



# Chatbot Dimensions that Matter: Lessons from the Trenches

Juanan Pereira<sup>(✉)</sup>  and Óscar Díaz 

University of the Basque Country (UPV/EHU),  
Manuel Lardizabal Ibilbidea, 1, 20018 Donostia, Gipuzkoa, Spain  
{[juanan.pereira](mailto:juanan.pereira@ehu.eus),[oscar.diaz](mailto:oscar.diaz@ehu.eus)}@ehu.eus

**Abstract.** Chatbots are becoming pervasive. From health assistance to cooking advisers, a myriad of chatbots are out there to help you in a broad range of activities. Natural Language capabilities have received a lot of coverage as one of the most distinctive and enabling features of this revolution. Nevertheless, other less glittering dimensions might well also play a major role on chatbots' success. This paper reports on the experience of three chatbots development. Here, unsophisticated script-based conversational capabilities are provided but the focus was on other four dimensions: interaction, integration, testing and analytics. The paper reports on the activities, arranged along these four dimensions. Our final aim is to bring to the forefront the concern for those dimensions as also main enablers of chatbot success.

**Keywords:** Chatbots · Development · Effort estimation · Education  
Health · MOOCs

## 1 Introduction

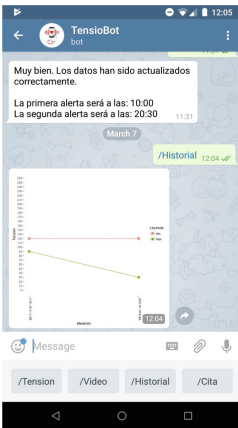
Chatbots are *not* a new kid on the block. They have been around since 1966 (Eliza bot, [7]). However, since their integration in the instant messaging (IM) application world (Facebook Messenger, Telegram, Slack, Skype, WeChat), and the pervasive smartphone adoption, chatbots are on the rise. Chatbot development is usually linked to Natural Language Processing (NLP). However, latest research shows that from the 100 most popular Facebook Messenger chatbots, very few of them use NLP techniques [5]. It is perfectly possible to program useful chatbots that only make use of simple rule-based conversations, basically state machines. This apparent simplicity however, hides some labor-intensive tasks to be considered to assess the real cost of chatbot development. In fact, due to their novelty, there is a lack of research about how to effectively workout a Return-Of-Investment (ROI) plan. This paper brings to the forefront chatbot concerns other than NLP but with a large impact on the ROI. Our aim is to combat some simplistic understandings of what developing a chatbot is [1, 3]. To this end, we report on three chatbots developed during the last two years. We start by presenting these three case studies. These chatbots are available to interact with through Telegram just by search through their names (i.e. @retosmoocsbot, @dawebot, @tensiobot).

## 2 Case Studies

### 2.1 @tensiobot

**Domain:** *@tensiobot* is a chatbot that helps patients control their blood pressure (Fig. 1). Just a week before meeting the doctor, *@tensiobot* asks patients to measure their blood pressure twice a day (the alert times are customizable). Once the user gets the alert, they should proceed to use the tensiometer and write down the blood pressure values (highest, lowest), answering the chatbot questions. The chatbot supports detection and fixing wrongly-typed values as well as the option to watch a video about how to correctly use the tensiometer. It also allows to present an evolution graph for all the blood pressure values stored. *@tensiobot* has been designed with the help of a doctor and a nurse from the Basque public health system. It is currently used in a one-year long controlled trial.

**Motivation:** Despite its importance for detecting health conditions, some patients don't adhere to a proper blood pressure control when using traditional methods. A chatbot could alert the patient when blood pressure needs to be measured. Moreover, it also collects data that can later be accessed by the doctor (Table 1).



**Fig. 1.** Blood pressure measures are recorded and showed in a figure on demand. The figure is generated through *R* and the *ggplot2* package

**Table 1.** Tensiobot. Main features

Functional dimension
Embedding how-to-proceed videos
Collecting blood-pressure values
Detecting and fixing incorrectly typed blood pressure values
Displaying <i>did-you-know</i> kind of tips periodically
Message customizability (for tips and conversations)
Integration dimension
Delivering daily alerts to patients
Customizable alert times
Set (and remind) schedule time for next doctor visit
Analytics dimension
Display blood pressure evolution graphs

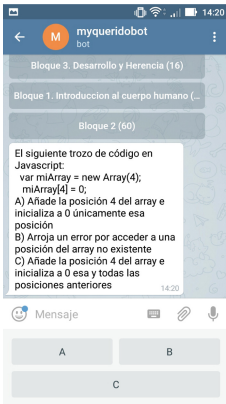
### 2.2 @daweobot

**Domain:** *@daweobot* is a bot for training students using multiple choice question quizzes. It was evaluated in a 15 week long subject with 23 students of a Computer Science subject. The interaction is set in terms of quizzes. The chatbot

displays the name of the quiz (related to any area of the subject) and the number of questions contained in that quiz. The student next selects the quizzes and the first question shows up. of the selected quiz, the keyboard will be adjusted to show just the buttons for the answers at hand (see Fig. 2). The student clicks on any of these buttons, and the bot immediately gives feedback. Gamification techniques are used to honor different degrees of participation and quizz success. At any time, students can request an appointment with the lecturer. This makes @dawebot find a free slot on the lecturer’s agenda using Google Calendar’s API. The appointment will be recorded and the lecturer will be notified via email, automatically updating his/her calendar.

**Motivation:** Some of our students spend considerable time commuting. Through gamification techniques, we wanted to push students to use this commuting time to play with @dawebot (Table 2).

**Table 2.** @dawebot. Main features



**Fig. 2.** Response Keyboard is adjusted to show just the buttons for the answers at hand

<b>Functional dimension</b>
Show the list of themes for a given subject
For each theme, ask the student multiple-choice questions that should be answered just clicking the correct-answer button
Allow to include pictures in the questions
Show explanations about the correct and wrong options
<b>Integration dimension</b>
Allow to make an appointment with the teacher
Customizable messages (for questions and answers)
<b>Analytics dimension</b>
Track the evolution of each student
Show the number of correctly and incorrectly answered questions
Assess question difficulty based on answer and time spent
Show student evolution in terms of quizz progression and results

**2.3 @retosmoocbot**

**Domain:** @retosmoocbot is a chatbot that dare students with question related to an online MOOC (Fig. 3). The challenge rests on answers to be recorded using voice messages. Once recorded, the chatbot distributes the answers to other peers. Evaluation is conducted through rubric supported as a chat conversation: challenges ask the evaluator to rate in a 1–10 scale different aspects of the recording.

**Motivation:** MOOCs need to handle hundreds of students. This introduces important scalability challenges at evaluation time. Traditionally, peer-to-peer evaluation is used. We wanted to test whether using students' voice for answering was a viable option that could improve the motivation of students (and thus, lower the typical MOOCs' very high drop rates) (Table 3).



**Fig. 3.** @retosmooobot allows to record voice-messages that will be evaluated by other peers.

**Table 3.** @retosmooobot. Main features

Functional dimension
A Voice-recording peer evaluation chatbot
Distribute questions that should be voice-answered among the MOOC students (near 700 students)
Record and store the answers (voice-recorded messages)
Assign peer evaluators and distribute the voice-based answers to be evaluated
Help the evaluator to assign grades to voice-based answers using a 1–10 scale and different evaluation criteria
Integration dimension
Customizable messages (for questions and answers)
Alert students when new questions/answers show up
Analytics dimension
Show teachers who is answering&evaluating and who is lagging behind

### 3 Beyond the Conversation

Chatbots can come in two main different flavours as for the conversation user interface: programmed scripts versus NLP. Chatbots that follow programmed scripts have limited conversational scope because they follow only predetermined paths. However, script-based chatbots can go a long way. Despite the strong coverage on AI-powered chatbots, other dimensions can turn out to be more influential on chatbot adoption. Specifically, our case studies provide insights into four of these dimensions.

**The interaction dimension** refers to the way of interaction between the user and the chatbot. Through bot mock-designing applications [2], developers should design the conversation script. In addition, different interaction means should be weighted: getting information from the user (response or inline buttons, text commands, audio messages) and displaying content to the user (text, pictures, video, links, carousels, buttons also.) [4].

**The integration dimension** is concerned with the ecosystem in which the chatbot is going to be deployed. The backoffice still matters, and the chatbot should smoothly interact with the different resources to account for a seamless user experience. Bot developers frequently needs to face integration concerns for both databases and API integration. The former is needed to store the state and context as well as the history and user interactions. In general, it is not rare for bots to interact with external systems through REST or GraphQL APIs. We should know how to hook your bot business logic with this external services. This might include tasks related to configuration management like how to store credentials (login, password, API tokens...) both for testing and deployment stages, or how to dynamically synthesize natural language user expressions into API invocations [8].

**The Analytics Dimension.** Chatbots' stakeholders include users and trackers. The former refers to those who directly interact with the chatbot. By contrasts, trackers do not use directly the chatbot but need to monitor chatbot usage. Some examples follows: in @dawebot, students are the users while lecturers are the trackers; in @tensiobot, patients are the users while practitioners need to track the results. Users and trackers differ not only in how the interact with the chatbot but also the granularity at which information is provided. This is similar to the database world and the difference between transactional systems and data warehouses. Developers might need to face this dimension though analytics and control panels. In chatbot development the term analytics is usually linked to system that get metrics about how users interact with the bot (ranking of most used commands, user segmentation, statistics...). Telegram does not offer a good, simple and accessible analytics service so we should resort to third-party systems (like botan.io) or create our custom control-panel.

**Quality Assurance Dimension.** Perfective and Corrective maintainability are also present in chatbot development. Developers need to account for testing environments. Testing functional requirements in chatbots presents some challenges due to the lack of a linear input. This means that when using a chatbot the user can enter literally any string or voice command, and the chatbot should answer in a reasonable manner. From a testing perspective, this is very difficult to test and 100% coverage is impossible. As for non-functional requirements, the promptness of the chatbot's answers is a critical metric. Some of these metrics could be easily tracked using some chatbot testing emerging libraries [6] but still there is a nascent industry here.

Figure 4 collects how these different dimensions have been addressed for the case studies. Most importantly, each table cell introduces an estimate of the effort involved for each dimension in terms of hours invested in design/development.

	<u>Integration</u>		<u>Interaction</u>	<u>Analytics</u>		<u>Quality assurance</u>
<u>BOT</u>	<u>DB</u>	<u>API</u>	<u>UI Design</u>	<u>Usage</u>	<u>Control Panels</u>	<u>Testing of Functional and not functional requirements</u>
@ D A W E B O T	Need to generate custom tables for storing questions, answers and user performance data (3h)	Integration with Google Calendar's API to allow the student to make an appointment and with Plotly API to get graphs about students performance (10h)	Inline buttons for quiz selection (there were many of them) and response buttons for easy navigation between multiple-choice-questions possible answers. (20h)	Used botan.io to know more about how users use the bot's commands (4h)	Used for easy addition of new questions and answers (5 h)	Custom tests with PHPUnit (10 h)
@ T E N S I O B O T	Need to generate custom tables for storing blood pressure measures and scheduling alerts. (3h)	Integration with R and specifically with the ggplot2 package to generate great looking blood pressure measure graphs (20h)	Response buttons for easy access to main features, simple texts for input data. Emojis selection (10h)	Created a custom panel and administration commands so the doctor can easily check each patient's introduced data (5h)	A Django/Python custom control panel for easy edition of strings used by the bot in conversations with the patient (8h)	TDLlib for thorough testing of conversations and UI elements.. Tgcli for testing basic text-based commands TDLlib and ad-hoc code for testing response times (20h)
@ R E T O S M O O C S B O T	Special tables for storing metadata of audio files and peer-assigned grades (5h)	Integration with ffmpeg library for audio transcoding (raw audio recordings were not playable in iOS without transcoding them to mp3) (10h)	Used audio recording and playing features and simple text for easy typing of numeric input values (8h)	Created a custom panel and administration commands so teachers can easily check who has interacted with the bot and how many and which exercises did (10h)	Special ad-hoc control panel for assigning peer evaluators to audio recordings (15h)	Due to time constraints we had to resort to manually test the commands and conversations (4h)

Fig. 4. Concerns risen during the development of the chatbot case studies, classified along dimensions. Each cell details an estimation of number of hours invested in each task

## 4 Conclusions

We report on chatbots developed for three different domains: online regular teaching, massive online teaching and health related subjects. Specifically, we

enumerate different activities which are arranged them along four main dimensions: interaction, integration, analytics and quality assurance. The take-away message is that there is more than NLP to chatbot development. Chatbot conversation-like interaction might simplistically lead to believe that chatbot development is easy. This might be true in some straightforward scenarios, but integration, testing and other concerns lurk in the backend. Before initiating a chatbot adventure, developers should have a holistic view of the different dimensions chatbot development involves. To this end, Web Engineering methodologies should percolate also the chatbot world. Do not let their simple interfaces mislead you.

## References

1. Beck, B.: How to Build Your Own Facebook Chatbot in About 10 Minutes, March 2018. <https://www.clearvoice.com/blog/build-facebook-chatbot-10-minutes/>
2. Botsociety: Conversational design and prototype - Botsociety. <https://botsociety.io>
3. Hossain, M.: Create your own Viber Chatbot in Minutes, November 2017. <https://chatbotslife.com/create-your-own-viber-chatbot-in-minutes-with-zero-coding-1a622accedcc>
4. Klopfenstein, L.C., Delpriori, S., Malatini, S., Bogliolo, A.: The rise of bots: a survey of conversational interfaces, patterns, and paradigms. In: Proceedings of the 2017 Conference on Designing Interactive Systems, DIS 2017, pp. 555–565. ACM, New York (2017). <https://doi.org/10.1145/3064663.3064672>
5. Pereira, J., Díaz, O.: A quality analysis of Facebook Messenger’s most popular chatbots. In: Proceedings of Symposium on Applied Computing, pp. 1–8. ACM, Pau (2018). <https://doi.org/10.1145/3167132.3167362>
6. TestMyBot: testmybot: Automated Testing for Chatbots, March 2018. <https://github.com/codeforequity-at/testmybot>. Original-date: 2017-03-08T22:12:19Z
7. Weizenbaum, J.: ELIZA—a computer program for the study of natural language communication between man and machine. *Commun. ACM* **9**(1), 36–45 (1966)
8. Zamanirad, S., Benatallah, B., Chai Barukh, M., Casati, F., Rodriguez, C.: Programming bots by synthesizing natural language expressions into API invocations. In: Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering, pp. 832–837. ASE 2017. IEEE Press, Piscataway (2017). <https://doi.org/10.1109/ASE.2017.8115694>