Chatbot Design Principles

Maria Ferman

July 14, 2017

Currently, it is necessary to make software customized and accessible for users who do not have any technological background. Fortunately, chatbots can help with both issues. Chatbots allow users, through their own words, to interact and communicate with software. Therefore, software can become easier to understand and use. Chatbots also allow the interface to save the user preferences. By doing this, software can adapt to the interest of the users. According to Shevat, "Bots are a new user interface. This new user interface lets users interact with services and brands using their favorite messaging apps. Bots are a new way to expose software through a conversational interface" [1]. The design of a chatbot is more effective by using specific guidelines that allows to the designer not to start from scratch, but to give him or her a specific structure of how to design an efficient chatbot.

Chatbot Purpose

The purpose of the chatbot should be clear to users. A chatbot that makes a user feel confused about its purpose is futile. Users need to have a clear understanding of why the chatbot was designed and what its capabilities and limitations are.

Chatbot designers need to identify a balance between the constraints of the interface and the goals of users that they want to accomplish. Besides, this designers should analyse what kind of chatbots they require, and if it is feasible to create it or not. After finding this balance and making this analysis, the designers should start to define what the chatbot 's capabilities will be (what the chatbot can do in order to help users in their tasks).

In addition, designers should understand when to use a direct manipulation interface and where to use a chatbot. Usually, conversational agents that do not have a clear purpose make the user feel that the job can be done by a regular website. Chatbots can be very useful as long as they are well-developed and oriented to the purpose.

Conversational Shared Knowledge

Conversations need to follow certain rules and to have a certain semantic order for the sake of having a clear understanding between participants. Therefore, to have a meaningful conversation between chatbots and users, the chatbot script should follow the same rules and semantic order as the conversations. The user should be able to follow the meaning of the chatbot remarks, while the chatbot should be able to select a response that fits coherently into what the user is

saying. Therefore, it is vital to understand how chatbots and users follow each others' thoughts.

According to Reichman [2] people need to share situational knowledge in order to understand and follow a conversation.

Common situational knowledge

In order to have a mutual understanding in a conversation, it is necessary that participants share a Situational Context. Situational Context involves previous knowledge about the topic that is being discussed.

Listeners must be able to follow the context of the conversation; they need to grasp what the speaker's point is and what he or she said before. Listeners need to understand the prior and the current context of the conversation to avoid being confused and to grasp the conversation's assertion. At the same time, speakers need to follow the order of a conversation to make their remarks easily understandable to the listeners. Therefore, the user and the chatbot should *share the same information to be on the same page.*

Alexa, Amazon's voice control system, is a clear example of the importance of context for conversational agents. When a San Diego TV station was covering the report of a girl ordering a dollhouse by mistake, the viewers' Alexa devices also recognized the command to purchase a dollhouse. "If Alexa has recognized the context of the news programme (the TV show presenter speaking in a past tense), then Alexa would not have taken action" [3].

Chatbot Interactions

According to the Merriam-Webster dictionary, interaction is defined as the action or influence of things on one another [4]. Chatbots should have a well-designed series of interactions with users. Every single intended interaction should have a specific script of the different conversations that the chatbot and the users may have. Conversational scripts allow users to have a "draft outlining the situation and actions the chatbot need to support" [5]. Those scripts enable to determine the limitation of the chatbot by having specific scenarios about the conversation.

The most important and complex part of designing a chatbot is the creation of the conversational script, because the user can take many paths (chatbots scenarios) in order to achieve one task [6]. In other words, users can select different paths which all should arrive at the same destination. One effective way to create those different scenarios is using mind maps, and then designers can use tools like InVision in order to create the conversational chatbot mocks.

Greeting

Since the first interaction, users should have a clear idea of the chatbot main functionality and what they can expect from it. The welcome or greeting section is the first interaction in which chatbots should clearly communicate what they are capable of doing, in order to avoid functionality misunderstandings. According to Lurchenko, the information in the welcome screen should not exceed 160 characters [7].

Conversation

The conversation section is when the topic will be developed. According to

Mishra, "Effective communication is getting the message across as intended and getting desired feedback by influencing and attracting attention" [8]. Therefore, chatbots should be able to express a clear message to the user in order to have an effective communication. This section can have assertions, questions, comments, and feedback. The participants of the conversation should take turns interacting in order to communicate a clear message.

Farewell

The farewell section is the last part of a conversation. By this phase, the users should have had the help that they need in order to finish their task and to have relevant feedback.

Conversational content or interaction elements?

According to Allen et al. [9] the speaker can emit several elements, such as assertions, questions, and support in a conversation. Chatbots can emit the same elements.

Assertions are affirmations or messages that the chatbot gives to the user; assertions determine the content of the conversation and allow the listener to increase hir or her knowledge about a certain topic.

Questions can be used to make sure that the provided information is required by the user. However, designers should avoid the open-ended questions, because users may respond in a non-supported way. A solution for this may be the use of questions that provide three different options in buttons, and the chatbot should ask users to select one. If the options provided by the chatbot contain several images, a good way to display them is the use of carousels.

According to Pernice "Carousels enable more than one piece of content to occupy the same piece of prime real estate on the homepage, which can help diffuse any infighting about whose content is most deserving" and it should not be included more than 5 frames in each carousel because it is hard for users to remember more than five topics at the same time [10].

Finally, ***support*** is any extra help that the chatbot has given to the user. How users can get help, should be in a straightforward way. Commonly, users only need to type the word "help" or the command "/help" in order to get an additional support from the chatbot. Besides, when an error has occurred the chatbot should automatically ask users the question about if they need help. Besides, chatbots can also give feedback as a part of the conversational support, or provide meaningful insights about the data analysed or the conversation that the user and the chatbot had.

The conversational script should take into account the following elements: timing, being an active participant and an active listener, and avoiding over sharing information.

The chatbot's messages elements can include text, emojis, attached files, images, video or audio. The message can be also structured; this type of messages follow the command line format and usually has buttons to allow users to get information or to answer a question previously asked by the chatbot. According to Lurchenko the content of the buttons should not exceed 20 characters including spaces [7].

Chatbot Language

According to Allen et al. [9] in order to have a meaningful conversation, it is necessary to share a language and vocabulary in common. Chatbots should use the same language and words as the user, and avoid using technical and system-oriented terms. If the design of the conversational agent is oriented to certain populations, chatbots can mimic how users normally speak. Therefore, before the creation of the chatbot, it is necessary to have a "solid understanding of the audience we seek to appeal to, and to have a vocabulary familiar to them" [11].

Chatbot conversation flow

According to Reichman [2], in an usual conversation many details are being shared. To avoid an incoherent conversational flow, it is necessary that listeners understand when and why the conversational topic has been changed. For instance chatbots can twist the conversation from talking about a certain graph and when the user has selected the graph, to what color is better to use. Shifting the conversation topic should be made carefully to avoid misunderstandings.

Chatbots need to follow a structured development of the conversation, also known as a conversational script, to avoid references that are not clear to the participants, and engage users in the conversation.

The conversation between the user and the chatbot should not lead to any ambiguity about what to say in the conversation. Therefore, the chatbot's conversation should use structured messages that are clear and concise. By using structured messages chatbots can "guide a user through the interaction" [11].

However, designers should be careful of not letting the chatbot lose the feeling of a normal conversation. Some chatbots only use structured messages; for this reason they seem to be a command line, instead of giving the user the conversational feeling that chatbots should have. The flow of the conversation should be natural and evident.

User's Control

Users are accustomed to and like having control over the software. Therefore, it is extremely important to make the user feel that he or she has the control over the interface. Chatbots should not make users believe that they are powerless over the interface; they are there to help users to have a better experience using the interface. Chatbots should be guides or an additional aid that allows users to understand the interface and have better results. According to Sheneiderman [12] it is "necessary to give the users the feeling of being in control and therefore they can be responsible for the decisions they make."

User's Freedom?/chatbot undo and redo?

It is normal that users make mistakes when they are choosing different options in an interface. Chatbots should have the ability to undo or redo the changes that the user just selected. In that case, users know they are able to undo an unwanted change triggered by a misinterpreted message or a mistaken click [11].The weather forecast conversational agent Poncho has the ability to ask

users if the information provided was appropriate; if not, users can ask again for the right location [13]. In addition, the users should be able to ask for extra information at any point in the conversational flow.

Error Messages

Chatbots should express error messages in a straightforward way. They should not use codes or technical words in order to avoid confusing and stressing the user. Chatbots should be able to express that an error has occurred and add suggestions about how to solve or undo that error.

Error prevention allows users to understand that something not expected just happened in the interaction. For example, an error can be a wrong input data from the user. In those cases (events *) the chatbot should allow users to get additional help or undo what they just have done. Some chatbots allow to speak with a live agent in the event that the user is struggling with the chatbot interaction. An example of this is 1-800-Flowers; this conversational agent can connect a live agent if the conversation with the chatbot fallow for a certain period of time [14].

In conclusion, a good chatbot should be able of providing evidence around error prevention and allow users to recover when an error just happened [11].

User's Recognition and Recall

According to Scott, users do not like to read a large amount of text: "they will read the first message and then their eyes glaze over. They skim the rest of the text" [11]. Besides, the principle of least effort, set forth Zipf, specifies that people use shortened words and expression in a speech, in order to obtain the maximum communication by using the least cost [9]. Therefore, chatbots should keep their dialogues short and give the users a small number of options, in order to avoid users getting lost or feeling overwhelmed.

In addition, users do not remember details of the options given by an interface. When users avoid reading a large amount of text, they may misunderstand the chatbot's message and finish with an unsuccessful result. The design of the chatbot dialogue should only have relevant information to help users with their tasks and prevent a dull interaction with the chatbot.

Chatbot's Personality

The Oxford dictionary defines personality as "the combination of characteristics or qualities that form an individual's distinctive character" [15].

A Chatbot's personality is what makes a chatbot different. By adding extra information to the dialogue and having a matching language, the chatbot can have a distinctive personality. However, designers should be careful to not include too much additional information that makes the user feel bored or annoyed.

According to Spectrm "The personality of a chatbot consists of a purpose, matching characteristics and a suiting tonality. Crafting a personality is the most important part of chatbot creation" [16].

The chatbot's personality should take into account the target audience for which it was designed. The personality should match with the users: therefore,

how friendly, sarcastic or humorous should only depend on the people who will use the chatbot. Users like to interact with chatbots in a human way [11]. For this reason designers should consider how to engage users in the conversation.

A clear example of how to get a user engagement is the Aura's bot. Aura Dione is a Danish singer who is using a chatbot in order to be in contact with her fans. The designer of the Aura's bot use the personality of the singer. Therefore, the tone of the chatbot is like an artist, and fans can communicate with it and ask extra information about the singer. Fans have the feeling of being interacting with the singer, and this lead to the user engagement and retention [16].

According to Scott [11], there is a difference between the content (relevant information to help the user) and the medium (the chatbot's personality). Users need to be entertained and at the same time to be helped.

Spectrm establishes a series of questions in order to select the right personality for a chatbot: "Do you have a mascot (for a product or brand)? Does this mascot have a personality that might serve as inspiration? Is there a core value for your company in general or to its way of communication that could inspire the personality? Can you think of a person you know, a celebrity or a fictional character that your bot could resemble?" [16]. Those questions allow to designers to create a template or structure about how they want the chatbot behaves and interact with the users. Another important considerations to take into account is the age, heritage, friends and occupation that the chatbot may have. All these specifications enable designers to have a clear definition of to whom the chatbot will resemble.

In conclusion the chatbot's success may be defined by how designers balance between entertainment and guidance. Therefore, the difference between chatbots that are being used and those that are not, is how compelling and pleasant is the chatbot experience.

Chatbot's versatility

Chatbots should be versatile enough to be able to understand open questions, commands, and requests written by the user. If a user wants to use a command in the interaction with the chatbot, the dialogue should continue smoothly without breaking the flow. Commands start with a slash and they are not longer that 32 characters, those characters can be Latin letters, numbers and underscores [17]. Commands can also be followed by an argument.

Chatbot's memory

Chatbots have the capacity to keep track of the user's interest and preferences, even if they change over time. Conversational agents can save the user's habits and preferences in the different sessions that he or she had with the chatbot [12]. As an example of this, Poncho can save the location of a certain user. Therefore, the next time that the user asks for a weather broadcast, Poncho already knows which location to look at, [13] saving the user time and keystrokes. Besides, this allow the user to have a customised experience.

Chatbot's limitations

Chatbots should have a delimited script of the possible scenarios. If a user wants to go through an unknown topic for the chatbot, the chatbot has to specify to the user that this question goes beyond its knowledge. For example, when a user asks Poncho, the weather chatbot, something that it was not created to answer, Poncho replies with: "Oops, I didn't catch that. For things I can help you with, type help" [11]. Chatbots need to gracefully establish limits to users that want to go beyond the chatbot's knowledge.

Chatbot's documentation

Chatbots should be used without the need for extra information. How to use a chatbot should be a simple conversational flow. However, there are some users that like to deeply understand how the interface works and detect its limitations. A chatbot appropriate documentation should be precise and short. It should provide extra information about how the chatbot can help users in their tasks. Besides, it also "should include a concrete list of what steps can be carried out" and the critical points of the chatbot [11].

Finally, the chatbot should provide an easy way to access the documentation. One example of this is when the user type slash "/" a list of the supported commands of the chatbot should appear on the screen [17].

References

[1] Amir Shevat. *Designing Bots Creating Conversational Experiences.* O'Reilly Media, May 2017.

[2] Rachel Reichman. *Getting computers to talk like you and me: discourse context, focus, and semantics.* MIT press Cambridge, MA., 1985.

[3] Jonathan Bree. What we can learn from alexa's mistakes. Chatbots Magazine, February 2017.

[4] Simon and Schuster. *The new merriam-webster pocket dictionary.* Richmond Hill, Ont, 1971.

[5] Robert Sens. Designing a chatbot conversation. Behance, May 2016.

[6] Obaid Ahmed. Designing chatbot conversations. Chatbot's Life, March 2017.

[7] Anna Lurchenko. Cheat sheet: All facebook messenger bots interactions. Chatbots Magazine, February 2016.

[8] Sneha Mishra. Effective communication. BUSINESSTOPIA, 2017.

[9] Donald E. Allen and Rebecca F. Guy. *Conversation analysis: the sociology of talk*, volume 200. Walter de Gruyter, The Hague, 1978.

[10] Kara Pernice. Carousel usability: Designing an effective ui for websites with content overload. Nielsen Norman Group, September 2013.

[11] Kevin Scott. Usability heuristics for bots. Chatbots Magazine, May 2016.

[12] Ben Shneiderman and Pattie Maes. Direct manipulation vs. interface agents. *interactions*, 4(6):42–61, 1997.

[13] Poncho Inc. Poncho, 2017.

[14] TopBots. 1-800-flowers, 2017.

[15] *Oxford Dictionaries*. Oxford University Press, 2017.

[16] Spectrm. Your chatbot's personality is the key. Chatbots Magazine, February 2017.

[17] Bots: An introduction for developers, October 2014.