

# Towards best practices for a visualization chatbot

Maria Ferman

August 11, 2017

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. The efficiency of the chatbots can be augmented by considering the user preferences. By doing this, software can adapt to their preferences. According to Shevat, “Bots are a new user interface which 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].

In order to make the interaction between the user and the chatbot more efficient, designers required a comprehensive list of best practices in chatbot’s design. The design of a chatbot is more effective when the designers use specific practices that allows them not to start from scratch, but to give them a specific structure of how to design an efficient chatbot.

## Objective

Develop a set of best practices that allow designers to create and design a chatbot.

## Motivation

To help designers to create more efficient chatbots by having a good (complete, accurate)\* set of best practices. The best practices will allow users to have a specific guide about the chatbot’s creation.

## Project Description

The project consists of three main phases. The first is the design of a Chatbot’s best practices. In order to develop a good set of best practices, it will be necessary to read blogs and best practices online. Besides of selecting and using the most popular chatbots. The second phase is the muck-up of a visualization chatbot that fulfil (according to)\* the best practices of the first phase. Finally the third section is the case study. This study will be carried out to evaluate the chatbot best practices. Therefore, if the chatbot fulfil the user requirements (needs)\* in the case study, this is a clear indication that the best practices are

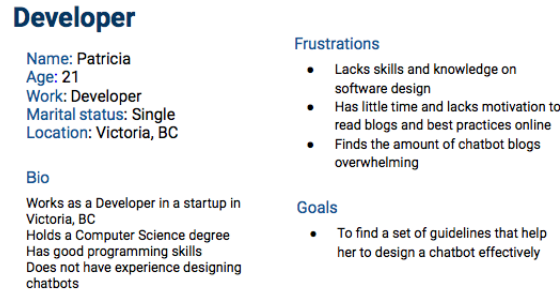


Figure 1: Best Practices' Persona

adequate (accurate)\* for the creation of a chatbot. (that gives evidence that the best practices are accurate.)\*

Throughout the project I will use two different user-personas (Laura and Patricia). The creation of the two personas will guide me in the creation of these best practices, and to help me in the design of the chatbot. Finally, the personas will help me in the case study to select users that are close to the description of the project user-personas.

The first user-persona is oriented to the Chatbot's best practices (Figure 1). Patricia is a developer who works in a startup company in Victoria, BC. She holds a Computer Science degree and has good programming skills. However, Patricia does not have experience designing chatbots. She lacks skills and knowledge on software design, and she has little time and lacks of motivation to read blogs and best practices online. In addition, she finds the amount of chatbots blogs overwhelming. Therefore, her goal is to find a set of guidelines that help her to design a chatbot effectively.

The second user-persona is oriented to the chatbot (Figure 2). Laura is a Visualization novice who works as a project manager in a mid-sized company in Victoria, BC. She has managed several projects successfully. However, she does not have any background creating visualizations. On a daily basis, she uses her computer, iPad, and mobile phone to work. She has problems in creating visualizations to explain her ideas and project results. She has little time to devote in the visualization creation process, and struggles in learning new technologies. Besides, she finds the amount of visualization tools overwhelming. Laura would like to find a good way, by using her mobile phone, to create an effective visualization. In addition, she would like to get help in the visualization creation and data analysis process.

**Chatbot Best Practices' Design:** The best practices are created to allow people like Patricia to design an effective chatbot. These best practices give developers the most important aspects about the chatbot's design. Therefore, from the beginning they will have a specific structure to follow of how to design their chatbot.

**Chatbot:** The goal of the chatbot is to help people like Laura to create a meaningful visualization about their data. Laura can use natural language statements in order to communicate with the chatbot. In response, the chatbot will provide her with several visualization options.

**Chatbot's Tasks on Visualization:** The Visualization goal of the chat-

## Visualization novice

**Name:** Laura  
**Age:** 25  
**Work:** Project Manager  
**Family:** Married, 2 kids  
**Location:** Victoria, BC

### Bio

Works as a Project Manager for a mid-sized company in Victoria, BC  
Holds a Business degree  
Does not have any background creating visualizations  
Has managed several projects successfully  
Uses her computer, iPad and her mobile phone for work on a daily basis

### Frustrations

- Has problems in creating visualizations to explain her ideas and project results
- Has little time to devote to visualization creation
- Struggles in learning new technologies
- Finds the amount of visualization tools overwhelming

### Goals

- To find a good way to create an effective visualization
- To be able to use her phone to create a visualization
- To get help in visualization creation and data analysis

Figure 2: Chatbot’s Persona

bot is to help users to analyse data by using visual context images (graphs). The chatbot will offer different options to visualize data. The options will be oriented to allow users to see patterns, trends, and correlations that might be not detected by the use of data tables or excel sheets. Therefore, the chatbot will allow users to easily analyse their data in order to have deeper insights.

## 1 Chatbot Purpose

The purpose of a chatbot should be clear to its 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.

Currently, chatbots’ designers can deal with this issue in two different ways. The first is to have a clear explanation about the chatbot’s capabilities at the beginning of the interaction with the users. A clear example of a chatbot purpose definition is the CNN chatbot (Figure 3). When the user opens the chatbot for the first time, the chatbot describes him or her the top stories and gives an explanation about how to get more news according to his or her preferences. The second option is to allow users to discover by themselves the purpose and capabilities of the chatbot during the interaction.

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, the kind of chatbot needed should be decided based on the final user’s requirements, and if it is feasible to create them or not. After finding this balance and making this analysis, the designers should define what the chatbot’s capabilities will be (i.e., 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 (e.g., graphic user interface) and when 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.

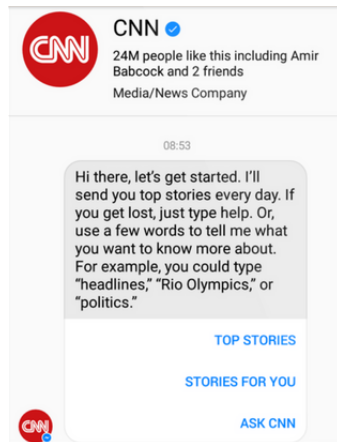


Figure 3: CNN Chatbot

## 2 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 the user and the chatbot. Therefore, to have a meaningful conversation, the chatbot script should follow the same rules and semantic order as a regular conversation. In other words, there should be a mapping between the script and the actual conversation. Designers need to understand what the actual conversation will feel like for the users and then create the chatbot accordingly.

The user should be able to follow the meaning of the chatbot's 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.

### 2.1 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.*

(Figure 4)



Figure 4: Poncho Weather Forecast Chatbot



Figure 5: Visualization Chatbot's Conversational Mind Map

### 3 Chatbot Interactions

According to the Merriam-Webster dictionary, interaction is defined as the action or influence of things on one another [3]. 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” [4]. Those scripts 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 (conversation scenarios) in order to achieve one task [5]. In other words, users can select different paths which should all arrive at the same destination. One effective way to create those different scenarios is using mind maps. According to Eppler [6], “Mind maps are defined as a multicoloured and image centred, radial diagram that represents semantic or other connections between portions of learned material hierarchically”. (Figure 5)

After creating the mind maps, designers can use tools like InVision <sup>1</sup> in order to create the conversational chatbot mockups.

#### 3.1 Greeting

From the first interaction between a user and a chatbot, the users should have a clear idea of the chatbot context and what they can expect from it. Therefore, users should know the context in which the chatbot need to be used.

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. The welcome section of the conversation should be short and concise, to avoid users feeling overwhelmed due to the amount of information provided by the chatbot.

<sup>1</sup>[www.invisionapp.com](http://www.invisionapp.com)

### 3.2 Conversation

The conversation section is when the topic will be developed. According to Mishra [7], “Effective communication is getting the message across as intended and getting desired feedback by influencing and attracting attention”. 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.

### 3.3 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. In this section, the chatbot should ask the user if the provided information or help was useful, and if the user would like to do anything else.

## 4 Conversation Elements

According to Allen et al. [8], the speaker can emit several elements, such as assertions, questions, and support in a conversation. Chatbots can emit the same elements. The following interaction elements are described from the chatbot’s perspective.

### 4.1 Assertions

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 his or her knowledge about a certain topic.

### 4.2 Questions

Questions can be used to make sure that the provided information is required by the user. An example is when Poncho<sup>2</sup>, the weather forecasts chatbot, asks users about the city for which they want to know the weather. If the user would like to know the weather in Manhattan, Poncho will ask the following verification question, “New York, NY? Right now it’s mostly cloudy there. Is that the right city?” (Figure 6).

However, designers should avoid open-ended questions, because users may respond in a non-supported way. A solution for this may be the use of questions that provide different options in buttons inside the chatbot message, and the chatbot should ask users to select one. Each provided option will invoke different actions. If the options provided by the chatbot contain several images, a good way to display them is the use of carousels.

According to Pernice [9], “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 five frames in each carousel because it is hard for users to remember more than five topics at the same time.

---

<sup>2</sup><https://poncho.is/>

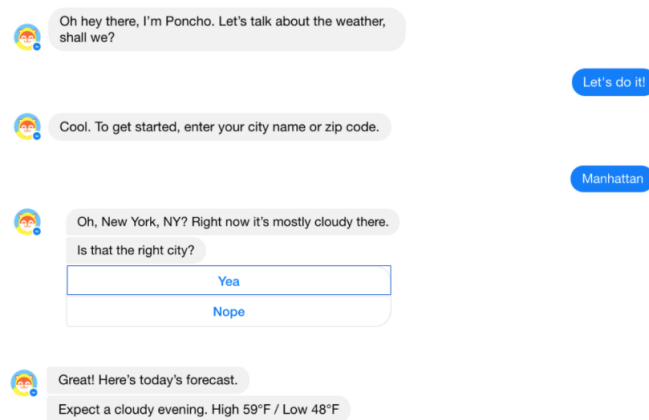


Figure 6: Poncho Weather Forecast Chatbot

### 4.3 Support

Finally, support is any extra help that the chatbot gives to the user. This help should be straightforward for users. Commonly, users only need to type the word “help” or the command “/help” in order to get an additional support from the chatbot. When an error has occurred the chatbot should automatically ask users if they need any help. Besides, chatbots can also give feedback as a part of the conversational support, or provide meaningful insights about the performed task or the conversation that the user and the chatbot had. In addition, users should be able to ask for extra information at any point in the conversational flow.

## 5 Conversational Script

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 structured; these types of messages follow the command line format and usually have buttons to allow users to get information or to answer a question previously asked by the chatbot. The content of the message should be short in order to have a minimal design and a better user comprehension.

## 6 Chatbot Language

According to Allen et al. [8], in order to have a meaningful conversation, it is necessary to share a language and have 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



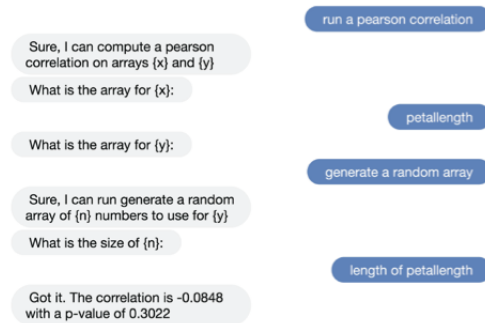


Figure 7: Iris Data Science Chatbot

of the audience we seek to appeal to, and to have a vocabulary familiar to them” [10].

(Figure 7)

## 7 Chatbot Conversation Flow

According to Reichman [2], in a 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. The conversational chatbot flow should follow a certain order. This means that after particular information was given, the chatbot should perform a certain task. After these steps, the conversational flow may change. Users can change the topic of the conversation at any point. For example, ShopBot<sup>3</sup> from Ebay helps users to select a pair of jeans, and after selecting the desired jeans, the user can also ask for a blouse from a different department (Figure 8). Another example is Poncho, users can ask him about the weather, and then change the conversation’s topic and have a little chat with Poncho about astrology.

\* For instance, the design of the 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 /find another example/. Shifting the conversation topic should be made carefully to avoid misunderstandings. \* ? (Figure 9).

Chatbots need to follow a conversational script that is a structured development of the conversation, 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 be structured and have messages that are clear and concise. By using structured conversation, chatbots can guide a user through the interaction [10].

However, designers should be careful of not letting the chatbot lose the feeling of a normal conversation. Some conversations might be structured too rigidly; for this reason they seem to be a command line, instead of giving the

<sup>3</sup><https://shopbot.ebay.com/>

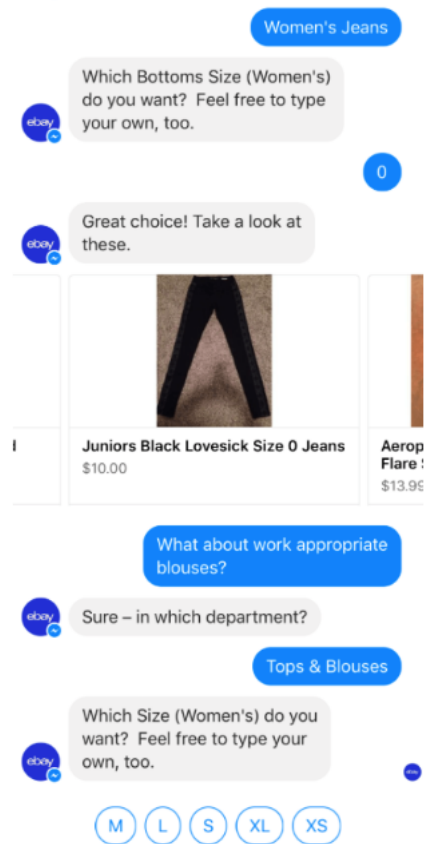


Figure 8: ShopBot Ebay

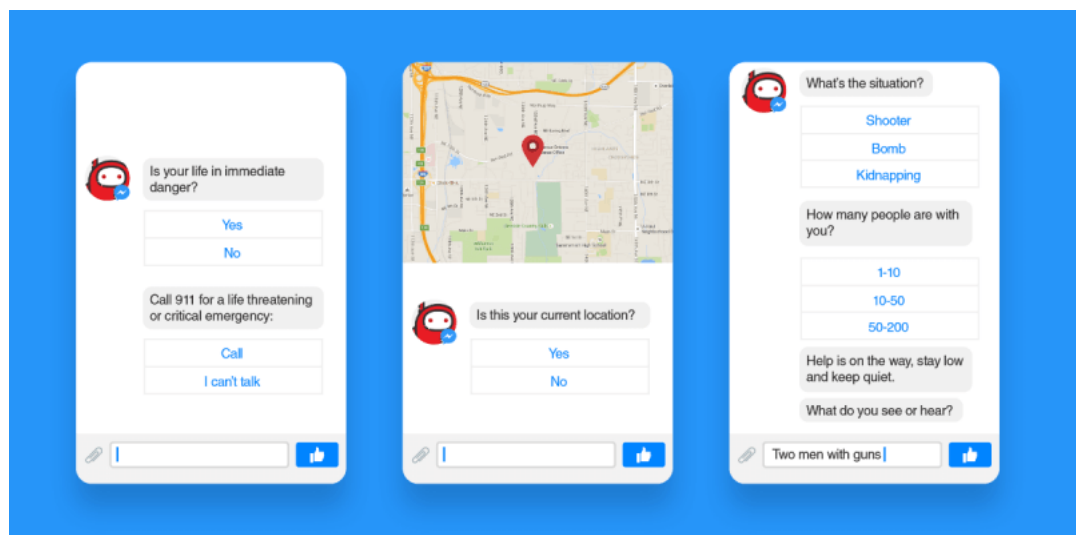


Figure 9: Rescue Emergency Chat Service



Figure 10: Chatbot Without an Undo Capability

user the conversational feeling that chatbots should have. The flow of the conversation should be natural and evident.

## 8 User's Control

Users are accustomed to and like having control over the software. For instance, when a user has the ability of doing more customizations in an interface, they tend to feel more engaged in using that interface, because it has the user preferences. Therefore, it is extremely important to make the user feel that he or she has control over the interface. According to Maes [11], “users must be able to turn over control of task to conversational agents but users must never feel out of control”.

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 Shneiderman [11], it is “necessary to give the users the feeling of being in control and therefore they can be responsible for the decisions they make.” - \* Only users can change the topic of the conversation? or also the bot?

## 9 Making Changes on the Fly

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 responses that the user gave. In that case, users know that they are able to undo an unwanted change triggered by a misinterpreted message or a mistaken click [10]. The chatbot Poncho has the ability to ask users if the information provided was appropriate; if not, users can clarify their question. (Figure 10).

## 10 Error Messages

Chatbots should express error messages in a straightforward way. They should not use codes or technical words to avoid confusing and stressing the user. Chat-

bots 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 unexpected just happened in the interaction. For example, an error can be a wrong input data from the user. In the case of such an event, the chatbot should allow users to get additional help or undo what they have just 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 Assistant <sup>4</sup>, this conversational agent can connect a live agent if the conversation with the chatbot fallows for a certain period of time.

To summarize, a good chatbot should be able to provide evidence around error prevention and allow users to recover when an error just happened [10].

## 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” [10]. The principle of least effort, set forth by Zipf, also specifies that people use shortened words and expression in speech, in order to obtain the maximum communication by using the least cost [12]. 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.

## 12 Chatbot’s Personality

The Oxford dictionary defines personality as the combination of characteristics or qualities that form an individual’s distinctive character [13]. A chatbot’s personality is what makes a chatbot unique. 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. There has to be a balance between giving just the relevant information and adding extra information to make the chatbot’s conversation more engaging. Therefore, the chatbot personality is an important part to make the user more engaged with the interface.

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 [10]. For this reason designers should consider how to engage users in the conversation.

---

<sup>4</sup><https://www.facebook.com/1800FlowersAssistant/>

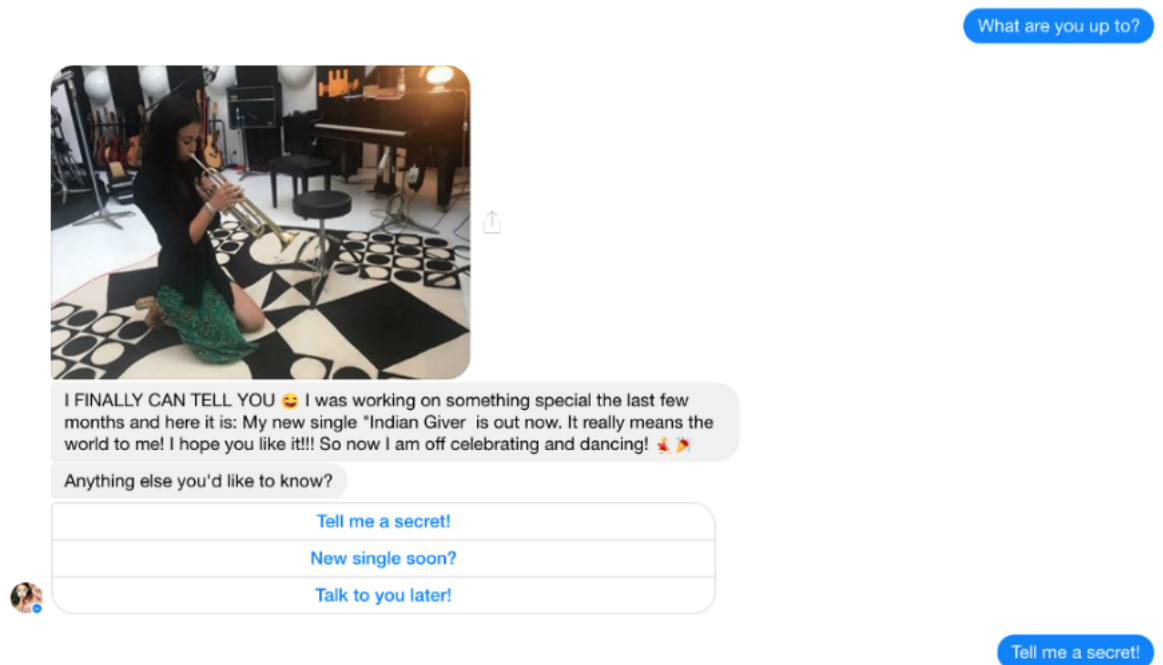


Figure 11: Aura's Bot

A clear example of how to get user engagement is the Aura's Bot <sup>5</sup>. 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 uses the personality of the singer. (Figure 11). 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 interacting with the singer, and this leads to the user engagement and retention [14].

According to Scott [10], 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.

Spectrum 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?" [14]. Those questions allow designers to create a template or structure about how they want the chatbot to behave and interact with the users. Other important considerations to take into account are the age, heritage, friends and occupation that the chatbot may have. All these specifications enable designers to have a clear definition of whom the chatbot will resemble.

<sup>5</sup><http://m.me/aurapower>

To summarize, 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 the chatbot experience is.

## 13 Chatbot’s Audience - Should be a subsection of Personality?

Chatbots can be developed for business, customer oriented tasks, or just be oriented to teen customers [1]. Business oriented tasks’ chatbots are designed to help teams to collaborate in a more effective way. This kind of chatbots can be useful to help software developers in their task because “it can help reduce the friction points they have to face when working collaboratively” [15]. An example of a business oriented task chatbot is the slackbot<sup>6</sup>. The consumer oriented tasks’ chatbots are designed to help users communicate, get additional information or help. An example is Poncho the weather forecast chatbot. Finally, the teen oriented chatbots are developed to entertain young customers and these kinds of users like to play games, chat and share content with friends. An example of this chatbot is Kik<sup>7</sup>.

According to Shevat, in order to define the chatbot’s audience, developers need to answer the following questions: Are you addressing a business use case? Is this a consumer use case? Are you targeting teens? Families? Adults at work? When are they using the service? [1].

## 14 Chatbot’s Versatility

Chatbots should be versatile enough to be able to understand open-questions, commands, and requests written by the user. An example of an open-question is when users ask Poncho, “What’s the weather like in Brooklyn this weekend?” and then Poncho answers with the Brooklyn weather forecast.

However, if a user wants to use a command in the interaction with the chatbot, the dialogue should continue smoothly without breaking the flow. Commands to chatbots can be conveyed via special characters followed by an argument. For example, chatbots’ commands in Slack and Telegram start with a slash followed by an argument. The characters of the argument can be Latin letters, numbers and underscores [16]. It is a common practice to have slash commands in the following pattern: slash “command name” “arguments” [1].

## 15 Chatbot’s Memory

Chatbots should be built with the ability 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 [11]. 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

---

<sup>6</sup><https://api.slack.com>

<sup>7</sup><https://bots.kik.com/>

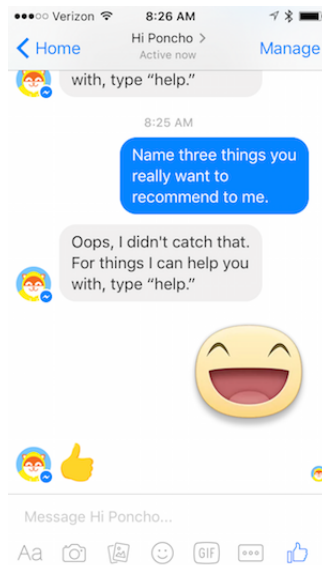


Figure 12: Poncho Weather Forecast Chatbot

broadcast, Poncho already knows which location to look at, saving the user time and keystrokes. Besides, these allow the user to have a customised experience.

## 16 Chatbot’s Communication Capabilities

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*” [10] (Figure 12). Chatbots need to gracefully establish limits to users that want to go beyond the chatbot’s knowledge.

## 17 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 [10].

Finally, the chatbot should provide an easy way to access the documentation. One example of this is when the user types slash “/” and a list of the supported commands of the chatbot should appear on the screen [16].

## 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] Simon and Schuster. *The new merriam-webster pocket dictionary*. Richmond Hill, Ont, 1971.
- [4] Robert Sens. Designing a chatbot conversation. Behance, May 2016.
- [5] Obaid Ahmed. Designing chatbot conversations. Chatbot's Life, March 2017.
- [6] Martin J Eppler. A comparison between concept maps, mind maps, conceptual diagrams, and visual metaphors as complementary tools for knowledge construction and sharing. *Information visualization*, 5(3):202–210, 2006.
- [7] Sneha Mishra. Effective communication. BUSINESSSTOPIA, 2017.
- [8] Donald E. Allen and Rebecca F. Guy. *Conversation analysis: the sociology of talk*, volume 200. Walter de Gruyter, The Hague, 1978.
- [9] Kara Pernice. Carousel usability: Designing an effective ui for websites with content overload. Nielsen Norman Group, September 2013.
- [10] Kevin Scott. Usability heuristics for bots. Chatbots Magazine, May 2016.
- [11] Ben Shneiderman and Pattie Maes. Direct manipulation vs. interface agents. *interactions*, 4(6):42–61, 1997.
- [12] George Kingsley Zipf. *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio Books, 2016.
- [13] *Oxford Dictionaries*. Oxford University Press, 2017.
- [14] Spectrm. Your chatbot's personality is the key. Chatbots Magazine, February 2017.
- [15] Carlene Lebeuf, Margaret-Anne Storey, and Alexey Zagalsky. How software developers mitigate collaboration friction with chatbots. *arXiv preprint arXiv:1702.07011*, 2017.
- [16] Bots: An introduction for developers, October 2014.