Documentation Artifacts For Conversation-Related Requirements Specification in Chatbots

Larissa Pereira Gonçalves[0000−0002−3355−9527], Geovana Ramos Sousa Silva[0000−0002−0304−0804], and Edna Dias Canedo[0000−0002−2159−339X]

University of Brasilia (UnB), Brasilia – DF, Brazil
`larissa.goncalves@redes.unb.br`, `geovannna.1998@gmail.com`, `ednacanedo@unb.br`

Abstract. Context: Chatbots are complex applications due to their capacity to engage and maintain a conversation with humans. However, the conversational-related requirements of chatbots are hard to elicit, document, and test. Another challenge is the documentation since there are not so many directions on how to register and test subjective requirements. Methods: We followed systematic literature review (SLR) guidelines and identified 42 relevant papers that address the artifacts used by practitioners to document conversational-related requirements in literature. We also investigated what conversational requirements are addressed in requirements documentation. Results: The main results indicate that UML diagrams, prototypes, tables of requirements, conversational flows, and scenarios are present in most chatbot documentation. Except for UML diagrams, those artifacts are used to document standard requirements or conversational requirements. In those artifacts, context-dependent behavior, assertivity, error handling, and human-like attitude are the most approached conversational requirements in the studies. In sequence, based on our findings, we propose the conversational integrated map, a meta-model solution as documentation of conversational requirements.

Keywords: Chatbots conversational agents · specification conversational requirements · conversational integrated map.

1 Introduction

Unlike other types of software, the main feature of a chatbot is good conversation. While in most applications there are buttons and screen elements to interact, the most valuable deliveries for chatbot applications are conversational flows that meet their purpose, whether to answer a question, complete a task, or entertain the user [18].

Mimicrying human conversation differs chatbots from other applications, as they require attention to emotional and conversation approach [8]. User experience focus is making applications more friendly by displaying specific items on

Proceedings of the 27th Workshop on Requirements Engineering (WER24), August 07-09, 2024, Buenos Aires, Argentina.
DOI: 10.29327/1407529.27-19

screens, hierarchical organization, prevention of user mistakes, and on-time interactions [15]. Chatbots can present these tools but work mainly on user input and answers, which turns the conversation itself focus on eliciting requirements and quality assurance.

Software requirements are the basis for the development and tests of the applications, making Requirements Engineering (RE) a critical component in system development [17]. AI-based chatbots, particularly those also composed of machine learning, present particular data-related challenges in requirements engineering, as they heavily rely on data integrity, availability, and compliance [13], turning software requirements engineering even more critical for them.

Those requirements are registered in software documentation, which increases development speed and facilitates the communication between practitioners and customers [6]. However, there is a trade-off between the number of details presented in the documentation and the ease of further updates in the generated artifacts [2]. For chatbots, this situation represents a triple issue: First, the requirements are challenging, since they can be data-driven [13] and often subjective [18]; Second, there is no consensus on how would be the best form to document the subjective requirements that arise from this type of application, so practitioners often adapt existent software artifacts [18]; Third, subjective requirements are difficult to document and test, as the metrics for success or failure are hard to be established [18], athough attemps are being conducted to analyse requirements' ambiguity after documented [12].

In this paper, we have conducted a systematic literature review to focus on the software documentation produced when designing and developing a chatbot application. The main results indicate that UML diagrams, prototypes, tables of requirements, conversational flows, and scenarios are present in most chatbot documentation. Except for UML diagrams, those artifacts are used to document standard requirements or conversational requirements. In those artifacts, context-dependent behavior, assertivity, error handling, human-like attitude, feedback, and proactivity are respectively the most addressed conversational requirements. Based on the results of SLR, we created a metamodel comprising the main features of most popular artifacts to register conversational requirements. Our next goal is to evaluate it with professionals involved in chatbot projects.

2 Background and Related

According to Habib et. al [7] there are six (6) components of a software documentation process that make up documentation considered light and at the same time sufficient, namely: 1) Design document; 2) Project Overview; 3) Requirements document; 4) The support document; 5) Operation document; 6) Product documentation; and 7) System documentation. Software documentations are an agreement between the customer and the developer [7]. Artifacts produced during development may or may not have more than one owner, and generally adopt version control to accommodate changes that arise throughout the project [19].

Chatbots are diverse and usually can be classified in more than one type at the same time [1]. According to Adamopoulou and Moussiades [1], categories of chatbots can comprise domain coverage, response generation method, input or output of response, platform, and others. Schon et. al [17] conducted a broad SLR on agile artifacts. They reported User Stories as being the most common artifact used in agile software development (ASD), followed by Prototypes, Use cases, Scenarios, and Story cards. It was reported that although User Stories are the most used artifact, it is not suitable to document non-functional requirements [17]. Use cases, although, showed to be more suited to address non-functional requirements since they can be highly adaptable [14].

Silva and Canedo [18] conducted a survey with practitioners and addressed several challenges in eliciting or documenting chatbots' requirements, such as distance from stakeholders' needs, poorly documented requirements and lack of specific tools for requirements engineering. Conversational flows and user stories were the most common documentation techniques used by the respondents [18]. Despite these works, documentation and specification are still source of problems in requirements engineering [5], and human elicitation and documentation for chatbots are even newer in literature.

3 Methodology

Our study consisted of two steps. First, we conducted a systematic literature review to identify documentation artifacts addressing chatbot requirements, focusing on conversation-related ones. Next, we used our SLR findings to propose a meta-model documentation, based on perceived needs addressed in literature.

SLR: For literature review, we followed Systematic Literature Review guidelines by Kitchenham [11] to conduct the research. First, we defined our research question, which will guide the rest of the work. Then we chose in which databases the search was to be applied. Some databases needed adaptations of the search string due to specific query parameters. When the search was done, the duplicated studies were excluded and we established inclusion and exclusion criteria for the evaluation of the remaining papers. A quality assessment was applied, and the approved papers were ready for analysis. To address the specific subject of this research, the following research question was made:
RQ 1: "How are chatbot's conversational requirements documented, and which conversational aspects are present in the requirements documentation?".

The Framework chosen to create the textual search sequence was PICOC [16]. Initial search string was constructed from PICOC table (Supplementary Material `https://zenodo.org/records/11187626`, file "Table - PICOC terms used in systematic review.pdf") and the final sequence was defined as follows: ***("chatbot" OR "conversation interface" OR "conversational agent" OR "conversational interface") AND ("documentation" OR "elicitation" OR "requirement" OR "specification") AND ("artifact" OR "document" OR "format" OR "framework" OR "guideline" OR "prac-***

tice" OR "recommendation" OR "technique" OR "toolkit") AND (NOT "chatbots for requirement" OR NOT "chatbot to elicit")

Following Kitchenham methodology, 5 databases were chosen to perform this research [11]: ACM, IEEEXplore, SCOPUS, Springer Link e Web of Science. The initial search string was adapted to each one of the bases, and the searching process was made by the title, keywords, and abstract of the papers retrieved. The digital databases were chosen due to their relevance in software engineering research [3], the scope in indexing conferences and newspapers, and the possibility of applying the generic search string directly in the search field.

Aligned with Kitchenham's proposal [9], inclusion and exclusion criteria were defined. Inclusion criteria were: (IC 1) Studies must present information related to conversational requirements in chatbots; (IC 2) The study presents artifacts containing requirements specifications in chatbots; and (IC 3) The study presents empirical analyses (case studies, experiments, research) or theoretical (systematic reviews, conceptual analyses) on documentation of chatbot requirements.

The following exclusion criteria aimed to exclude works that did not meet the expected characteristics: (EC 1) Studies that are not related to the documentation of chatbot requirements or that do not directly address the research questions mentioned above; (EC 2) Studies that are written in a language other than that understood by the authors (Portuguese, Spanish and English); (EC 3) The focus of the study is not the analysis of documentation artifacts for chatbots, but other procedures (i.e.: automation, classification with NLP, application of chatbots for elicitation or documentation of general requirements, elicitation techniques, etc.); (EC 4) The text does not follow the format of a scientific study (opinion papers, books, editorials, technical reports, theses, dissertations, conference reports, presentation summaries, or posters); and (EC 5) The main subject of the paper is not related to software engineering or related areas, but to other areas of research.

To carry out this review, the Parsifal tool was used. The open-source web platform was chosen due to the suitability of its functionalities and work structure to the SLR process adopted in this study, as proposed by Kitchenham and Charters [10]. This tool facilitates the review, covering all phases of the systematic review (planning, conducting, and reporting),with automatic identification of duplicate papers.

The study collection period did not filter from a specific beginning year and was made until January 2024, resulting in a total of 933 papers (18 from ACM, 42 from IEEE Xplore, 409 from SCOPUS, 416 from Springer Link and 48 from Web of Science). 137 duplicate studies were removed, leaving 796 studies (9 from ACM, 28 from IEEE Xplore, 348 from SCOPUS, 378 from Springer Link and 33 from Web of Science) to apply the inclusion and exclusion criteria, reading the title and abstract. Although the studies analyzed by title and abstract addressed the topic of requirements, 649 were removed due to non-compliance with one or more criteria. Finally, the full text was read for more in-depth analysis and application of the quality assessment of 147 studies (5 from ACM, 7 from IEEE Xplore, 58 from SCOPUS, 61 from Springer Link and 16 from Web of Science), which resulted in the removal of 105 works and the final number of 42 selected review studies (1 from ACM, 2 from IEEE Xplore, 22 from SCOPUS, 10 from Springer Link and 7 from Web of Science).

4 Results

In the selected studies, 26 types of artifacts were found, and 15 of them were present in more than one work. Most studies addressing chatbot artifacts and conversational requirements were published between 2019 and 2023. All artifacts produced in our study, such as the statistical analysis overviews and model templates are available on Zenodo at `https://zenodo.org/records/11187626`.

We attempted to maintain the same names that were written in the studies, however, similar terms such as "dialog flows", "conversational streams" and "conversational flows" have been grouped under the same category "conversational flows", for example. Among these, UML diagrams, prototypes, tables of requirements, meta-requirements or design principles, conversational flows, and scenarios, were respectively the most recurring artifacts (Figure 1). Furthermore, other diagrams (that did not follow UML standards), personas, user stories, workflows, database information, decision trees, issues, knowledge graphs, timelines, and use cases have also been mentioned in more than one study. Although they are not cited as much as the first five, they may indicate different documentation in certain contexts.

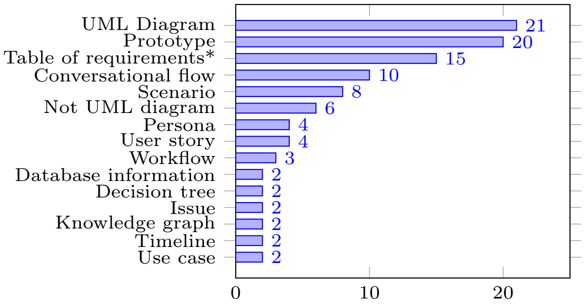

Fig. 1. Number of artifacts found for chatbot's requirements documentation.

Among the artifacts mentioned only once are ontology, chatbot's usability questionnaire, design and implementation checklist, video, mind map, empathy map, document analysis, template and meta-model of design decisions, result tree, business process maps, and story file.

Beyond quantification, we also grouped the available data by the combination of the most frequent attributes of chatbots addressed in the studies, as seen in Table 1. Most addressed chatbots' classifications were related to input/output (text-based or speech-based), domain knowledge (specific or broad), and response generation (rule-based, generative, hybrid, or task-based). Some studies also addressed the platform in which the chatbot operates (web, mobile, or inside an existing app).

Table 1. Groups of chatbots found in the selected studies.

Knowledge Domain	Response generation	Input Output	Cited
Domain-specific	Rule-based	Text	22
Domain-specific	Generative (AI)	Text	19
Domain-specific	Rule-based	Hybrid	14
Domain-specific	Rule-based	Speech	13
Broad domain	Rule-based	Hybrid	4
Domain-specific	Task-based	Speech	4
Domain-specific	Generative (AI)	Hybrid	2
Domain-specific	Hybrid	Text	2

The most common types of chatbots were domain-specific knowledge, with rule-based response generation. Input and output varied from text to speech or hybrid (Table 1). Most studies were conference or research papers, so a restricted domain of knowledge and a less robust response generation were expected in the early stages of research. The distribution of artifacts for the most frequent combinations of types showed a similar pattern to the results found, with the exception of predominance from UML diagrams and prototypes, that changed between them. The remaining combinations did not follow the pattern but showed at least one of the first six frequent artifacts, as shown in Table 1.

From the artifacts found, we analyzed each one to find the conversational requirements addressed. For “conversational requirements", we considered those that appeared directly to the user in the moment of conversation, such as voice tone, conditional answers, presence of textual and non-textual elements (such as emojis). It was noted that some types of artifacts are widely used for chatbot software documentation, but do not frequently address conversational requirements, such as UML diagrams (Figure 2). They are present in most chatbot documentation (Table 1), but their content is variable, such as software architecture or relationships between inner components. Except for UML diagrams, the address of conversational requirements by the artifacts commonly followed the pattern from Figure 1.

The analysis of artifacts content revealed a variation of conversational requirements addressed, as shown in Figure 3. Most artifacts are used to address the differential answers or actions the chatbot should take while handling different situations. Since most chatbots perform tasks, beyond the activity of speaking to the user itself, this requirement is responsible for ensuring the chatbot can manage to handle a variety of situations. Assertivity, simplified answers, and error handling, which are other conversational-related requirements in documentation, are also applied to chatbots' performance while speaking or performing tasks.

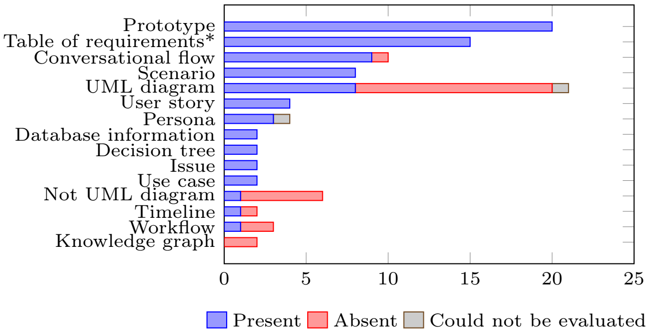

Fig. 2. Presence of conversational requirements in each addressed chatbot artifact.

Feedback is partially related, since users may need to report the overall quality of the answer.

Subjective aspects such as proactivity, flexibility, social cues, adaptability, emotional awareness, and empathy were also addressed, but in a minor number (Figure 3). Visual appearance was also a concern in some of the studies, especially regarding the vision of users when entering an app or speaking with the chatbot for the first time. Security-related issues were also addressed in a dispersed way, as articles addressed security related to user data, trust, privacy, or the transparency of the conversational agent.

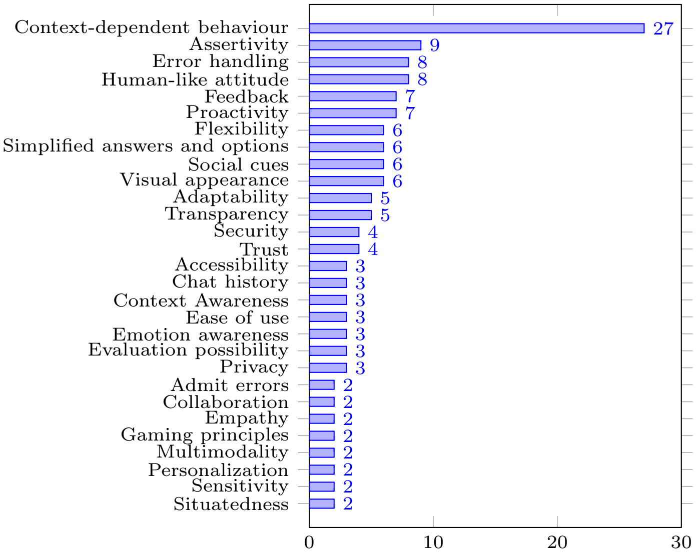

Fig. 3. Conversational requirements found in chatbot documentation artifacts.

Based on the SLR findings above, we propose a meta-model to register chatbot requirements. The meta-model is guided by the most popular artifacts for the direct address of conversational requirements in chatbots (Figure 2) - prototypes, table of requirements, conversational flows, scenarios, and diagrams - we developed a template that can be adapted to the reality of practitioners.The metamodel does not intend to replace all artifacts, but to reduce multiple documents for chatbot requirements documentation while being flexible, according to the practitioners' needs; It also must connect requirements and conversational elements, represent visual appearance and show a sequence of interactions between the chatbot and the user.

Table 2. Conversational artifact meta-model composition.

Component	**Mandatory or optional**	**Description**
Title	Mandatory	Presents the specific situation to be specified.
Epic	Optional	If the scenario is part of a larger situation, it addresses the father scenario.
Scenario description	Mandatory	Describe a summary of the situation presented. Can contain information that the practitioner finds important, such as preconditions.
Sequence number	Optional	Shows the timeline of interaction and change in chatbot and user turns. It's recommended if needed to cite a previous message.
Chatbot message	Mandatory	Shows the chatbot's response or interaction.
User message	Mandatory	Shows the user's response or interaction.
Requirement	Mandatory	Explanation of the requirement.
KPI or acceptance criteria	Optional	Explanation of how the requirement is considered attended.
User annotation	Mandatory	Highlights of user response.

To address the premises above, we defined the parts that the model should contain. They include a title, an epic (when this scenario is part of a bigger situation), a case description, a sequence of turns, and the content of messages between the chatbot and the user, along with the description of requirements attached to each message (Table 2). Additionally, we decided to add an optional field for KPIs [4] (or acceptance criteria), linked to each requirement, due to the common subjectivity of conversational requirements [18]. The model template can be adapted in a table format (Table 3), prototype format, and diagram format (Supplementary Material `https://zenodo.org/records/11187626`, files "Conversational integrated map - prototype.png" and "Conversational integrated map - diagram.png").

Table representation can be done in any software that manipulates sheets or text, being the most accessible presentation of artifacts. In this representation, however, the provision of the visual appearance is less addressed, therefore the visual fidelity is low. However, tables offer the additional capacity of the world search inside documents and filtering and are also easier to scale if used for bigger scenarios.

We used Draw.io, an online and free software to make the diagram representation. However it can be done in any software with UML notations, and is almost as versatile as table representation. The requirements and KPIs table on the right side were adapted from UML class representations. The messages were

Table 3. Conversational integrated map model. “*" means mandatory field.

<table>
<tr><td colspan="5"><b>Title*:</b> Providing specific help desk to an already logged user</td></tr>
<tr><td colspan="5"><b>Epic:</b> Help desk
<b>Description*:</b> The user is already logged in the chatbot application. User starts the conversation by asking the chatbot the nature of the problem she is having. Chatbot answers both the problem she is experiencing and the solution for each possible problem. User feels satisfied with the response and thanks the chatbot. The chatbot answers politely and makes the user comfortable.</td></tr>
<tr><th>N</th><th>Turn*</th><th>Message Content*</th><th>Requirement*</th><th>KPI</th></tr>
<tr><td>1</td><td>User</td><td>Good morning, the question mark key on my computer is not working, what could be?</td><td>Greets; Asks a question without “?”. Asks for help.</td><td></td></tr>
<tr><td>2</td><td>Chatbot</td><td>Morning Joana, I'm sorry to hear your problem. If a question key is not working, it could be due to:
<b>Configuration</b>: Check whether the keyboard configuration in the operating system is correct;
<b>Dirt/Obstruction</b>: Try cleaning the keyboard with compressed air. If it does not work, consider calling technical assistance.</td><td><b>1.Error Handling</b>: Understand and respond user intent even though the input has one or some typos.
<b>2.Personalization</b>: Chatbot answers using the user's name.
<b>3.Empathy</b>: Chatbot recognizes the user is going through a bad moment and express its empathy.
<b>4.Proactivity</b>: Chatbot answers the question and also provides the solution
<b>5.Direct to specialist</b>: When a user asks a technical, legal or medical answer, in the end of the response there must be an advise to contact a human professional.</td><td>3. Present an issue (technical or not) and receive a polite recognition of the situation.
4. Ask a technical question and receive an answer and a solution.</td></tr>
<tr><td>3</td><td>User</td><td>Thanks a lot, it worked!</td><td>Thanks the chatbot for the help in a well succeeded task.</td><td></td></tr>
<tr><td>4</td><td>Chatbot</td><td>I'm glad to hear it! If you have any more questions or need further assistance, feel free to ask!</td><td><b>6.Show availability to the user</b>: End conversation showing to user that the chatbot is still available if she has further questions.</td><td></td></tr>
</table>

adapted from UML objects. However, simple geometrical forms can be used to compose the diagram, regardless notation.

The diagram representation (Supplementary Material `https://zenodo.org/records/11187626`, file “Conversational integrated map - diagram.png")) can be adapted and incremented with other resources, such as photos or emojis. It also gives a increased sense of visual appearance than the tables because in the left side there is the simulation of baloon messages is user-friendly. Addressal of requirements can be done by highlighting the phrases with requirement assigned color (Supplementary Material `https://zenodo.org/records/11187626`, file “Conversational integrated map - diagram.png")). This representation can be classified as medium-fidelity, and can be used to address bigger interactions, such as the tables.

For the prototype representation (Supplementary Material `https://zenodo.org/records/11187626`, file “Conversational integrated map - prototype.png"), Figma was used. The creation process was composed of choosing a device and representing a conversation as a user would see it. Each requirement receives a color and is highlighted as it appears on the interaction. Prototype representation shows similar behaviors to the diagram, but addresses visual appearance and usability requirements directly, and can also show dynamic interactions as they can be configured in the Figma file. The paid version also allows to inclusion of audiovisual resources such as videos or audio, which can be a more complete option if the chatbot is speech-based, or requires a video interaction. In general, audiovisual resources are rarely represented in software documentation (1).

Table 4. Comparision between proposed conversational integrated map and popular artifacts for chatbots' documentation.

Artifact	Visual appearance	Requirements description	Interaction sequence	Link of requirements and elements
Conversational integrated map	Yes	Yes	Yes	Yes
Conversational flow	No	No	Yes	No
Decision tree	No	No	Yes	No
Scenario	No	No	Yes	No
Prototype	Yes	No	Yes	No
UML diagram	No	No	Yes	No
Requirements table	No	Yes	No	No

Compared to some popular artifacts for chatbot's specification, the conversational integrated maps appear as a versatile and robust approach (Table 2). It comprises the visual appearance offered by the prototypes (Supplementary Material `https://zenodo.org/records/11187626`, file "Conversational integrated map - prototype.png"), the detailed requirement description found in tables of requirements (Table 3), and the global sequenced visualization offered by UML diagrams, scenarios or mind maps (Supplementary Material `https://zenodo.org/records/11187626`, file "Conversational integrated map - diagram.png")).

5 Discussion

SLR findings: In sum, we have found 42 relevant studies according to our previously described criteria. The articles described documentation artifacts and also addressed conversational requirements, which were the focus of our work. Next, we will discuss on the findings of our SLR. Overall, SLR results show that requirements documentation for chatbots have been addressed in literature, despite not being the focus of most articles. We can conclude that this is an important and recent research topic of interest since our analysis showed that the majority of the studies were published between 2019 and 2023 (Supplementary Material `https://zenodo.org/records/11187626`, file "Graphic - Papers per year of publication.pdf"). It is also possible to note that most used artifacts for chatbots are common to software development in general. This resembles the findings of [18], which conducted a survey with practitioners of chatbot development. Conversational flows and tree structures have been addressed regularly and are not so common in other software areas.

A new finding of our work is that tables of requirements, meta-requirements, and design principles, which are also not so common in software documentation, are widely used for chatbots. This can be a difference between literature and the practice itself, as this pattern was not found in related works. Since they are somehow abstract, this finding can relate to the complaints mentioned by [18], as practitioners found it difficult to develop a chatbot since users and stakeholders requirements are difficult to develop and test, and can also be subjective.

The subjectiveness could also explain the presence of a prototype as one of the most used documentation artifacts. Prototypes are flexible artifacts that can address visual appearance 3, and also verbal or non-verbal expected behaviors, such as social cues or empathy, making an idea less abstract.

Another relevant topic in our review is the presence of rule-based, domain-specific, and text-input chatbots in literature. This can be contradictory to the present interest of the software community on AI chatbots. However, AI chatbots can be expensive to implement, especially when an idea is only in its beginning. Rule-based chatbots can also be easier to test, as they are more restrictive. Another possible explanation can be that the present focus on AI chatbots is not heavily related to documentation, but data or privacy-related concerns, or analysis about existing commercial chatbots.

Predominance of requirements such as “Context-dependent Behavior" and “Assertiveness" highlights the need for chatbots that not only respond in a relevant way to user inputs but are also able to anticipate needs and adapt their responses based on the context of the conversation. The ability to recognize a text or speech input is not enough anymore: Users’ expectations have been raised to dialogue with a human-like application. Furthermore, the emphasis on “Error Handling” and “Human-Like Attitude” underlines the challenge of developing chatbots that can gracefully handle ambiguities, typos, and errors inevitable in natural conversations.

Conversational integrated map: The proposed model comprises the main features from key artifacts. Title and epic come from user stories, which are popular documentation artifacts for other types of software. They help to narrow large and unspecific requests, especially subjective non-functional requirements. The description below is derived from scenarios and helps to understand the general picture of the situation. The associated requirements are the main contribution of this model (Table 4), since current artifacts do not directly address this link.

Requirements can present or not a KPI, as it is an optional field (Table 2). However, we recommend that each requirement has one or more KPIs. In user stories, they would behave as acceptance criteria. Our KPI proposal is similar to Nguyen et. al [14] work, which adapted used cases for non-functional requirements [14]. The main reason for this proposal briefly explained above, is to enhance the importance of established measures of quality for subjective requirements, as they pose a challenge on testing and success or failure are difficult to address [18].

Our meta-model currently does not include a specific field to show architectural components or entities such as intents and entities. They were not included intentionally because the meta-model is designed to be a user-friendly document and enhance collaboration between stakeholders, and development and testing teams. However, this meta-model should be evaluated by practitioners to address if it is adherent to its purpose or if the features should be changed. Another improvement that can be implemented in future works is the hierarchical division of scenarios inside epics and bigger contexts, similar to the visualization of ontology, mind maps, or decision tree visualizations, enhancing dynamic organization.

5.1 Limitations and Threats to Validity

SLR Findings: The study was conducted following a consolidated methodology, however, there are some external and internal limitations to address. *External:* First, there is no way we can guarantee the quality of the studies used. We addressed this risk through a rigorous process for deciding on databases included and also applying inclusion and exclusion criteria, along with quality assessment. Second, it is important to highlight the issue of the nomenclature used in the articles analyzed. Vocabulary and specific terms may vary significantly between different fields of study or may have evolved over time. Another relevant consideration is the potential discrepancy between the descriptions and discussions found in academic literature and actual practices in the development and implementation of chatbots. Literature may not fully capture the complexities and challenges faced by professionals in the field, which suggests the need to complement bibliographic studies with empirical investigations.

Internal: The classification methodology adopted in this work may present challenges, especially with regard to the representativeness of the classification groups. In particular, categories that contain a smaller number of articles may not provide a reliable representation of the literature or current trends in the area of chatbots, raising questions about the veracity of the results found. Such categories may reflect less explored areas or specific niches in the specific context of chatbot requirements documentation. Finally, the validity of this study may be impacted by the fact that data analysis and cataloging were conducted by a single author. To mitigate these concerns, the analysis process was repeated three times, seeking to ensure the accuracy of the results and the robustness of the conclusions.

Proposed model: The documentation model proposed in this work has been based in the research conducted before but poses some internal and external challenges. *Internal:* First, variations of representation tend to emphasize differently the fields in the model. For example, in tables, the message sender (user or chatbot) is not so visible as in the diagram or prototype visualization. The model is also recommended only for short scenarios and conversations, so far. Longer interactions with such an amount of information can bring a high cognitive load when visualizing. *External:* The meta-model is still to be evaluated among practitioners involved in the creation and development of chatbot projects, in order to access if the proposed solution meets the needs of software professionals. It should also be evaluated if the documentation is understandable by stakeholders [17] and if it can be used as official documentation for chatbot development.Another concern regarding the usability of the meta-model is that currently, it presents adaptations from previous existing structures. Especially for prototypes, the creation process can be really manual, since there's the need to highlight specific points inside the conversation and present them outside.

6 Conclusions

In the selected studies, we found that UML diagrams, prototypes, tables of requirements, conversational flows, and scenarios are present in most chatbot documentation. However, the UML diagrams do not always address conversational requirements, as they are more commonly associated with software architecture demonstration. Most artifacts, despite differences in nomenclature, addressed the conversational requirements of context-dependent behavior and responses, assertive, error handling, human-like attitude, proactivity, social cues, simplified answers, and visual appearance. That reveals an expectation pattern for anthropomorphized chatbots by users.

The conversational integrated map presented is a flexible meta-model designed to address conversational requirements along showing user and chatbot interactions. It allows the direct link between the conversation itself, requirements, and KPIs that should be tested. We believe the meta-model will help practitioners to easily integrate conversational flows and requirements presentation, in order to avoid multiple documents for the same situation. This is the first work to review literature seeking patterns between artifacts and conversational requirements for chatbots. Additionally, we present an adaptation of the artifact for chatbot conversational-related requirements documentation. The next step of our work is the validation of this model with software professionals. However, future studies should investigate if this model and existing artifacts are enough to address all complex aspects of chatbots' specific conversational requirements or require further adaptation. Another remaining question would be the prioritization and testing of subjective requirements, an additional challenge posed by chatbot applications.

References

1. Adamopoulou, E., Moussiades, L.: Chatbots: History, technology, and applications. Machine Learning with Applications **2**, 100006 (2020), `https://www.sciencedirect.com/science/article/pii/S2666827020300062`
2. Bjarnason, E., Unterkalmsteiner, M., Borg, M., Engström, E.: A multi-case study of agile requirements engineering and the use of test cases as requirements. Inf. Softw. Technol. **77**, 61–79 (2016), `https://doi.org/10.1016/j.infsof.2016.03.008`
3. Brereton, P., Kitchenham, B.A., Budgen, D., Turner, M., Khalil, M.: Lessons from applying the systematic literature review process within the software engineering domain. J. Syst. Softw. **80**(4), 571–583 (2007), `https://doi.org/10.1016/j.jss.2006.07.009`
4. Dragicevic, S., Celar, S., Novak, L.: Use of method for elicitation, documentation, and validation of software user requirements (medov) in agile software development projects. In: Al-Dabass, D., Ameti, V., Skenderi, F., Halili, F. (eds.) Sixth International Conference on Computational Intelligence, Communication Systems and Networks, CICSyN 2014, Tetova, Macedonia, May 27-29, 2014. pp. 65–70. IEEE (2014), `https://doi.org/10.1109/CICSyN.2014.27`
5. Epifânio, J.C., Esteves, E., Lucena, M., Trindade, G.O.: Identifying knowledge gaps in requirements engineering: An empirical study with professionals in the brazilian software industry. In: WER (2023)

6. Habib, B., Romli, R.: A systematic mapping study on issues and importance of documentation in agile. In: 2021 IEEE 12th International Conference on Software Engineering and Service Science (ICSESS). pp. 198–202 (2021). https://doi.org/10.1109/ICSESS52187.2021.9522254
7. Habib, B., Romli, R., Zulkifli, M.: Identifying components existing in agile software development for achieving "light but sufficient" documentation. Journal of Engineering and Applied Science **70**(1), 75 (Jul 2023), https://doi.org/10.1186/s44147-023-00245-1
8. Hu, T., Xu, A., Liu, Z., You, Q., Guo, Y., Sinha, V., Luo, J., Akkiraju, R.: Touch your heart: A tone-aware chatbot for customer care on social media. CoRR **abs/1803.02952** (2018), http://arxiv.org/abs/1803.02952
9. Kitchenham, B.: Procedures for performing systematic reviews. Keele, UK, Keele University **33**(2004), 1–26 (2004)
10. Kitchenham, B.A., Pfleeger, S.L.: Personal Opinion Surveys, pp. 63–92. Springer London, London (2008). https://doi.org/10.1007/978-1-84800-044-5_3, https://doi.org/10.1007/978-1-84800-044-5_3
11. Kitchenham, B.A., Pretorius, R., Budgen, D., Brereton, P., Turner, M., Niazi, M., Linkman, S.G.: Systematic literature reviews in software engineering - A tertiary study. Inf. Softw. Technol. **52**(8), 792–805 (2010), https://doi.org/10.1016/j.infsof.2010.03.006
12. Kiyavitskaya, N., Zeni, N., Mich, L., Berry, D.M.: Requirements for tools for ambiguity identification and measurement in natural language requirements specifications. In: WER (2008)
13. Kostova, B., Gurses, S., Wegmann, A.: On the interplay between requirements, engineering, and artificial intelligence. In: Joint Proceedings of the 26th International Conference on Requirements Engineering: Foundation for Software Quality (REFSQ 2020), Pisa, Italy, March 24, 2020. CEUR Workshop Proceedings, vol. 2584. CEUR-WS.org (2020), https://ceur-ws.org/Vol-2584/RE4AI-paper7.pdf
14. Nguyen, Q.L.: Non-functional requirements analysis modeling for software product lines. In: ICSE Workshop on Modeling in Software Engineering, MiSE 2009, Vancouver, BC, Canada, May 17-18, 2009. pp. 56–61. IEEE Computer Society (2009), https://doi.org/10.1109/MISE.2009.5069898
15. Nielsen, J.: Ten usability heuristics. NN Group (2005), http://www.nngroup.com/articles/ten-usability-heuristics/
16. Petticrew, M., Roberts, H.: Systematic Reviews in the Social Sciences. Blackwell Publishing Ltd, Oxford, UK (2006). https://doi.org/10.1002/9780470754887
17. Schon, E., Thomaschewski, J., Escalona, M.J.: Agile requirements engineering: A systematic literature review. Comput. Stand. Interfaces **49**, 79–91 (2017), https://doi.org/10.1016/j.csi.2016.08.011
18. Silva, G.R.S., Canedo, E.D.: Requirements engineering challenges and techniques in building chatbots. In: Rocha, A.P., Steels, L., van den Herik, H.J. (eds.) Proceedings of the 14th International Conference on Agents and Artificial Intelligence, ICAART 2022, Volume 1, Online Streaming, February 3-5, 2022. pp. 180–187. SCITEPRESS (2022), https://doi.org/10.5220/0010801800003116
19. Wazlawick, R.: Engenharia de software: conceitos e práticas. Elsevier (2019), https://books.google.com.br/booksid=d1qnDwAAQBAJ&printsec=frontcover