ANALYSIS OF CHATBOT VERIFICATION METHODS IN MESSENGERS

Artur Vasyliev, Software Engineering Department,
e-mail: artur.vasyliev@nure.ua.
Supervisor: Ph.D., Prof. Igor Shubin.
Kharkiv National University of Radio Electronics

Abstract: chatbot is a type of program which can be accessed by user using a natural language. Many popular messengers have introduced their own programming interfaces which allows creating and integrating chatbots to their internal communication system. Because of this fact, chatbot engineering started to grow fast and amount of chatbots increased intensively. It caused the necessity to research and analyze properties of chatbots, distinguish the most significant quality criteria and define how to verify chatbot quality in terms of software verification.
Keywords: chatbot, verification, quality, messenger

During past three years chatbots continue to gain the popularity as one of the variants for people to interact with a software. Some IT companies realized perspectives of chatbots and many of them presented their own programming interfaces for creating chatbots. Facebook Messenger, Skype, Telegram, Slack have allowed developers to connect their own solutions to their communication systems. Developers were astonished by the fact that it became possible to develop chatbots for the most popular messengers so chatbots could potentially reach more audience because most people use messengers more frequently than other aplications. Nowadays chatbots can provide a lot of different opportunities, which depend only on business requirements for a specific chatbot. A variety of chatbot abilities include searching and processing information from the Internet, performing operations on user data or even communicating with the user as a real person by embedding Natural Language Processing. Because of these skills and simple logic of interaction with chatbots, they are used in many areas: education, medicine, e-commerce, IT-support etc. Despite the popularity and the rate of growth, chatbot engineering is quite young and developing area, and it has its defects. For the end user, the main disadvantage is a large probability that the chatbot does not match its expectations. For example, it may not respond to messages which have been written without having an appropriate format. Also, it may not understand some phrases at all, or it may send links to malicious sites or some

inappropriate content. That is where verification of chatbot quality come up.

There are many research materials related to chat bots. They could be divided into three groups which are given below.

Materials of the first group describe the work principles of such fundamental solutions as ELIZE and ALICE, which created the foundation of designing chatbots which try to speak like a human. Solutions like that have a specific architecture and they are aimed at promoting the chatbot capabilities to the level of communication to the real person.

The second group is represented by the research works that describe the complete process of creating their own stand-alone solutions, or solutions that can be connected to different messengers as chatbots. The principle of messengers combined with chatbots is described in the work of Simon Draxinger “The Generativity of Messaging Platforms: A Case Study on Facebook Messenger and Chatbots” [1].

The last group contains works that describe machine learning algorithms and NLP (Natural Language Processing), which went further AIML (Artificial Intelligence Markup Language, used to work on ELIZE and ALICE) long ago. These algorithms allow chatbots to understand user requests better and communicate with users in a more advanced form because of the usage of machine learning algorithms.

All described groups proof that there are many resources that provide information about how chatbots work and how to develop them. But there are not many researches related to evaluation of the chatbot quality and verifying it in accordance with the software quality criteria. Bayan Abu Shawar and Eric Atwell [2] analyzed evaluation of a chatbot quality as a part of The Loebner Prize Competition and compared this approach to their own method.

The main purpose of this work is to analyze the methods of chatbot verification from the point of view of software quality assurance. In this article, I will consider main criteria for the quality of chatbots, their importance and what methods and approaches are used to assess a chatbot quality. The advantages and disadvantages of existing methods will be highlighted and analyzed. Based on this information, the domain model will be described and will also be considered how to improve the verification process for chatbots. This research is currently very relevant and important because there are more and more opportunities for creating your own text assistants. The importance of the chatbot area

was noted by leading IT companies (Microsoft, Facebook), which began to provide their own solutions and platforms, but despite their and the developers' interest, the level of trust in chatbots remains low, as it is very challenging task to determine the level of product quality and security of communication with chatbots. In addition, this task should be considered from the point of view of several interested actors: developers, users and messengers.

Chatbots are determined as a type of software, which has a specific feature such as command management. The management itself is presented in the form of text written in natural language. This feature not only affects the process of interaction with the bot, but as a result transforms some stages of the process of creating this software. A diagram of the chatbot creation process is given below (Figure 1).

Obviously, the stages of the chatbot development cycle do not differ from the stages of creating any usual software. The basic stages are fundamental and the purpose of each stage is described in a vast number of resources, one of which is Richard Murch's book [3]. The diagram is different from the representation of the spiral model of the application life cycle because each stage includes some steps that are specific to the chatbot development process.

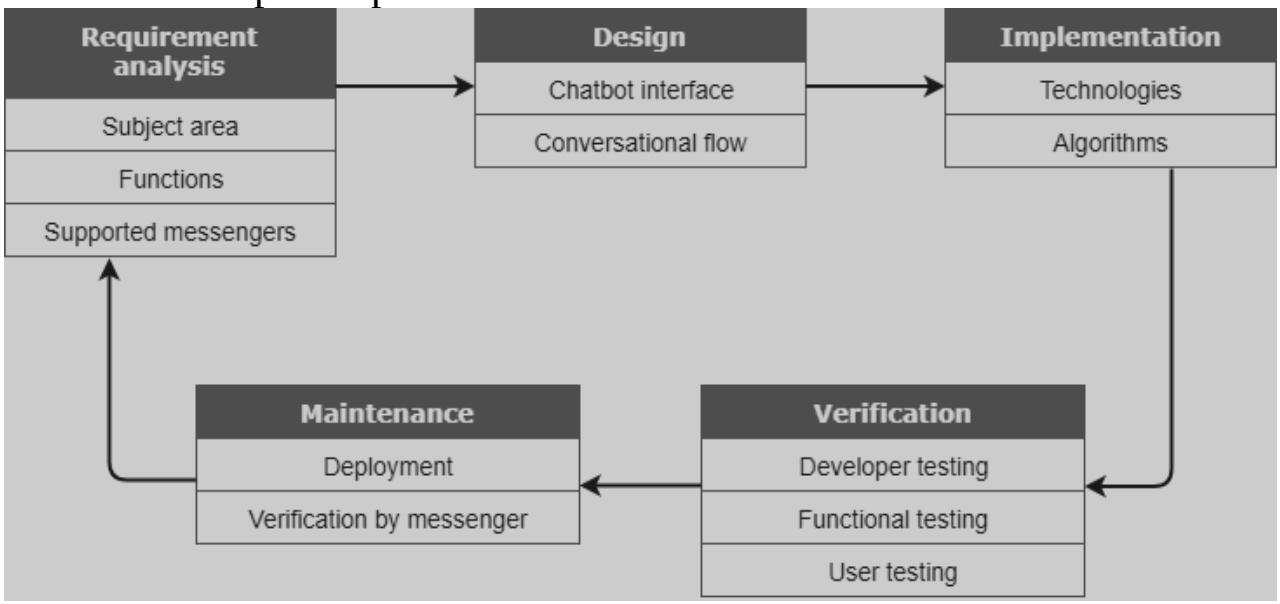

Figure 1 – Chatbot development lifecycle

The most stable and well-established stages are requirement analysis, design and implementation. Analysis of requirements includes sub-stages that relate to defining the subject area of the chatbot, its functionality and supported messengers. Selecting the subject area for

the chatbot is important, since the user interface initially assumes no boundaries (unlike mobile or desktop interfaces limited by elements of controls). Regarding chatbots text input is not a control element, but messages with the help of which the user will communicate with the bot. Since the number of possible different phrases that can be sent by the user tends to infinity value, then interaction boundaries between the catbot and the user disappears. This fact is unacceptable for most chatbots now, because not all of them can respond correctly to new questions. Therefore, to clearly delimit the scope of chatbot's responsibility and its knowledge, it is necessary to determine its subject area, which relates to the chatbot and the functions that it will perform. Also, at the requirements analysis stage it is necessary to establish which messengers will host the chatbot. Depending on the chosen messenger, some interface elements or even functionality may be changed.

The design phase includes designing a chatbot interface and a conversational flow, through which the user will communicate with the system. The chatbot interface refers to the additional controls that a messenger provides to the developer in addition to the standard text input field. For example, in Facebook Messenger and Telegram, bots often use buttons as pre-defined controls. They are most often fixed sending the most used commands so that the user could get a result with one click without typing a command every time he needs to use it. Conversational flow means the sequence of messages between a bot and a person, which ultimately leads to the bot executing the required task. Simulating conversational flow allows the developer to analyze and improve the behavior of the system at the level of communication with the user.

The implementation phase is the most accessible for understanding, since it is not much different from the implementation of any other application. The differences include stacks of technologies and algorithms that are related to Natural Language Processing, Machine Learning and Artificial Intelligence (if any of these is needed).

The most critical stages of the chatbot development lifecycle are testing and maintenance. At the testing stage, a developer intends to check both the program modules of which the chatbot consists, and whether the chatbot performs tasks that were defined in the requirements analysis phase. Also, there is a need to verify the quality of chatbot. A diagram that shows the most important quality criteria for chatbots [4] is given below (Figure 2).

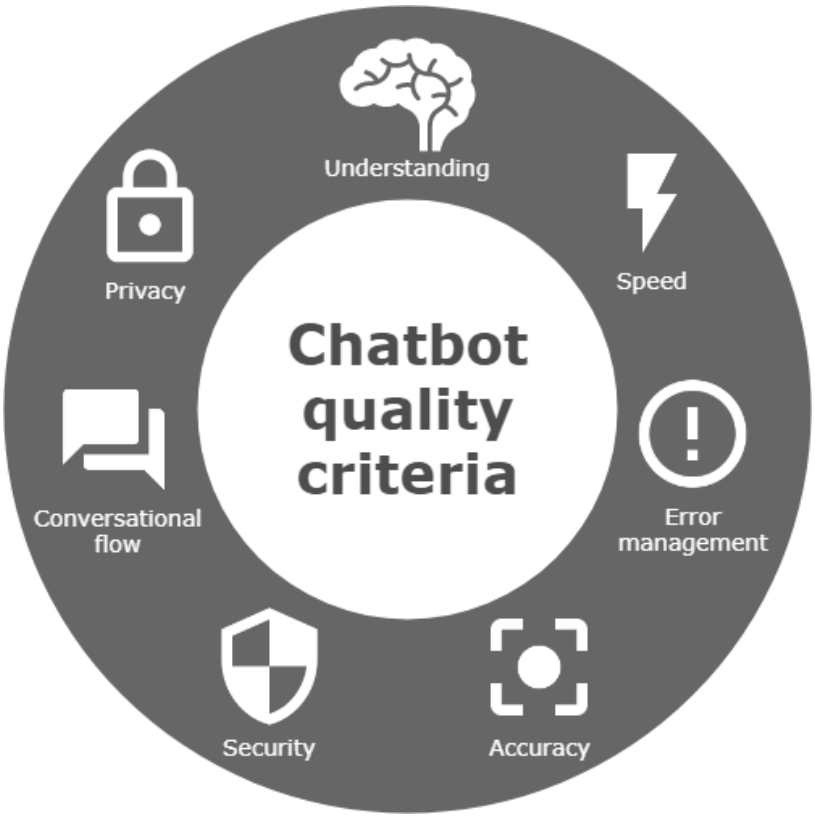

Figure 2 – Chatbot quality criteria

I identified seven important quality criteria for the chatbot, which should be considered in the process of developing the chatbot by developer, in the process of communicating with it by user and in the process of quality verification by messenger.

Understanding is a parameter that defines how well the chatbot understands what the user is writing to him and, as a result, what is necessary to respond to the user. Nowadays, chatbots are often not universal assistants, they are limited by the amount of information that relates to their subject area. To improve the quality of chatbot understanding, it is necessary to properly allocate the target domain and deliver it to the end user, so that there is no misunderstanding about user requirements and chatbot possibilities.

The speed of any software is an important criterion of quality, and chatbots are no exception. Acceptable time for feedback on messages can be compared with the response time on websites, according to many studies, the maximum acceptable response time is 2 seconds. If this time is longer, then the user will most likely think that there has been an error occurred in the chatbot and it is not possible to continue the dialogue.

In case of errors in chatbot understanding, it is necessary to set up a high-quality error management so that this error is passed in

conversation with the user as inconspicuously as possible and does not leave an unpleasant impression on the user.

Accuracy is very closely connected to the criterion of understanding. After the user's request was understood, it is necessary to generate a response that will satisfy the user as much as possible. The accuracy of this answer is based on whether the response was related to the topic of the incoming request, whether it was formulated clearly and whether the user was satisfied with the response.

Security in chatbots is one of the two most problematic criteria. The user want to be sure that software he uses does not harm him or his device. Now it’s impossible to find out that the response from the chatbot does not contain a link to malware. Such popular mobile application stores like Google Play and the AppStore face daily problems during the validation phase of the software which is uploaded to their stores. Even with the source code of the program, these platforms do not always can detect the difference between a good application and a malware. Concerning the chatbots, this task is much more complicated, because architecturally, chatbot is an API that can be connected to the messenger remotely, so there is no access to the chatbot source code.

Conversational flow determines how easy it is to chat with a bot. This criterion includes several components, one of which is the sequences of steps that need to be performed to obtain the result of a chatbot function. You can also measure the message metrics (keywords), as well as count them, to investigate how many messages are needed on average to accomplish the task with a chatbot.

The last criterion for chatbot quality is the privacy that it provides to users. Now the importance of privacy is growing, users are extremely interested that conversation with the interlocutor, regardless of whether he is a chatbot or a person, does not leave the boundaries of the dialogue. Obviously, if the chatbot sends information about sent and received messages to some third-party services, it will not satisfy potential users. Probably users will prefer an even less functional solution, in case it protects their privacy.

Now, based on the above model of chatbot creation and its evaluation, we will consider what problems are observed in a chatbot engineering and verification of its quality.

Tracking problems in conversational flow is a very difficult task, but in the future, it will be impossible. The fact is that there are several

models of chatbot behavior, the simplest one is the rule-based model, which means that each request to the chatbot has its unique response. Checking the conversational flow, understanding and accuracy of such bots is easy enough, because over the time the answers to the same query will not differ, that will allow certainly evaluate this quality criterion. This model concerns chatbots, which can be controlled even without using text writing, but only using the chatbot interface controls. Such chatbots are more similar to conventional mobile applications than smart assistants, but only this kind of chatbots can be verified by the conversational flow verification.

The most promising model for generating a response is the generative model, which implies that an answer is generated for each incoming message. To implement such model, the approaches of AI and NLP algorithms are used and as a result, it is impossible to estimate the quality of the chatbot answer in a long period of time, since at any time the answer to the same query can be different. Of course, in the case of obtaining data on the weather in a city using a single word, it is possible to unequivocally assess whether the answer was specific, but it still does not make sense to automate this process, since such simple bots are included in the intermediate stage of chatbot evolution.

There is an option to improve the quality of chatbot understanding, based on the importance of the system requirements. It is better to initially provide the user with a description of the entire range of chatbot capabilities, in order to form the scope of opportunities that the chatbot can provide. Thus, the creator of the chatbot can initially allocate acceptable boundaries for communication with its program. Obviously, the evaluation of this criterion is highly subjective, because you can’t expect any chatbot to return an accurate response to a random text. In general, you can represent two sets: the set of all messages related to the whole subject area and the set of those messages related to the subject area that the chatbot understands. Then the quality of chatbot understanding can be described by the ratio of the number of elements of the second set to the number of elements to the first set. The problem is that this calculation is simply impossible, because of the uncertainty of the sets.

The most measurable chatbot parameters are speed and error management. For the automated evaluation of these parameters, it is possible to create a test account, which will periodically access the

chatbot and collect information on the speed of response to various requests and how many requests have not been processed by the bot.

It should be emphasized that the effective verification of chatbot quality is important directly for several actors considered in this domain work. It is profitable for a developer to make a quality product that people will use. Users are interested in using fast, secure and stable software.

If you look at chatbot problems from the messenger's side, you can watch the big picture, which adds additional factors when evaluating chatbots, which you need to pay attention to. The important thing is how the messenger architecture is built in combination with chatbots. It is described in detail in the work of Simon Draxinger. The main fact is that messengers give developers the opportunity to connect their solutions, but not their hosting. That is, the messenger is not a platform for deploying chatbots, but a platform for providing potential users. In the work mentioned above, there is a logical argument for such an architecture: "Since they are not incorporated into the central core, they do not threaten speed and usability of the platform. Chatbots are not integrated into the core of the platform. Users that do not want to communicate with chatbots do not interfere or draw away resources from core activities, that is, instant messaging among users" [1]. Performance objectives are certainly an important factor of the modular architecture of the system. But now it can be observed that because of the implementation of such architecture in which the bots are created very simply and at the same time they pass a minimal verification process, the messengers do not feel responsible for the products that they allow to interact with their users. If the company does not want to lose reputation or become less competitive, then the level of quality requirements of any software that touches their system should not fall.

It is impossible to verify chatbot security or whether it transmits user data somewhere when messengers use such architecture approach. In any interaction, the user should understand that there is a probability that the dialogue will be transferred to the limits of the chatbot's responsibilities, or the chatbot will send a link to the malicious site. It is not true that the communication between the bot and the user is completely open. Many chat messengers (WeChat, Facebook, Telegram) support the transfer of money and these transactions occur safely, without sending credit card data to chatbots. The problem is that the

remaining stages of communication between the bot and the messenger are not so trusted.

There are two options that need to be considered, suggesting an improvement in the chatbot verification process. The first option deals with the case where messengers are aware of the responsibility for the chatbots. In this case, the best option is to add the ability to host chatbots and charge developers for this service. In this case, the user should be able to distinguish between bots that are hosted by the messenger (more secure) from others. This option resembles the work of the Google Play and AppStore app stores, because, it's needed to add analysis of the source code and it will allow evaluating such parameters as security and privacy. This option is the most improbable, since it is impossible to predict whether the messenger policy will change.

Another variant is the one we already have. He assumes that the state of things does not change and chatbots are as flexible as possible in terms of connecting to messengers and because of this unsafe and less open to assessing quality. In this case, it is possible to create a service that allows you to collect such information about the chatbot, which will increase the level of trust of users and will allow to form a rating of chatbots. In order to make the platform as honest as possible, blockchain technology can be used in the process of evaluating bots in order to avoid faking evaluation results. As additional features of the service, you can provide the opportunity to share the chatbot's source code, if it is open source, create an opportunity for communication between users and developers, automate speed tests and test the chatbot's performance.

The offered service solves many problems. The ability to view the source code and communicate between the user and the developer adds confidence in the quality of the product, because the developer is interested in communicating with the user to improve his software. Estimating the speed of the chatbot and the status of its activity is also an important parameter for evaluation and automation of this process provides users with a clear understanding of what kind of bot they are dealing with. With the inability to automate the verification of the communication process with the chatbot, you can fight using a formal process of assessing its quality by chatbot end users. As described above, such parameters as understanding, accuracy and conversational steps are very relative. Requirements for them for different bots may differ, which does not mean that one surpasses the other in terms of. It's just a matter of distinguishing initial requirements, subject areas and

methods of interaction. In order to evaluate these criteria, it is necessary to rely on opinions of real users.

Bayan Abu Shawar and Eric Atwell, who in their work on the measurement metrics for chatbot system [2], concluded that it is impossible to efficiently apply some universal process of assessing the quality of a bot (e.g. method used in The Loebner Prize Competition): "Evaluation should be adapted to the application and to the user needs. If the chatbot is meant to be adapted to provide a specific service for users, then the best evaluation is based on whether it achieves that service or task. Warwick Analytics [5]. It surveyed 551 chatbot developer and 93% claimed that human validation is important to maintain and improve the performance of their chatbots. This statistic proves that service is on demand nowadays.

During my research, I examined the subject area of the study, analyzed the messengers and their interaction with chatbots, identified the criteria for evaluating the quality of chatbots and the existing methods for verifying the quality of chatbots. I concluded that at this stage in the development of chatbot engineering, the use of human resources to assess the quality of chatbots is critically important. As a result, I proposed options for improving the process of verifying the quality of chatbots and described the practical implementation of these options.

References.

1. Simon Draxinger The Generativity of Messaging Platforms: A Case Study on Facebook Messenger and Chatbots // iSCHANNEL The Information Systems Student Journal, Vol. 12, Issue 1, pp. 4-9, 2017.

2. Bayan Abu Shawar, Eric Atwell Different measurements metrics to evaluate a chatbot system // Bridging the Gap: Academic and Industrial Research in Dialog Technologies Workshop Proceedings, pp. 89-96, 2007.

3. Murch R. The Software Development Lifecycle - A Complete Guide // 132 p., 2012.

4. Chatbot testing. [https://chatbotslife.com/chatbot-testing-1f359b5459a]. Accessed April 12, 2018.

5. Love J. Warwick Analytics research finds human-in-the-loop validation critical for chatbot owners. [https://warwickanalytics.com/warwick-analytics-research-finds-human-in-the-loop-validation-critical-for-chatbot-owners/]. Accessed April 13. 2018.