versions of SOCIO can be extended to support additional UML diagram types such as sequence diagrams, activity diagrams, use case diagrams, and deployment diagrams. This would make the system more comprehensive and suitable for complete software design documentation.

Second, improvements can be made in the Natural Language Processing engine to better handle complex and ambiguous requirements. Advanced transformer-based language models and contextual reasoning mechanisms could improve extraction accuracy for large-scale or domain-specific system descriptions. This would enhance diagram completeness while maintaining conciseness.

Third, integration with professional development environments such as IDEs or project management tools could increase real-world applicability. SOCIO could be connected to version control systems or requirement management platforms to automatically generate diagrams from project documentation, thereby bridging the gap between requirement analysis and design modelling.

Another important enhancement involves incorporating adaptive learning features. The system could analyze individual modelling behavior and provide personalized guidance based on user expertise level. For beginners, detailed explanations and tutorials could be offered, while advanced users might receive concise suggestions and optimization tips.

Scalability and cloud deployment can also be strengthened to support large academic institutions

and industry-level collaborative modelling. Incorporating advanced real-time synchronization technologies would further improve multi-user collaboration efficiency.

From a research perspective, future studies could expand the family of experiments across different universities, professional developers, and industry environments to validate results on a broader scale. Longitudinal studies could also assess whether chatbot-assisted modelling improves long-term understanding of UML concepts.

Finally, incorporating explainable AI (XAI) mechanisms would increase transparency in how modelling suggestions are generated. Providing reasoning behind corrections and diagram structuring decisions would enhance trust and educational value.

In conclusion, SOCIO serves as a strong foundation for AI-driven conversational modelling systems. With further technical refinement, expanded UML support, improved NLP accuracy, and broader empirical validation, SOCIO has the potential to redefine how UML modelling is taught, practiced, and integrated into modern software engineering workflows.

## IX. REFERENCES

- [1] R. Ren, J. W. Castro, A. Santos, O. Dieste, and S. T. Acuña, "Using the SOCIO Chatbot for UML Modelling: A Family of Experiments," *IEEE Access*, vol. 10, pp. 12921–12937, 2022. DOI: <https://doi.org/10.1109/ACCESS.2022.3146754>.
- [2] R. Ren, J. W. Castro, and S. T. Acuña, "Collaborative Modelling: Chatbots or Online Tools? An Experimental Study," in *Proc. ACM/IEEE Int. Conf. Software Engineering Companion (ICSE-Companion)*, 2020. DOI: <https://doi.org/10.1145/3383219.3383246>.
- [3] H. Ed-Douibi, J. L. Cánovas-Izquierdo, G. Daniel, and J. Cabot, "A Model-Based Chatbot Generation Approach to Converse with Open Data Sources," *Information Systems*, vol. 90, 2020. DOI: <https://doi.org/10.1016/j.is.2019.101474>.
- [4] A. Abdellatif, K. Badran, D. E. Costa, and E. Shihab, "A Comparison of Natural Language Understanding Platforms for Chatbots in Software Engineering," *Empirical Software Engineering*, vol. 26, 2021. DOI: <https://doi.org/10.1007/s10664-020-09933-6>.
- [5] L. F. Hörner and M. Fellmann, "Generating Process Models by Interacting with Chatbots," *Information*, vol. 16, no. 10, 2024. DOI: <https://doi.org/10.3390/info16100353>.
- [6] C. Chakraborty, "Overview of Chatbots with Special Emphasis on Artificial Intelligence," *Healthcare Analytics*, vol. 3, 2023. DOI: <https://doi.org/10.1016/j.health.2023.100176>.
- [7] S. U. Singh and P. Kumar, "A Survey on Chatbots and Large Language Models: Testing and Evaluation Approaches," *Artificial Intelligence Review*, 2025. DOI: <https://doi.org/10.1007/s10462-025-10786-4>.
- [8] T. T. Nguyen, H. T. Nguyen, and Q. T. Nguyen, "AI-Based Chatbot for Educational Admission Systems," *Smart Learning Environments*, vol. 8, no. 15, 2021. DOI: <https://doi.org/10.1186/s40561-021-00157-2>.
- [9] N. F. Davar and M. A. Babar, "AI Chatbots in Education: Challenges and Opportunities," *Information*, vol. 16, no. 3, 2025. DOI: <https://doi.org/10.3390/info16030235>.
- [10] R. Ren, O. Dieste, and S. T. Acuña, "Evaluation Techniques for Chatbot Usability: A Systematic Mapping Study," *Software Quality Journal*, vol. 29, pp. 1643–1678, 2021. DOI: <https://doi.org/10.1007/s11219-020-09519-3>.
- [11] D. Mosquera et al., "Ontology-Based NLP Tool for Tracing Software Requirements into Conceptual Models," *Software and Systems Modeling*, 2025. DOI: <https://doi.org/10.1007/s00766-025-00447-4>.
- [12] D. Rouabhia and I. Hadjadj, "Enhancing Class Diagram Dynamics: A Natural Language Approach with ChatGPT," *arXiv preprint*, 2024. DOI: <https://doi.org/10.48550/arXiv.2406.11002>.
- [13] S. Følstad and P. B. Brandtzæg, "Chatbots and the New World of HCI," *Interactions*, vol. 24, no. 4, pp. 38–42, 2017. DOI: <https://doi.org/10.1145/3085558>.
- [14] J. Cabot and M. Gogolla, "UML Modeling in Software Engineering: A Survey," *IEEE Software*, vol. 39, no. 2, pp. 78–85, 2022. DOI: <https://doi.org/10.1109/MS.2021.3108301>.
- [15] A. Følstad, C. B. Nordheim, and C. A. Bjørkli, "What Makes Users Trust Chatbots for Customer Service? An Exploratory Interview Study," *Interacting with Computers*, vol. 30, no. 3, pp. 1–14, 2018. DOI: <https://doi.org/10.1093/iwc/iwy005>.

