2021 IEEE/ACM 43rd International Conference on Software Engineering: Companion Proceedings (ICSE-Companion) | 978-1-6654-1219-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICSE-Companion52605.2021.00138

The Software Challenges of Building Smart Chatbots

Gwendal Daniel
Internet Interdisciplinary Institute
Universitat Oberta de Catalunya
Barcelona, Spain
gdaniel@uoc.edu

Jordi Cabot
ICREA
Universitat Oberta de Catalunya
Barcelona, Spain
jordi.cabot@icrea.cat

***Abstract*—Chatbots are becoming complex software artifacts that require a high-level of expertise in a variety of technical domains. This technical briefing will cover the software engineering challenges of developing high-quality chatbots. Attendees will be able to create their own bots leveraging the open source chatbot development platform Xatkit.**

I. The Rise and Importance of Chatbots

Instant messaging platforms have been widely adopted as one of the main technologies to communicate and exchange information [1]. Nowadays, most of them provide built-in support for integrating *chatbot applications*, which are automated conversational agents capable of interacting with users of the platform [2]. Chatbots have proven useful in many other contexts to automate tasks and improve the user experience, such as automated customer services, education, and e-commerce. Moreover, existing reports emphasize that chatbot design will become a key ability in IT hires in the near future [3]. The global chatbot market is projected to reach 2 billion dollars by 2024, growing at a CAGR (compound annual growth rate) of 29.7% [1].

This widespread interest and demand for chatbot applications has emphasized the need to be able to quickly build complex chatbot applications supporting AI-based natural language processing [4] in order to be able to fluently chat with the user. Furthermore, any non-trivial chatbot requires accessing an orchestration of internal and external services in order to perform the requested user actions (e.g. to check and query the data to be served back to the user or to actually execute some processes/actions in response).

As such, chatbots are becoming complex software artifacts that require a high-level of expertise in a variety of technical domains, ranging from natural language processing to a deep understanding of the APIs of the targeted instant messaging platforms and third-party services to be integrated.

II. Software Engineering for Chatbots

We believe a hands-on session would be of interest to both practitioners and researchers of the ICSE community.

First because chatbots are, more and more, part of new software development projects, e.g. as a complement to a "standard" GUI for the system. Therefore, we should all have a basic understanding of the main steps and obstacles in building a good chatbot.

Secondly, because as a complex artefact in itself, there is plenty of software engineering techniques that could be applied to improve chatbots development[2]. Testing would be just one of the many examples. As such, we believe that learning more about the process of chatbot development would help attendees to identify good research opportunities in this very important (at least from a social and economical perspective) topic.

III. Speakers bio

The Hands-on session will be conducted by Jordi Cabot and Gwendal Daniel.

Gwendal Daniel is a post-doctoral fellow in the SOM Research Lab at Internet Interdisciplinary Institute (IN3), a research center of the Universitat Oberta de Catalunya (UOC). He received his PhD degree in 2017 in the AtlanMod Team, at the Ecole des Mines de Nantes, France. His research interests include Model Driven Engineering, Model Persistence, Query, and Transformation techniques, Domain Specific Languages, as well as applying model-based techniques for large-scale data applications.

Jordi Cabot received the B.Sc. and Ph.D. degrees in computer science from the Technical University of Catalonia. He was a Leader of an INRIA and LINA Research Group at Ecole des Mines de Nantes, France, a Post-Doctoral Fellow with the University of Toronto and a Visiting Scholar with the Politecnico di Milano. He is currently an ICREA Research Professor at Internet Interdisciplinary Institute. His research interests include software and systems modeling, formal verification and the role AI can play in software development (and vice versa).

Jordi and Gwendal have been working on a chatbot platform, Xatkit [5][3], for the last two years. Xatkit has evolved from an open source research prototype to a university spin-off. As such, Jordi and Gwendal know rather well the main players in the (chat)bot landscape and the key challenges when

[1] https://www.alliedmarketresearch.com/chatbot-market

[2] And related fields such as conversational user interface design

[3] https://xatkit.com/

978-1-6654-1219-3/21/$31.00 ©2021 IEEE
DOI 10.1109/ICSE-Companion52605.2021.00138

building chatbots for a variety of scenarios. For instance, you can see some reflections in this invited talk[4].

IV. Covered topics

The introduction before the hands-on part of the session will focus on: i) Clarify the terminology around chatbots, voicebots, conversational interfaces, etc ii) Illustrating that building a good chatbot requires a combination of a large number of software components, iii) Explaining typical platforms and (NLP) strategies of chatbot platforms and iv) Describing the different types of bots one could build and the key aspects to take into account for each one.

V. Creating your own bots

A. Exercise details

The hands-on part of the technical briefing will be organized in two subsections:

- **Predefined exercises (75%)**: a series of bots to build where each exercise focuses on a different dimension of the bot building process.
- **Community-built bot (25%)**: an interactive session where attendees will specify a bot to build and start its implementation.

B. Predefined exercises

We will start the predefined exercises session with a classical *Hello World* chatbot that replies to users greetings. While simple, this initial exercise emphasizes some **core aspects** of chatbot development: on which platform should the bot listen and reply? what is a user *intention* and how to train the bot to recognized it? what should the bot do once it has answered the user? what if the bot doesn't understand the user input?

Then, we will start building on top of this initial bot and we will introduce the concept of **entity extraction** and how to use it to tune our bot with personalized messages. This exercise will emphasize the need to carefully craft training sentences for optimal results, and will introduce powerful NLP engines such as DialogFlow[5] and NLP.js[6]. We will also show how different entities can be used to extract city names, dates, or custom domain-specific information.

We will then extend our bot to integrate **sentiment analysis**, and adapt its answers to how the user is feeling. This part will showcase complementary natural language understanding solutions such as Apache OpenNLP[7] and HuggingFace[8], and emphasizes the need to integrate multiple approaches to create advanced bots.

The fourth exercise focuses on integrating **complex processing** in a bot, such as information retrieval and formatting. To do so we propose to develop a bot able to print the weather forecast for a given location by querying the OpenWeatherMap's REST API[9]. This exercise stresses the fact that chatbots are first-class citizens of complex software systems, and need to be designed accordingly.

We will then discuss how bots can subscribe to **external events**. In this exercise we propose to create a bot that listens to Github push notifications (e.g. when an issue is opened or closed) and reacts to them by 1) providing feedback to the contributor by checking that the issue follows some basic formatting guidelines, and 2) alerting the software maintainers on Slack that a new issue has been created.

Finally, we dedicate an exercise to additional tooling aiming to ease the development and maintenance of complex bots. We will specifically focus on the need for **continuous integration and deployment**, and discuss why, when, and how to test chatbots. Practically, attendants will setup and run a test suite for the previously showcased bots. We will close the session with a discussion on the interest of monitoring deployed bots and detect issues at run-time.

Note that for some of the bots we will provide a partial skeleton to be able to focus on the specific learning aspect of the exercise.

C. Community-built bot

This interactive session aims to gather concrete requirements from the audience and start building a bot to address them. The predefined exercises presented above can be used to kick-off the discussion and identify interesting features to explore. We will assist the process by providing recommendations and insights on the key elements to keep in mind while developing a complex bot. If applicable, we will also illustrate the discussion with enterprise-ready bots we have built.

D. Showcased Technologies

All bots will be built using Xatkit, an open source chatbot development platform. Continuous integration and testing aspects will be illustrated with Botium[10].

However, attendees will be able to easily transfer their learning experience to other chatbot platforms, because Xatkit works as an orchestration layer combining existing NLP solutions instead of reinventing the wheel.

References

[1] B. Nardi, S. Whittaker, and E. Bradner, "Interaction and Outeraction: Instant Messaging in Action," in *Proc. of the 3rd CSCW Conference*. ACM, 2000, pp. 79–88.

[2] L. Klopfenstein, S. Delpriori, S. Malatini, and A. Bogliolo, "The Rise of Bots: a Survey of Conversational Interfaces, Patterns, and Paradigms," in *Proc. of the 12th DIS Conference*. ACM, 2017, pp. 555–565.

[3] G. Inc., *The Road to Enterprise AI*. RAGE Frameworks, 2017.

[4] P. Jackson and I. Moulinier, *Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization*. John Benjamins Publishing, 2007, vol. 5.

[5] G. Daniel, J. Cabot, L. Deruelle, and M. Derras, "Xatkit: A multimodal low-code chatbot development framework," *IEEE Access*, vol. 8, pp. 15 332–15 346, 2020.

[4]https://livablesoftware.com/lessons-learned-building-commercial-oss-bot-platform/
[5]https://dialogflow.cloud.google.com/
[6]https://github.com/axa-group/nlp.js
[7]https://opennlp.apache.org/
[8]https://huggingface.co/
[9]https://openweathermap.org/
[10]https://www.botium.ai/