



**HAL**  
open science

# An Interactive Paradigm for the End-User Development of Chatbots for Data Exploration

Ludovica Piro, Giuseppe Desolda, Maristella Matera, Rosa Lanzilotti, Sara Mosca, Emanuele Pucci

► **To cite this version:**

Ludovica Piro, Giuseppe Desolda, Maristella Matera, Rosa Lanzilotti, Sara Mosca, et al.. An Interactive Paradigm for the End-User Development of Chatbots for Data Exploration. 18th IFIP Conference on Human-Computer Interaction (INTERACT), Aug 2021, Bari, Italy. pp.177-186, 10.1007/978-3-030-85610-6\_11 . hal-04215521

**HAL Id: hal-04215521**

<https://inria.hal.science/hal-04215521v1>

Submitted on 22 Sep 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

# An Interactive Paradigm for the End-User Development of Chatbots for Data Exploration

Ludovica Piro<sup>1</sup>, Giuseppe Desolda<sup>2</sup>, Maristella Matera<sup>1</sup>,  
Rosa Lanzilotti<sup>2</sup>, Sara Mosca<sup>2</sup>, Emanuele Pucci<sup>3</sup>

<sup>1</sup>DEIB, Politecnico di Milano, Italy

<sup>2</sup>Computer Science Department, University of Bari Aldo Moro, Italy

<sup>3</sup>Awby Srl, Florence, Italy

`ludovica.piro@mail.polimi.it, giuseppe.desolda@uniba.it,  
maristella.matera@polimi.it, rosa.lanzilotti@uniba.it,  
s.mosca3@studenti.uniba.it, emanuele.pucci@awby.it`

**Abstract.** This paper presents an interaction paradigm for the design of chatbots. Its novelty is the completion of conversational patterns that progressively guide the design activity and provide an interactive, immediate representation of the conversation under construction. Thanks to the automatic generation of code, the paradigm facilitates the rapid prototyping of the conversational UI, thus it empowers non-programmers to master the design process. The paper also illustrates some preliminary user studies and discusses some lessons learned for the definition of interactive paradigms for the design of conversational UIs.

**Keywords:** End-User Development, Chatbot Design, Conversational UIs.

## 1 Introduction

Chatbots for data exploration support conversations that let the user progressively discover and interactively retrieve data from known data sources [8]. Given the multiple advantages that the literature recognizes to the conversational paradigm for data access, and its current diffusion through different applications, the literature is now posing emphasis on methodologies for chatbot design [10; 21; 26]. A diffused opinion is that domain experts are critical in the development of chatbots; but it also emerges that engaging them in chatbot development is difficult due to their lack of technical competencies. To overcome this drawback, this paper presents an interactive paradigm for the End-User Development (EUD) of chatbots for data exploration.

The new interactive paradigm has been developed on top of CHATIDEA, a methodology for the rapid prototyping of chatbots that is also complemented by a software framework [8]. The paradigm is based on a visual front end that enables non-programmers to complete conversation patterns that mimic the interaction between users and the chatbot under construction. It, therefore, masks the need for technical specifications and allows the designers to manipulate directly the elements of the conversation, also

providing immediate feedback on the designed conversation. This modus operandi helps overcome some complexity factors that, more in general, characterize the design of AI-based systems [26], such as the difficulty of identifying the functionality that the system can afford, as well as the possible output that the system can produce. The proposed paradigm also facilitates sketching and rapid prototyping of the conversational UI, two fundamental activities in interaction design, which allow designers to understand what the technology is and can do, engage in creative thinking, and assess and improve on their designs [10; 26].

After discussing some related works (Section 2), this paper illustrates the interactive paradigm for the EUD of chatbots (Section 3) and the user-centered process adopted to assess the adequateness of the paradigm with respect to the expectations of both chatbot developers and non-expert programmers (Section 4). The paper then ends by discussing some lessons learned (Section 5) and by outlining our future work (Section 6).

## 2 Rationale and Background

The idea of computers behaving like humans dates back to 1950, when Alan Turing proposed his famous test [23]. However, it is from the 90s that several applications benefitted from the advancements in artificial intelligence, natural language processing and speech recognition: the applications became more and more intelligent, capable of better understanding various conversations and performing more complex tasks. With the smartphone era, the technology witnessed an explosion of commercial applications: some examples are IBM Watson [14], Siri [3], WeChat [24], Alexa [1]. If we want to classify those applications from a high-level perspective, we can distinguish between *task-oriented* and *conversational chatbots* [9], the former being oriented toward a resolution of a specific task and the latter designed to carry on a general conversation.

Together with the technology, tools and frameworks to support the creation of conversational interfaces evolved themselves. Such frameworks and tools target both experienced and non-experienced users, trying to ease the process of chatbot creation. The majority of those tools help the user in two fundamental operations: *intent matching* (understanding the action the user wants to perform and match it with an appropriate response) and *entity extraction* (the ability to extract key elements from an utterance). The evolution of the tools, however, has not changed the underlying mechanism of a chatbot creation: either the designer or the developer has to handle from scratch all the questions and answers. Also, it seems to be missing a general approach to develop data-driven chatbots. Even though some tools let the user retrieve information from spreadsheets [6] or the internet thanks to webhooks, mapping intents and entities and developing conversational interfaces using a database as a source is still scarcely explored.

Researchers have been interested in using natural language to access large databases long before chatbots. The use of natural language can help users without formal knowledge of a query language access and perform actions on a structured database [2]. LUNAR, developed in 1972, was one of the first examples of natural language interfaces created to access information structured in a database [25]. One more recent example is NaLIR [16]: it takes complex sentences from the user and generates a query

in a technical language following a “human-in-the-loop” approach that asks the user to check the results of the intermediate generation steps. There are not many examples of explorative chatbots used to access databases yet: one of the few we found is Intellibot, a dialogue-based chatbot for the insurance industry [20]. However, Intellibot is still a custom-made chatbot, developed for a domain-specific conversation.

This paper addresses the gap by proposing a visual paradigm that facilitates the design of conversational interfaces for database exploration. The paradigm falls in the category of the End-User Development (EUD) tools [17], which empower non-technical users, e.g., domain experts, to achieve goals for which computing knowledge is needed. Some chatbot frameworks included similar environments that help the user define flow charts or configure the interaction through specific graphic user interfaces (e.g., Dialogflow [11] and Motion AI [13]). However, our approach differs from the previous because it adopts a dynamic interactive paradigm, which is made possible by the capability of the framework to automatically generate the conversation.

### 3 An interactive paradigm for the EUD of chatbots

The contribution of this paper is a new interactive paradigm for the EUD of chatbots for data exploration. The resulting solution is a Web front end built on top of CHATIDEA, a software framework for the rapid prototyping of chatbots for data exploration [8]. The following sections provide details on the CHATIDEA modeling abstractions and the new EUD paradigm.

#### 3.1 Modeling Abstractions

CHATIDEA automatically generates a chatbot starting from a dump of a relational database (DB) and a JSON descriptor file. The dump provides the data that can be explored by the users by means of the resulting chatbot, and the definition of the schema for the organization of data. The descriptor contains a set of *annotations* identifying key data elements and properties of the DB, which are relevant for managing conversations for data exploration. By combining the DB dump and the annotation descriptor, CHATIDEA automatically generates the chatbot dialog system, according to conversational patterns that progressively guide the users to explore the data.

One fundamental annotation refers to the *table role*, which can be Primary, Secondary, or Crossable. *Primary tables* represent data that can be queried directly in the chatbot. *Secondary tables* store data dependent on other entities, which are interesting for the conversation only when reached from other (primary) tables. *Crossable tables* represent many-to-many relationships; within the chatbot, they are represented as links to navigate between different entities - while no direct or deferred search is allowed on them. Among others, some annotations can rename tables and attributes by using *names and aliases* that are simpler and more understandable during the conversation. It is possible to specify which attributes can be used in user utterances to *filter the table instances*, which *display attributes* have to be shown when instances are retrieved by the chatbot, which attributes can be used to *categorize primary table instances and display*

*aggregated visualizations*. Lack of space prevents us from fully describing the annotations; a detailed description of the whole set of annotations needed to automatically generate a chatbot with CHATIDEA is reported in [8]. An example of annotations expressed in JSON format is available at <https://bit.ly/3x7N6ov>.

### 3.2 The Interactive Design Paradigm

Even if the CHATIDEA software framework makes it possible to create chatbots without programming, the designers are still required to manually write descriptor files using technical specification languages (e.g., JSON); this activity is time-consuming, error-prone, and requires complete knowledge of the technical syntax.

The new interactive paradigm, which is the main contribution of this article, aims to enable even people without expertise in programming and/or chatbot development to *visually complete* conversation patterns related to the interaction between users and the chatbot under construction. The chatbot designer is guided through the process by means of text prompts modeled on a hypothetical conversation that the final user may have with the chatbot. The text prompts are skeletons of training phrases to be completed with keywords and data taken from the database. This helps designers implicitly create the annotation schema, i.e., map DB data elements on conversational elements, within the context of a sample conversation based on patterns for DB navigation.

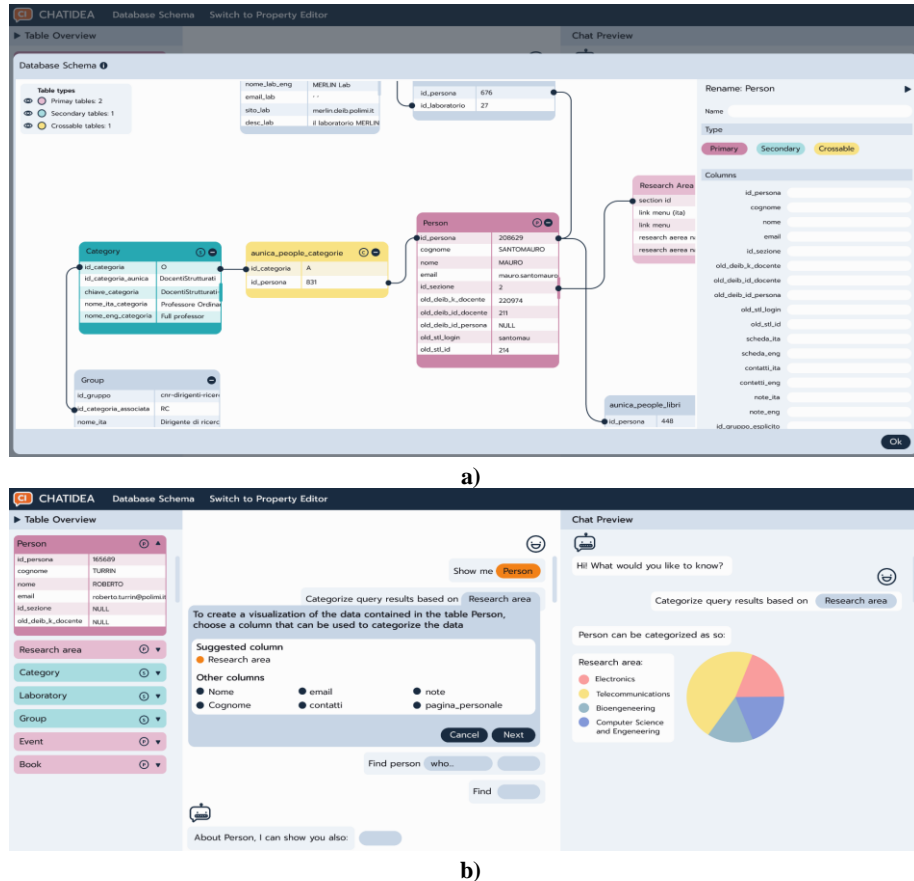
The visual front end enabling this paradigm consists of two main areas. After uploading a DB dump, in the first area (Figure 1a) the designer can skim the DB by operating on a graph-based visualization of tables and relationships. In a sidebar editor, the designer assigns table roles (primary, secondary and crossable, represented through different colors), and edits table and attribute names if needed to improve the conversation design. Moving to the next area, for each primary and secondary table tagged in the previous section, the designer completes some conversation patterns. For example, for the table *Person*, Figure 1b illustrates the pattern “*Categorize query results based on <a table attribute>*”. Its completion consists in selecting a categorical attribute, among the ones included in the selected table, which will be used to produce a visualization that categorizes the table data. The “*Research area*” part of the sentence is initially empty; when the designer clicks the empty label, a pop-up window asks to select the categorical attribute. In this example, the designer has selected *Research area*.

A characterizing feature is that a live-preview in the right-hand panel shows the effect of any design choice on a sample of instances retrieved from the DB. In the example, the designer can see that the response to the configured utterance is a pie chart that categorizes the *Person* instances according to four different research areas. The designers can thus control the outcome of their design decisions. A video demonstrating the visual paradigm is available at: <https://youtu.be/hP9IDyJnRG4>.

## 4 Evaluation

Two qualitative formative studies were conducted to enhance the quality of some initial, high-fidelity prototypes of the visual front end. A user study was then carried out

on an advanced Web-based prototype, to investigate the usability and the perceived workload. Due to the COVID-19 pandemic, the studies were carried out online, using Skype or WebEx, depending on the users' familiarity with these teleconferencing platforms. Each session was video-recorded. Each participant was introduced to the study purpose, informed on what to do, and signed a consent form.



**Fig. 1.** Two snapshots of the CHATIDEA visual front end: a) the area for annotating table roles; b) the area for completing conversation patterns.

#### 4.1 Preliminary Evaluation: Validity and Usefulness

A first formative study was organized to receive feedback on the validity of the interaction paradigm from users who had already performed annotation-related tasks with CHATIDEA using the JSON syntax. A first prototype was evaluated by interviewing 4 users who had participated in the development of the CHATIDEA software framework, being former students in Computer Science and Engineering who had worked on CHATIDEA for their master theses. After an initial demonstration of the prototype by one of the researchers moderating the interview, a discussion followed to gather the

developers' opinions. They found the UI complete with respect to the design capabilities offered by the software framework, and considered the interaction paradigm clear and effective. They also helped identify some improvements that were mainly related to the table tagging mechanisms on the first page. For example, they thought of expressing table annotations by "manipulating" directly the table representation, and not by using the right-hand panel. They also observed that the system could take some initiatives and guide the design through recommendations that could be derived from an analysis of the database schema. For example, they suggested that only tables with at least two key attributes can be tagged as crossable and this feature should be highlighted. They also observed that some explanatory messages appeared to be too vague.

After fixing the problems identified in the first evaluation session, the prototype was shown to an expert of chatbots, one of the founders of the *AwWhy* company (<https://www.awwhy.it/>) specialized in the development of chatbots for customer care. The focus of this second evaluation was on the acceptability and usefulness of the CHATIDEA framework and its visual paradigm. The interviewee did not point out any particular problem with the interaction paradigm and expressed that there is a diffused need to have tools for configuring chatbots for data exploration, especially in an enterprise. He only remarked that the annotation procedure might be hard to complete for huge databases.

## 4.2 Usability test

A third study was finally performed to evaluate the usability of CHATIDEA and the effectiveness of the interaction paradigm. The thinking aloud protocol was adopted. Users were asked to use a refined prototype of CHATIDEA, to design a chatbot for a reduced version of the DB of the Web site of the Department of Electronics, Information and Bioengineering at Politecnico di Milano. The DB consisted of 8 tables storing data on the faculty, their awards, the published books, the research projects.

**Participants.** A total of 12 participants (10 M and 2 F; mean age 24.5 years) were recruited. 5 participants were *chatbot developers* working at the *AwWhy* Srl company; 3 participants were students of a Computer Science university curriculum, i.e., *programmers* not expert in chatbot development; 4 participants were *non-programmers*: 1 game designer, 1 student in Bioengineering, 2 students in Communication Design.

**Procedure.** During the video call, every user was asked to activate the webcam and share the screen to let the observer identify facial expressions that could reveal spontaneous feelings and see the actions performed to complete the assigned tasks. Each session was video-recorded. Each user was asked to read a brief description of the functionality supported by the visual front end, in the form of a scenario providing information about a possible context of use. The description did not include details on the underlying software framework and its modeling abstractions. The general organization of the front end was explained in a short demo. The user was then provided with a sheet reporting 7 tasks and started the execution of each task. The first two tasks were related to table tagging, while the remaining five tasks covered each of the other annotation concepts. At the end of all the tasks, the participant filled in an online form with the SUS [5; 15] and NASA-TLX [12] questionnaires to be answered one after the other.



**Results.** Quantitative and qualitative data were collected. For quantitative data, the SUS and NASA-TLX scores were computed. SUS estimated the *system usability perceived by users*, as resulting from two factors, *System Learnability* (statements #4 and #10) and *System Usability* (the other 8 statements) [15]. The average SUS score was  $\bar{x}=74.2$ ,  $SD=15.0$ ; the System Usability score was  $\bar{x} = 76.3$ ,  $SD = 14.8$ ; the System Learnability score was  $\bar{x}=68.8$ ,  $SD=17.3$ . Due to the limited number of participants, it was not possible to compute inferential statistics for the comparison of the three user groups, but the means and the standard deviations were very similar (*chatbot developers*:  $\bar{x}=74.4$ ,  $SD=15.2$ ; *programmers*:  $\bar{x}=73.8$ ,  $SD=14.5$ ; *non-programmers*:  $\bar{x}=74.4$ ,  $SD=19.5$ ).

NASA-TLX was used to measure the *perceived workload* on a scale from 0 to 100, where 0 equals to low effort, 100 to excessive effort from the user. The average score was  $\bar{x} = 38.1$ ,  $SD = 13.3$ . On average, the highest score, and therefore the poorer performance, was recorded for the *Mental Demand dimension* ( $\bar{x} = 54.2$ ,  $SD = 17.3$ ), followed by *Effort* ( $\bar{x} = 44.2$ ,  $SD = 20.7$ ), *Performance* ( $\bar{x} = 37.5$ ,  $SD = 16.6$ ), *Temporal Demand* ( $\bar{x} = 31.7$ ,  $SD = 24.8$ ), and *Physical Demand* ( $\bar{x} = 23.3$ ,  $SD = 19.7$ ). In this case as well, the score means for the three groups are very similar (*chatbot developers*:  $\bar{x} = 39.6$ ,  $SD = 13.3$ ; *programmers*:  $\bar{x} = 39.6$ ,  $SD = 6$ ; *non-programmers*:  $\bar{x} = 35.6$ ,  $SD = 19.1$ ), with a lower cognitive load for non-programmers.

The qualitative analysis was performed on the participants' comments [4]. Few themes, not reported here for brevity, were related to very specific problems with few visual elements (e.g., expressiveness of tooltips). The three themes reported in the following, instead, convey more general reflections on the interactive paradigm.

**Theme 1: Conflicts between system architecture and user assumptions.** In relation to the annotations on the database schema, users agreed that knowing basic concepts of relational databases helps complete correctly the tasks ("The design is for sure aided by the system, but requires knowledge of the subject [database domain] and a bit of technical knowledge [relational database concepts]"). These concerns did not emerge, however, during the completion of conversation patterns.

**Theme 2. Perceived ease of use.** Despite their difficulties in executing some tasks, most participants reported the ease of use of setting and modifying conversation patterns, and noted that the workload is not overwhelming ("Very usable, [the tasks] flow well"). As also highlighted by the questionnaires, this perception was not influenced by the user programming skills, as the scores were comparable among all the three groups.

**Theme 3: Generating chatbots from databases.** Participants appreciated the flexibility offered by the prototype through table tagging ("It is particularly powerful that the designer has control on what tables to include and what relationships to use"), and the opportunity to configure user queries easily ("It is nice to design database queries through natural language"). Chatbot experts also asked for advanced features to be invoked on-demand, even by coding, for a deeper customization of the intents.

## 5 Discussion

Besides highlighting usability problems, the insights gained through the user studies brought to light general key points, which are in line with previous findings on EUD, but add interesting perspectives on interactive paradigms for conversation design.

**Closeness of mapping.** For the effectiveness of an EUD paradigm, it is important to adopt a representation of possible design choices that abstracts from the technical details, also giving immediate feedback on the results [7; 19]. As highlighted by Theme 2, directly manipulating utterances helped the designers understand the conversation patterns that the system can manage, and their effect on the conversation flow [10; 26]. Theme 1 and 3 in a sense confirm this assumption, as they suggest that, even if the users did not mind selecting entities by operating on database tables, they felt more comfortable when manipulating the utterance structure.

**Control on the conversation context.** Theme 2 suggests that having a preview on how the sentences will appear during conversation and how they relate to one another, being able to always modify previous annotations and see the consequent effect, were key features for the ease of the interactive paradigm. This is also in line with the results of previous works that studied the effect of immediate representations of the design choices [7; 19]. Looking at the general class of AI-based interactive systems, this feature can give the designer control over the outcome of AI models, and provides a lens for understanding AI’s challenges in the design of interactive systems [10].

**Assistance on intent and entity identification.** Theme 3 suggests that, despite the difficulties in dealing with database concepts, the users appreciated having at their disposal tables to represent the entities that could be covered by the conversation, and the flexibility of filtering the most relevant ones. A similar observation applies to the configuration of intents, as designers operate on a set of intents suggested by the system as representative of classes of queries for data exploration. This guidance can help conceptualize the dialogue system capabilities and choreograph the interactions [10; 26].

**Accommodating different levels of control.** Some facets of Theme 3 suggest that it is also important to enable the most skilled designers to extend and personalize the predefined patterns, to ensure a “gentle slope of difficulty” [18]. The designers could be enabled to act also on the modeling abstractions, for example by defining new annotations with impact on the underlying model for dialog generation. These mechanisms can be considered further ingredients to give designers control on AI models.

## 6 Conclusions and future work

This paper has presented a new interactive paradigm for the design of chatbots for data exploration. The preliminary evaluation conducted so far has some limitations (e.g., the limited number of users, lack of comparison with other design paradigms, limited number of DB tables in comparison to realistic scenarios). However, it highlights that the paradigm has some potential to bring conversation design within the reach of domain experts who don’t have programming knowledge, and also to fasten chatbot development for programmers. However, as also evident from the problems identified through the user studies, some aspects still remain open; our future work will focus on them. At a more general level, an important aspect concerns the control that designers should have on Human-Centered AI [22]. In this respect, our future work will focus on generalizing our research to consider the changing role of humans in the design of AI-based systems and foster a discussion on design practices for human-AI interaction [10].

## References

1. Amazon. *Amazon Alexa*. Retrieved from <https://www.amazon.it/b?ie=UTF8&node=15752736031> Last Access April 10, 2021.
2. Androutsopoulos, I., Ritchie, G.D. and Thanisch, P. (1995). Natural language interfaces to databases-an introduction. *arXiv preprint cmp-lg/9503016*.
3. Apple. *Apple Siri*. Retrieved from <https://www.apple.com/it/siri/> Last Access April 10, 2021.
4. Braun, V. and Clarke, V. (2006). Using Thematic Analysis *Psychology Qualitative Research in Psychology*, 3(2), 77-101.
5. Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability evaluation in industry*, 189(194), 4-7.
6. Canh, N.T. *Turn your database into a chatbot*. Retrieved from <https://medium.com/botfuel/turn-your-database-into-a-chatbot-10dae003b97d> Last Access April 10, 2021.
7. Capiello, C., Matera, M. and Picozzi, M. (2015). A UI-Centric Approach for the End-User Development of Multidevice Mashups. *ACM Transaction Web*, 9(3), 1-40.
8. Castaldo, N., Daniel, F., Matera, M. and Zaccaria, V. (2019). Conversational Data Exploration. In *Proc. of the International Conference on Web Engineering (ICWE '19)*. Springer International Publishing, 490-497.
9. Chen, H., Liu, X., Yin, D. and Tang, J. (2017). A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter*, 19(2), 25-35.
10. Feine, J., Morana, S. and Maedche, A. (2020). Designing Interactive Chatbot Development Systems. In *Proc. of the Making Digital Inclusive: Blending the Local and the Global (ICIS '20)*.
11. Google. *Dialogflow*. Retrieved from <https://dialogflow.com/> Last Access March 14th, 2021.
12. Hart, S.G. and Staveland, L.E. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Advances in psychology*, 52, 139-183.
13. HubSpot. *motion.ai*. Retrieved from <https://www.motion.ai/> Last Access March 14th, 2021.
14. IBM. *IBM Watson*. Retrieved from <https://www.ibm.com/watson/how-to-build-a-chatbot> Last Access April 10, 2021.
15. Lewis, J. and Sauro, J. (2009). The Factor Structure of the System Usability Scale. In: M. Kurosu Ed. *Human Centered Design - HCD 2009*. LNCS, Vol. 5619, Springer Berlin Heidelberg, 94-103.
16. Li, F. and Jagadish, H.V. (2014). NaLIR: an interactive natural language interface for querying relational databases. In *Proc. of the International Conference on Management of Data (SIGMOD '14)*. Association for Computing Machinery, 709-712.
17. Lieberman, H., Paternò, F., Klann, M. and Wulf, V. (2006). End-User Development: An Emerging Paradigm. In: H. Lieberman, F. Paternò and V. Wulf (Eds.) *End User Development*. Human-Computer Interaction Series, Vol. 9, Springer Netherlands, 1-8.
18. Lieberman, H., Paternò, F. and Wulf, V. 2006. *End User Development*. Springer.
19. Namoun, A., Nestler, T. and De Angeli, A. (2010). Conceptual and Usability Issues in the Composable Web of Software Services. In: F. Daniel and F. Facca (Eds.) *Current Trends in Web Engineering - ICWE 2010 Lecture Notes in Computer Science*, Vol. 6385, Springer Berlin Heidelberg, 396-407.
20. Nuruzzaman, M. and Hussain, O.K. (2020). IntelliBot: A Dialogue-based chatbot for the insurance industry. *Knowledge-Based Systems*, 196, 105810.

21. Ruane, E., Young, R. and Ventresque, A. (2020). Training a Chatbot with Microsoft LUIS: Effect of Intent Imbalance on Prediction Accuracy. In *Proc. of the Conference on Intelligent User Interfaces Companion (IUI '20)*. Association for Computing Machinery, 63–64.
22. Shneiderman, B. (2020). Human-centered artificial intelligence: Reliable, safe & trustworthy. *International Journal of Human–Computer Interaction*, 36(6), 495-504.
23. Turing, A.M. (2009). Computing machinery and intelligence. *Parsing the turing test*. Springer, 23-65.
24. WeChat. *WeChat*. Retrieved from <https://www.wechat.com/it/> Last Access April 10, 2021.
25. Woods, W., Kaplan, R. and Nash-Webber, B. 1974. The Lunar Sciences Natural Language Information System, Cambridge, MA.
26. Yang, Q., Steinfeld, A., Rosé, C. and Zimmerman, J. (2020). Re-examining Whether, Why, and How Human-AI Interaction Is Uniquely Difficult to Design. In *Proc. of the Conference on Human Factors in Computing Systems (CHI '20)*. Association for Computing Machinery, 1–13.