Collaborative modelling: chatbots or on-line tools? An experimental study

Ranci Ren
Universidad Autónoma de Madrid
Madrid, Spain
ranci.ren@estudiante.uam.es

John W. Castro[†]
Universidad de Atacama
Copiapó, Chile
john.castro@uda.cl

Adrián Santos
University of Oulu
Oulu, Finland
adrian.santos.parrilla@oulu.fi

Sara Pérez-Soler, Silvia T. Acuña, Juan de Lara
Universidad Autónoma de Madrid
Madrid, Spain
{sara.perezs, silvia.acunna, juan.delara}@uam.es

ABSTRACT

Modelling is a fundamental activity in software engineering, which is often performed in collaboration. For this purpose, on-line tools running on the cloud are frequently used. However, recent advances in Natural Language Processing have fostered the emergence of chatbots, which are increasingly used for all sorts of software engineering tasks, including modelling. To evaluate to what extent chatbots are suitable for collaborative modelling, we conducted an experimental study with 54 participants, to evaluate the usability of a modelling chatbot called *SOCIO*, comparing it with the on-line tool *Creately*. We employed a within-subjects cross-over design of 2 sequences and 2 periods. Usability was determined by attributes of efficiency, effectiveness, satisfaction and quality of the results. We found that *SOCIO* saved time and reduced communication effort over *Creately*. *SOCIO* satisfied users to a greater extent than *Creately*, while in effectiveness results were similar. With respect to diagram quality, *SOCIO* outperformed *Creately* in terms of precision, while solutions with *Creately* had better recall and perceived success. However, in terms of accuracy and error scores, both tools were similar.

CCS CONCEPTS

• Human-centered computing • Usability testing • *Software and its engineering* • *Software design engineering* • *Collaboration in software development*

KEYWORDS

Collaborative modelling, Usability, Chatbots, Effectiveness, Efficiency, Satisfaction, Quality.

ACM Reference format:

Ranci Ren, John W. Castro, Adrián Santos, Sara Pérez-Soler, Silvia T. Acuña and Juan de Lara. 2020. Collaborative modelling: chatbots or on-line tools? An experimental study. In *Proceedings of International Conference on Evaluation and Assessment in Software Engineering (EASE'20). ACM, New York, NY, USA, 10 pages.* https://doi.org/10.

1 Introduction

Modelling is an integral part of all engineering disciplines, like mechanical, electrical or software engineering. Often, modelling becomes a collaborative activity, requiring the participation of stakeholders with different backgrounds and technical expertise [7]. In software engineering, both asynchronous (i.e., based on version control) and synchronous (e.g., based on on-line tools) modelling mechanisms are typically used. For the latter purpose, a plethora of cloud-based platforms have recently emerged, supporting real-time collaboration. These include tools like *GenMyModel* (https://www.genmymodel.com/), *LucidChart* (https://www.lucidchart.com/), *Gliffy* (https://www.gliffy.com/), and *Creately* (https://creately.com/) among many others.

The advance in Natural Language Processing (NLP) techniques has favoured the emergence of chatbots [17]. These are software programs whose user interface is NL – either in text or speech forms – which are frequently embedded within social networks. Almost every industry is proposing chatbots to provide a more flexible access to their services – such as booking flights, perform bank operations, or checking traffic conditions – without the need to install dedicated apps [26]. The boost of chatbots is also partially due to the facilities offered by social networks – like Telegram, Twitter, or Slack – for their integration. Hence, any company can ride the wave of, e.g., Facebook Messenger's success and its huge audience to deploy a bot to engage with the customer. Tens of thousands of chatbots have been created for Facebook Messenger alone. According to forecasts and statistics from Gartner, the chatbot market is quickly growing, since 85% of customer relationships will be supported by artificial intelligence by 2020

[†]Corresponding Author.

EASE'20, April, 2020, Trondheim, Norway

https://doi.org/10.1145/1234567890

[9]. Not only for leisure, but chatbots are increasingly being used to automate software engineering tasks as well. For example, developers use bots to automate deployment tasks, assign software bugs and issues, repair build failures, schedule tasks like sending reminders, integrate communication channels, or for customer support [17]. Recently, chatbots have been proposed for collaborative modelling. For example, in [20][21] a chatbot called *SOCIO* (saraperezsoler.github.io/ModellingBot/) was proposed. The chatbot is integrated within social networks, and interprets the NL phrases of groups of users to create a domain model. This approach lowers the entry barrier to modelling of non-technical experts, promoting a more active role of all the involved stakeholders in a project. In addition, the social network provides natural collaboration support via short messages.

Given the prominent position that chatbots are expected to take in software engineering, our objective is to assess to what extent a chatbot-based approach is suitable for collaborative modelling. For this purpose, we evaluate two alternative tools for collaborative modelling: *Creately* and *SOCIO*. The former is taken as representative of on-line tools, providing a baseline for comparison. The evaluation is based on a user study with 54 participants, and the assessment is made in terms of usability, efficiency, effectiveness, satisfaction and quality of the results. Overall, we found that *SOCIO* saved time and reduced communication effort over *Creately*. *SOCIO* satisfied users to a greater extent than *Creately*, while in effectiveness the results were similar. With respect to diagram quality, *SOCIO* outperformed *Creately* in terms of precision, while solutions with *Creately* had better recall and perceived success. However, in terms of accuracy and error scores, both tools performed similarly. On the one hand, the experiment findings validate an approach based on chatbots for collaborative modelling. This fact is relevant for builders of future modelling tools. On the other, the experiment advances our general understanding of usability of chatbots and provides directions for how to evaluate the usability of chatbots. As noted in [27], the construction of chatbots frequently neglects usability concerns. Hence, techniques for measuring their usability need to be investigated, to help improving the user experience. While experimentation is key in software engineering, there are still few experiments specifically targeting chatbot usability [23]. Our experiment can serve as a guide for the evaluation of chatbots in software engineering.

Paper organization. Section 2 analyses related work, while Section 3 introduces *SOCIO* and *Creately*. Section 4 describes the research method of the experiment, and Section 5 presents the results and their analysis. Section 6 discusses the results and the threats to validity. Finally, Section 7 concludes the paper.

2 Related Work

Next, we review collaborative modelling approaches, with emphasis on user studies; and on evaluations of chatbots' usability.

Collaborative modelling. Software engineering is a team activity, involving multiple engineers and stakeholders [30]. In the analysis and design phase of a project, collaboration involves sharing and working on a set of models. Our focus is on synchronous modelling. As mentioned in the introduction, a plethora of cloud-based tools have emerged, and traditional desktop based platforms, like Eclipse are targeting the web as well (see e.g. EMF.Cloud www.eclipse.org/emfcloud/, or the Graphical Language Server Protocol, www.eclipse.org/glsp/). Usability is one of the limiting factors of collaborative tools, as reported in [7]. Usability can be evaluated via experiments, and we now review some representative ones.

Some collaborative modelling tools can be used from within mobile devices. This is the case of NetSketcher, a tool to build process models [2]. The tool was evaluated informally on a task performed by 6 undergraduates. Also in the area of process modelling, the Cheetah Experimental Platform (CEP) is a collaborative, desktop-based tool with support for collaboration [6]. The tool was evaluated informally, with two engineers creating a simple model. The experiment analysed the collaboration process itself, e.g., observing change of roles of the users (active vs passive) during the collaboration. In [8], Eclipse GMF editors were incorporated collaboration capabilities. The tool was evaluated by 14 students, which defined both a modelling tool (using MDE techniques), and then evaluated the generated tool. Evaluation was performed using questionnaires. Hence, these are small-scale experiments, while the field would benefit from larger ones.

Usability of chatbots. In [23] a Systematic Mapping Study (SMS) is presented, analyzing the HCI mechanisms used to evaluate the usability of chatbots in different fields. In the health care domain, chatbots helped to self-control diseases such as diabetes [4][25], or offered therapy for patients suffering from post-traumatic stress disorders [28]. Other bots were designed to facilitate travel planning [19], help in e-commerce like buying shoes [13], search for information [22] or being the personal assistants, such as Apple Siri and Amazon Alexa [5][18]. The SMS selected 15 papers as primary studies, where 10 described experimental studies of chatbot usability. In most cases, the studies compared the chatbot with another application or system with same functionality or similar key characteristics [4][13][19][22][25]. For example, in [19], a website application and a chatbot are compared to investigate the differences in the levels of satisfaction. In most experiments, simple tasks are proposed, like using Siri to find an inexpensive hotel in Osaka [5] or search a flight ticket and hotel room via the chatbot [19]. A within-subject design was used in three experiments [13][19][22], in which subjects must apply all the treatments to be evaluated. However, each treatment was only used in a particular order. All experiments used questionnaires to collect data about user experience and satisfaction. These were typically provided at the end of the experiment, although in some cases, they were also filled after each task and/or at the beginning of the experiment to better understand basic information about the users [13][22].

With respect to chatbots for collaborative modelling, in [20][21] two small-scale evaluation experiments for *SOCIO* (with 19 and 8 participants) were presented. In [21] the suitability of this chatbot was assessed, while in [20], they evaluated a consensus mechanism for choosing different modelling alternatives. The tasks in both experiments were carried out in groups. All 10 participants in [21] performed the proposed task via Telegram, and were divided in 4 groups (of 2 and 3 people). In [20], all participants formed a single Telegram group. The research method used in both experiments was based on survey questionnaires, filled after finishing the tasks. In [21], the questionnaire was based on the System Usability Scale (SUS) [3], with a part to evaluate user satisfaction, the use of NL, the integration in social networks, and open questions. Questions in [20] focused on evaluating the consensus mechanism. The tasks proposed were relatively simple. In [21], the task was creating a class diagram for an electronic commerce system in 15 minutes. In [20], to select among modelling alternatives to measure the degree of agreement based on the group preferences. Participants chose the best of three options for two projects, the first without consensus mechanism and the second with it. The results in [21] were positive in terms of satisfaction, the suitability of using NL and the idea of collaborating in social networks. Even though the accuracy of interpreting NL was relatively good, results suggested the need to improve in this line. The consensus mechanism was considered useful for large groups and with an outcome that reflects the opinion of the majority [20].

However, those experiments focused on evaluating SOCIO in isolation (i.e., no comparison to a baseline), while the number of participants was small. Therefore, here we report on an experiment with larger number of users and compare with an alternative collaborative modelling approach, based on a traditional GUI.

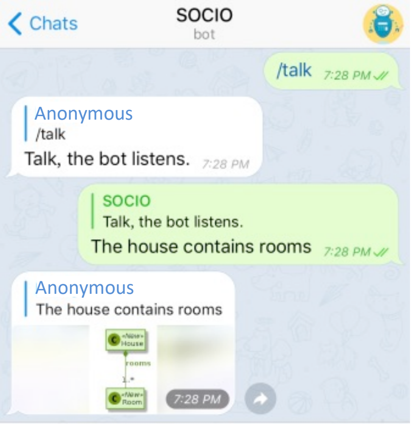
3 A brief overview of *SOCIO* and *Creately*

SOCIO is a chatbot that interprets NL (in English) to create class diagrams [21]. The chatbot is accessible from Twitter or Telegram (with nick *@ModellingBot*). Upon interpreting a NL phrase uttered by the user, it sends back an image with the current model state, with colors highlighting the changed parts. *SOCIO* supports commands to create new models, see user contributions, the percentage of authorship on the created models, among others.

SOCIO offers two types of interaction. The first one is similar to a casual task, based on descriptive phrases like "*the house contains rooms"* (cf. Figure 1). *SOCIO* identifies the relevant parts of a phrase (nouns, verbs, adjectives), to decide which actions to perform (creating or updating a class, an attribute, a relation). In the example of the Figure, it identifies two nouns (house, rooms), for which two classes are created. The "contains" verb is mapped to a containment reference, while the plural form of "rooms" suggests a *many* cardinality.

It can be noted that, in Telegram, the bot cannot directly listen to messages of the users in a group, which need to address the bot using the "*/talk*" command.

The second way to address the bot is more similar to using commands, like "*add class X*'', or "*set attribute size to int*''. Still, these commands have a flexible syntax, as illustrated in Figure 2 (where again the class names are added in singular).



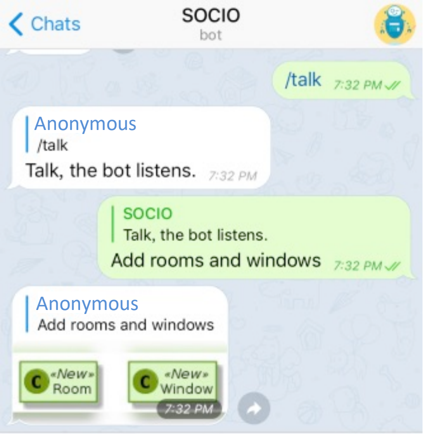
/talk

Anonymous
/talk
Talk, the bot listens.

SOCIO
Talk, the bot listens.
The house contains rooms

Anonymous
The house contains rooms

Figure 1: Processing descriptive NL messages



/talk

Anonymous
/talk
Talk, the bot listens.

SOCIO
Talk, the bot listens.
Add rooms and windows

Anonymous
Add rooms and windows

Figure 2: Processing command-like imperative messages

In contrast to *SOCIO, Creately* uses a traditional GUI, accessible through a web browser. The tool supports over 50 types of diagrams – including class diagrams – and real-time collaboration. In addition, the tool supports working offline, and re-synchronization when connectivity is available.

Figure 3 shows a screenshot of the tool. *Creately* is built on Adobe's Flex/Flash technologies and provides a visual communication platform for virtual teams. While *SOCIO* embeds modelling within a social network, *Creately* lacks an embedded chat. Hence, external ones, like Telegram should be used instead.

Since *Creately* is one of the most used online collaborative modelling tools[1] with friendly interface and learnability, we chose it as the control tool for comparing with *SOCIO*.

[1] according to modeling-languages.com/web-based-modeling-tools-uml-er-bpmn/

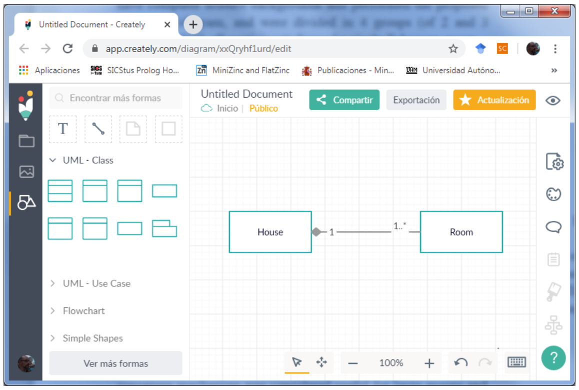


Figure 3: *Creately* being used for class diagram modelling

4 Research Method

The objective of the research is to evaluate the usability of the chatbot *SOCIO* by comparing it to the web tool *Creately* with respect to effectiveness, efficiency and satisfaction, from the point of view of users, and the quality of the class diagrams obtained. In particular, we make the following research question:

RQ: Compared to *Creately,* does the use of *SOCIO* positively affect the efficiency, effectiveness and satisfaction of the users when making class diagrams, and the quality of class diagrams?

The research hypotheses are:

H.1.0 There is no difference in efficiency between *SOCIO* and *Creately* when making a class diagram.

H.2.0 There is no difference in effectiveness between *SOCIO* and *Creately* when making a class diagram.

H.3.0 There is no difference in satisfaction between *SOCIO* and *Creately* when making a class diagram.

H.4.0 There is no difference in the quality of the class diagram made with *SOCIO* or *Creately*.

4.1 Experimental setting

The experiment was structured as a 2 **sequences** and 2 **periods within-subjects cross-over** design (see Table 1). Cross-over designs have the advantage of reducing variability – as subjects act as their own baseline – and require a smaller number of subjects than between-subjects designs – as subjects have as many measurements as periods [29].

The participants were grouped in *teams* of 3 members. The teams were randomly assigned to one out of two groups (Group 1 or Group 2, onwards), so each group applies the *treatments* in a different order (AB/BA). The *treatments* are two tools for creating class diagrams: the chatbot *SOCIO* and the web application *Creately*. Group 1 first applies *SOCIO* and then *Creately* (i.e., *SOCIO-Creately* sequence, SC-CR). Conversely, group 2 first applies *Creately* and then *SOCIO* (i.e., *Creately-SOCIO* sequence, CR-SC). Both groups implement the tasks in the same order (task 1 and task 2). Each task consists of a class diagram that needs implementing.

Table 1: Experimental Design

Tool	Task Period Sequence	Task 1 Period 1		Task 2 Period 2	
		SC	CR	SC	CR
Group 1:	*SC-CR*	X	__	__	X
Group 2:	*CR-SC*	__	X	X	__

Finally, participants in the same team are only allowed to communicate with each other in Telegram groups – so as to ensure that we record all the experimental data.

4.2 Participants

A total of 54 participants took part in the experiment. They all had a degree in Computer Science or a related degree from the *Universidad de las Fuerzas Armadas ESPE Extensión Latacunga* in Ecuador. All participants had studied or were studying a course on Software Analysis and Design. Thus, they had the necessary knowledge to make a class diagram.

4.3 Procedure

The 54 participants were split into two groups of 27 participants each. The participants in each group were further divided into 9 teams. The teams were randomly created. A total of 18 teams participated in the experiment (9 per group). To fit within the participants' timetable, the experiment was run in four sessions over two days, with each participant attending one session.

The subjects did not undergo any preparatory or practice session before the experiment took place. All the subjects signed an informed consent form indicating that they granted us permission to record their data via Telegram. Then, subjects completed a familiarity questionnaire designed to help us collect their basic information (i.e., age, gender, level of English, preconceived ideas regarding their use of social media, and level of knowledge on class diagrams).

All the participants first received a brief tutorial about the tool they had to use. Then, they were required to perform the first task with the tool in a maximum of 30 minutes. We found such length appropriate so as not to fatigue the participants. We adapted the complexity of the class diagram to the experimental session length. In particular, it was a class diagram representing a store, including management of products and customers. At the end of the experimental session the subjects filled in a modified and validated satisfaction questionnaire System Usability Scale (SUS) associated with the tool [3].

Once the questionnaire was completed, participants received a tutorial of the second tool. Then, they performed the second task with the tool in a maximum of 30 minutes. The task consisted in designing the class diagram of a school supporting courses and students. At the end of the allowed time the participants filled in another modified SUS satisfaction questionnaire, with questions about the tool. In this last questionnaire, the participants were asked if they preferred *SOCIO* or *Creately*. Figure 4 shows the detail of each session. The experimental data and materials can be downloaded from: https://bit.ly/2vfYZNB.

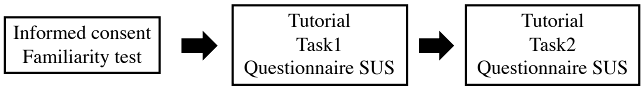


Figure 4: Experimental Procedure

4.4 Measure

The ISO/IEC 25010 [12] defines efficiency, effectiveness, satisfaction, and quality in use as common attributes for evaluating product usability. The response variables that we used and their respective metrics are outlined below.

We used the following metrics to measure efficiency:

- *Speed*: Time, measured in minutes, taken by a team to complete the task (with a maximum of 30 minutes).
- *Fluency*: Number of discussion messages generated by a team during the completion of the task via a Telegram group.

The metric we used to measure effectiveness was *completeness*, based on the *perceived success* in carrying out the task. Satisfaction was measured by the modified SUS questionnaire, including SUS questions, and three or four open-ended questions. The SUS questions are ordinal questions on a 5-point Likert scale – with a rating of 1 to 5, 1 representing "*strongly disagree*", and 5 representing "*strongly agree*". We select the median of the scores given by the three members of each team – to each question – as the score of the team. Finally, we calculate the average of the SUS scores of each team as their satisfaction score. We adopted Brook's equations [15][24] to derive the numerical value of each user's individual tool session score. The corresponding equations are shown below:

For questions 1, 3, 5, 7, 9:

$$\text{Sum1} = \text{score value} - 1 \qquad (1)$$

For questions 2, 4, 6, 8, 10:

$$\text{Sum2} = 5 - \text{score value} \qquad (2)$$

$$\text{SUS score} = 2.5 * (\text{sum1} + \text{sum2}) \qquad (3)$$

Based on the values derived from this equation, we compared these two tools in matters of *satisfaction*. This calculation provided us with a way of quantifying satisfaction. We took an ideal class diagram as a reference to measure the **quality** of the teams' class diagrams. Such class diagram was designed by Software Engineering experts before the experiment took place. We used the following metrics to measure quality [10]:

$$\textbf{Precision} = TP / (TP + FP) \qquad (4)$$

$$\textbf{Recall} = TP / (TP + FN) \qquad (5)$$

$$\textbf{Accuracy} = (TN + TP) / (TP + FP + FN + TN) \qquad (6)$$

$$\textbf{Error} = (FP + FN) / (TP + FP + FN + TN) \qquad (7)$$

$$\textbf{Success} = TP / (\#\text{predicted diagram elements}) \qquad (8)$$

The previous formulas can be computed by comparing the ideal class diagram with the class diagrams' true positives (TP), false positives (FP), false negatives (FN) and true negatives (TN):

- TP (true positive): Number of elements that are found in both the ideal class diagram and the team' class diagram.
- FN (false negative): Number of elements that are found in the ideal class diagram, but not in the team' class diagram.
- FP (false positive): Number of elements that are found in the team class diagram, but not in the ideal class diagram.
- TN (true negative): In the comparison of models there are no true negatives, and hence the value is always 0.

This way, *precision* gives the percentage of correct classes in the solution of each team, based on the elements of the ideal diagram. *Recall* is a completeness metric, giving the percentage of classes of the ideal diagram present in the solution. *Accuracy* combines both metrics, and *error* reflects how many elements are redundant or missing in each solution. The perceived *success* refers to success rate of each team, compared with the ideal class diagram directly.

5 Data Analysis and Results

We analysed each of the four response variables (i.e., efficiency, effectiveness, satisfaction and quality) with a Linear Mixed Model following the advice of Vegas et al. [29]. In particular, we fitted a linear mixed model with the following factors: **(1) sequence** (either *Creately-SOCIO* or *SOCIO-Creately*), accounting for the assignment of teams to a combination of task and treatment; **(2) period** (either Session 1 or Session 2), confounded with task, accounting for the task that the teams had to implement; and **(3) treatment** (either *Creately* or *SOCIO*), accounting for the tool applied by the teams to implement the tasks.

We complement the results of the statistical analysis with Cohen's *d* for the treatments (*d*, hereinafter) and their standard errors (SEs). For this, we follow the formulae provided in the Cochrane Handbook for cross-over designs [11]. In the next subsections, we go over the data analysis.

5.1 Descriptive Data

According to the data gathered in the familiarity questionnaire, the participants have the following characteristics:

- From a total of 54 subjects, 44 are men and 10 are women.
- Subjects have a mean age of 22 and a standard deviation of 1.74. The highest concentration of participants is in the range 21-23 years.
- 66.7% of subjects use social media frequently. WhatsApp, Facebook, Instagram and Telegram are the most used social media applications.

- All the participants believe they are knowledgeable about class diagrams, and 90% of them relatively familiar with class diagrams.
- 87.1% of the participants have used or use Telegram frequently. 12.9% have no experience using Telegram.
- In relation to chatbots, all participants consider they understand them – at least at the conceptual level. Regarding their usage habits, 29.6 % have never used a chatbot, while 70.4% have some experience (55.6% have used chatbots at times and 14.8% are regular users). The fact of having subjects lacking previous experience with chatbots contributes to the greater sensitivity to the usability of the tool and the validity of the results.
- Although no subject is a native English speaker, all of them considered having a fluent level of English.

5.2 Efficiency

We measured efficiency in terms of *speed* and *fluency*. Speed corresponds to the time taken to complete the tasks. Fluency corresponds to the number of discussion messages exchanged between the team members during the tasks' implementation. Figures 5 and 6 show the box-plots corresponding to speed and fluency, respectively. Table 2 and 3 show the results of the linear mixed model fitted to analyse the data.

5.2.1 Speed. As we can see in Figure 5, time spent seems to be less on *SOCIO* than in *Creately*.

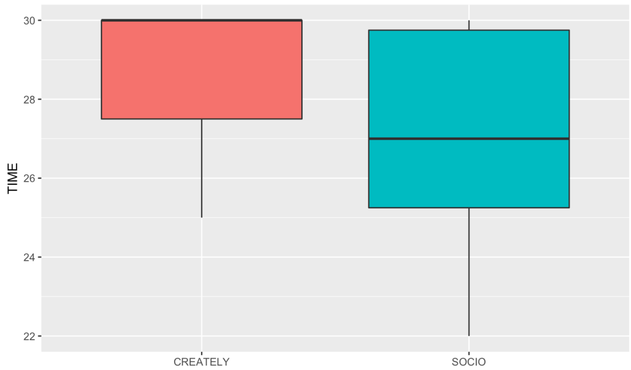


Figure 5: Time spent on completing the task

As we can see in Table 2, only the treatment has a statistically significant impact on time. In particular, the time spent with *Creately* is on average 1.78 minutes longer than that in *SOCIO*. Finally, d=0.80, SE(d)=0.41, suggesting that a large effect size – according to rules of thumb [1] – materialized for the treatment. This large effect size could be because: (i) *Creately* is built on Adobe Flash which caused errors, and at times users needed to re-enter the application during the experiment, and (ii) the delay in the collaborative process with *Creately* was noticeable, and sometimes users could not perform any operations while teammates were operating or (iii) Creately requires users to take care of layouting the diagram, which is not necessary in *SOCIO*.

Table 2: Linear Mixed Model for Time

	Estimate	Std. Error	*p*-value
(Intercept)	27.89	0.73	0.00
Seq	1.11	0.73	0.15
Treatment	-1.78	0.73	**0.03**
Period	0.78	0.73	0.30

5.2.2. Fluency. As we can see in Figure 6, the number of discussion messages seem smaller for *SOCIO* than for *Creately*.

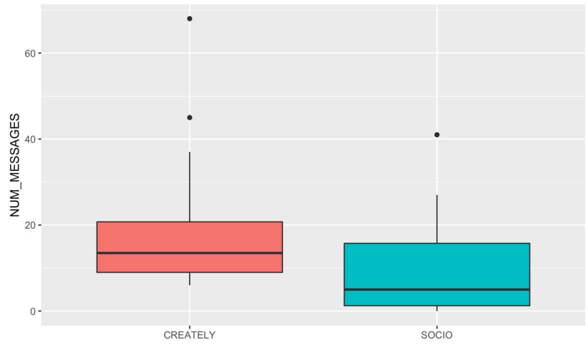


Figure 6: Number of Discussion Messages

As we can see in Table 3, only the treatment has a statistically significant impact on the number of discussion messages. Compared with *SOCIO*, the users sent 10 more messages with *Creately*. With d=0.70, SE(d)=0.22, a relatively large effect size materialized [1].

Table 3: Linear Mixed Model for Number of Discussion Messages

	Estimate	Std. Error	*p*-value
(Intercept)	22.72	4.78	0.0002
Seq	-3.17	6.10	0.61
Treatment	-9.94	2.92	**0.0036**
Period	-3.17	2.92	0.29

In sum, *SOCIO* saved more time in terms of communication effort than *Creately*. **In both aspects, *SOCIO* seems more efficient**.

5.3 Effectiveness

We used the degree of completeness of the tasks to measure effectiveness.

5.3.1 Completeness. Figure 7 shows a box-plot corresponding to the completeness scores of the teams per treatment. As we can see in Figure 7, the results for *completeness* seem similar – albeit more spread for *Creately*.

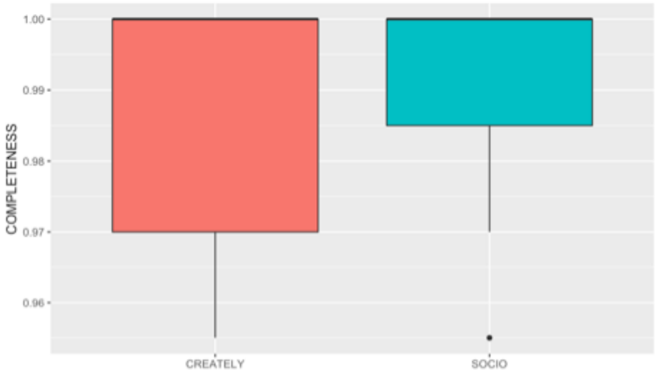


Figure 7: Completeness Scores

Table 4 shows the results of the linear mixed model fitted. As we can see in Table 4, none of the factors has a statistically significant impact on completeness. Finally, d=-0.21, SE(d)=0.34 suggesting a small effect size [1].

Table 4: Linear Mixed Model for Completeness

	Estimate	Std. Error	*p*-value
(Intercept)	0.985	0.00515	0.00
Seq	0.005	0.00518	0.34
Treatment	0.003	0.00512	0.52
Period	0.0083	0.00512	0.12

Thus, ***SOCIO* and *Creately* performed similar in terms of effectiveness**.

5.4 Satisfaction

We used a questionnaire to evaluate users' satisfaction towards *SOCIO* and *Creately*. Each questionnaire included the ten questions of the SUS and four open questions at the end.

5.4.1 Open-ended Questions. Both tools were reliable according to the participants. However, compared to *Creately*, *SOCIO* received more positive responses. Next, we analyse each question:

Q1: Please indicate three positive aspects that you want to highlight about the tool.

Both tools satisfied the participants because of their responsiveness, ease of use and collaboration capabilities. Besides, *Creately* was praised for its friendly interface. Some participants claimed that the chatbot was user-friendly and that it allowed them to have a more entertaining interaction. In other words, *SOCIO* was better suited to entertain the users.

Q2: Please indicate three negative aspects of the tool

Some participants complained that *SOCIO*'s documentation on its website were not sufficient. Additional answers mentioned that commands for *SOCIO* were not easy to learn, and some commands were lacking. Some of the participants expressed that *SOCIO* requires prior knowledge.

The biggest problems with *Creately* were related to real time collaboration, which produced some errors when loading on some of the user's computers. Some participants were not satisfied with the interface as it was too simple. Besides, some users claim that *Creately*'s functions were not comprehensive enough.

Q3: Do you have any suggestions for improvement?

For SOCIO, a number of participants suggested an improvement in its support for NLP. For *Creately,* participants suggested to improve its real time collaboration, and improve its user interface, which some participants considered too simple.

Q4: Which tool do you prefer?

Participants showed relatively positive emotions towards both tools, especially in the aspect of anticipation. Besides, they expressed more trust and joy for *SOCIO* than for *Creately*. Overall, 30 of the participants preferred *SOCIO*, while 24 expressed their preference towards *Creately*.

5.4.2 Questions of the SUS. We used the SUS score given by the participants to both tools and compared them side-by-side. Figure 8 shows the box-plot for the SUS scores.

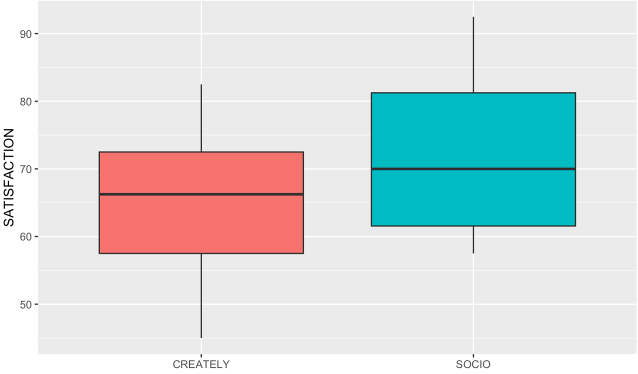


Figure 8: Satisfaction Scores

As Figure 8 shows, the satisfaction scores are typically higher for *SOCIO* than for *Creately*. As we can see in Table 5, the treatment has an almost statistically significant effect on satisfaction (p=0.1). In particular, d=0.58, SE(d)=0.35, thus, suggesting a medium effect size [1]. This indicates that ***SOCIO* satisfies users to a greater extent than *Creately***.

Table 5: Linear Mixed Model for Satisfaction

	Estimate	Std. Error	*p*-value
(Intercept)	64.51	3.88	0
Seq	1.69	3.97	0.69
Treatment	6.60	3.79	0.10
Period	-1.18	3.79	0.75

5.5 Quality

We analysed the quality of the class diagrams in various aspects: *precision*, *recall*, *accuracy*, *error* and *perceived success* (cf. equations 4-8). The box-plots for such metrics are shown in Figure 9-13, while the linear mixed models fitted are shown in Tables 6-10.

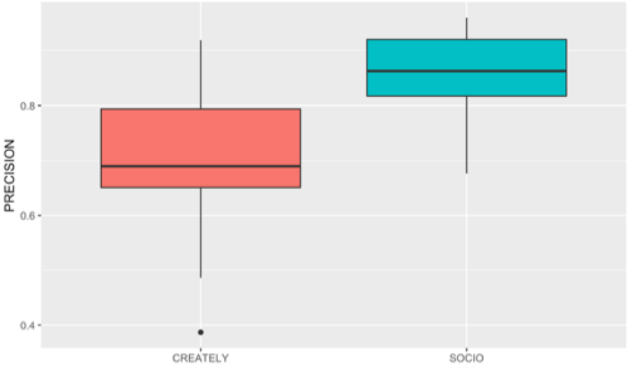


Figure 9: Precision Scores

Table 6: Linear Mixed Model for Precision

	Estimate	Std. Error	*p*-value
(Intercept)	0.731	0.055	0
Seq	0.008	0.066	0.911
Treatment	0.108	0.041	**0.018**
Period	-0.141	0.041	**0.003**

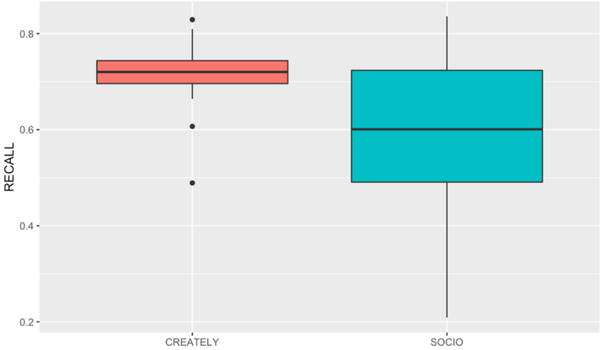


Figure 10: Recall Scores

Table 7: Linear Mixed Model for Recall

	Estimate	Std. Error	*p*-value
(Intercept)	0.729	0.051	0
Seq	-0.026	0.061	0.677
Treatment	-0.145	0.038	**0.001**
Period	-0.006	0.038	0.885

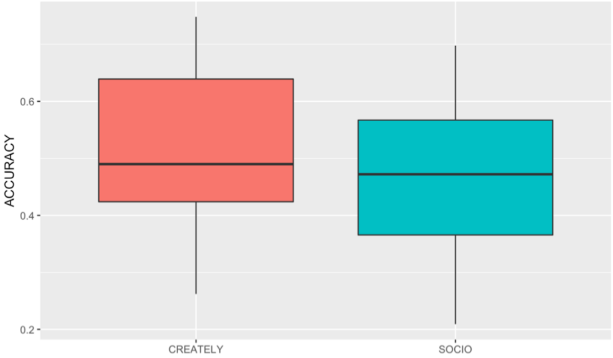


Figure 11: Accuracy Scores

Table 8: Linear Mixed Model for Accuracy

	Estimate	Std. Error	*p*-value
(Intercept)	0.546	0.048	0
Seq	0.015	0.067	0.81
Treatment	-0.048	0.031	0.13
Period	-0.069	0.031	**0.04**

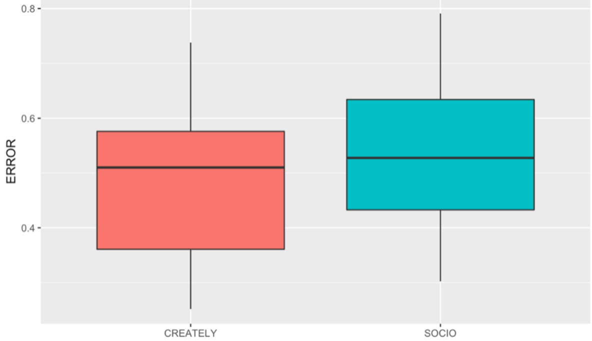


Figure 12: Error Scores

Table 9: Linear Mixed Model for Error

	Estimate	Std. Error	*p*-value
(Intercept)	0.453	0.048	0
Seq	-0.015	0.061	0.812
Treatment	0.048	0.031	0.135
Period	0.069	0.031	**0.039**

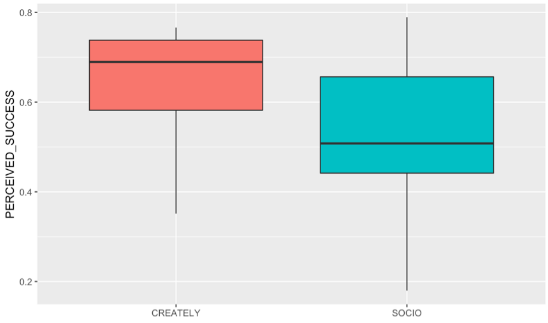


Figure 13: Perceived Success

Table 10: Linear Mixed Model for Perceived Success

	Estimate	Std. Error	*p*-value
(Intercept)	0.642	0.049	0
Seq	0.066	0.058	0.270
Treatment	-0.147	0.038	**0.001**
Period	-0.049	0.038	0.218

As shown above, the treatment has a significant impact on *precision* (where d=0.619, SE(d)=0.324, *SOCIO* outperforming *Creately*); *recall* (d=0.976, SE(d)=0.232, *Creately* outperforming *SOCIO*); and *perceived success* (d=0.996, SE(d)= 0.307, *Creately* outperforming *SOCIO*). However, in terms of accuracy and error scores both tools seem to perform similarly as indicated by the non-significance of the treatment factor, and the smallest effect sizes in such cases (d=0.334, SE(d)=0.240 for accuracy; d=-0.334, SE(d)=0.240 for error scores).

Summarizing, ***SOCIO* outperforms *Creately* in terms of precision, while *Creately* outperforms *SOCIO* in terms of recall and perceived success.**

6 Discussion and Threats to Validity

Overall, *SOCIO* seems superior to *Creately* in terms of efficiency and satisfaction, while in effectiveness they are similar. This suggests that *SOCIO* saved time and communication effort to the users. Also, that *SOCIO's* look and feel met the users' expectations to a greater extent than *Creately*. In addition, users created more precise class diagrams with *SOCIO* than with *Creately*. This means that a larger percentage of the classes created with *SOCIO* were also included in the ideal solution. This, and the observation that *Creately* was superior to *SOCIO* in terms of recall and perceived success, suggests that users made fewer classes with *SOCIO* than with *Creately* – albeit the diagrams were more complete with *Creately*. In plain words, users seemed to create more classes from the ideal solution with *Creately* than with *SOCIO* – despite it took longer to the users creating such classes with *Creately*, and *Creately's* interface seems not as appealing as *SOCIO's*.

Our take away from these results is that despite its greater precision, *SOCIO's* class diagrams may be lacking completeness due to the low training of the participants with the tool, and its English interface (as none of the participants was a native English speaker). In fact, participants highlighted the need of *SOCIO* to support more languages (namely, Spanish), social media platforms (e.g., Facebook Messenger rather than Telegram), and the need of more detailed examples in the manual. Also, they wished *SOCIO* helped auto-correcting spelling mistakes. Despite this, the satisfaction of the participants with *SOCIO* is relatively high. Notice that *SOCIO* scores better than *Creately* in some respects, but since it is not known or validated how good *Creately* is, this cannot be used as a basis for a comprehensive evaluation for *SOCIO*. However, the fact that *Creately* is one of the most used tools suggests that it is at least one of the best ones, and supports the conclusion that *SOCIO* is a good modelling tool.

Next we analyse threats to validity. **Internal validity** pertains to confounding factors that could influence our results. In the experiment, participants had to create two class diagrams. Although they already had the necessary knowledge for this task, the first task may have refreshed this knowledge. Therefore, the second treatment applied may provide better results. This can be mitigated by comparing the results in the two periods (two tasks), and studying any improvement observed. Although the sessions did not have an excessively long duration (an hour and a half) there could be a threat of tiredness or boredom. The subjects participated voluntarily, and their collaboration did not imply any impact on the grades of the course, so they might have suffered from a lack of motivation. An additional threat to the internal validity is related to the fact that participants were not English native speakers. Hence, the user experience and time spent may be affected by their English fluency.

Regarding **external validity** (generalizability of the results), our participants are university students with knowledge in computer science and class diagram design. Hence, the results are not generalizable to the industrial field, but can only remain in the academic realm. In addition, the evaluation has used *SOCIO* and *Creately*, therefore the results cannot be directly generalized to other modelling chatbots, or on-line modelling tools.

Besides, there is a threat to **conclusion validity** because we have performed many statistical tests, and hence, this has increased the risk of a statistical error for type I (saying there is an effect, when there is not). We decided not to apply any correction for multiple tests, like Bonferroni, due to the relatively small sample size of the experiment. However, we have complemented the statistical results with the effect sizes, to facilitate the interpretation of the practical relevance of the findings. All in all, we consider these results preliminary and proper sized experiments are still required to draw definite conclusions on the performance of SOCIO and *Creately*.

Finally, there is another threat to conclusion validity regarding the experimental tasks-because they may have impacted the experiment's results. To tackle this shortcoming, we plan to run more experiments assessing the performance of SOCIO and Creately with a different set of tasks.

7 Conclusion

Modelling is a team activity that is often performed in collaboration. Traditionally, collaborative modelling has been performed asynchronously in offline environments, or using online collaboration, sometimes in cloud-based tools. However, we have recently witnessed the emergence of chatbots, which are being used for all types of activities, including software engineering tasks like modelling. As usability of chatbots – in particular for modelling – is largely unexplored, in this paper we have reported on an evaluation comparing the *SOCIO* chatbot with the *Creately* on-line tool. Our aim was to answer the following research question:

> **RQ:** Compared to *Creately,* does the use of *SOCIO* positively affect the efficiency, effectiveness and satisfaction of the users when making class diagrams, and the quality of class diagrams?

We evaluated the usability of *SOCIO* from four aspects: efficiency, effectiveness, satisfaction and quality. Regarding efficiency, teams using *SOCIO* finished earlier than those using *Creately*. For collaboration, those using *SOCIO* showed high fluency, with an interaction-cost advantage over those using *Creately*. For effectiveness, *SOCIO* and *Creately* performed similarly in terms of completeness. For satisfaction, *SOCIO* satisfies users to a greater extent than *Creately* with respect to the results of the SUS score. More users expressed they preferred *SOCIO* rather than *Creately*. For quality *SOCIO* outperformed Creately in terms of precision, while solutions with *Creately* had better recall and perceived success. In sum, usability of *SOCIO* has a positive effect on most aspects, when taking *Creately* as a baseline.

In the future, we plan to conduct a second round of evaluations engaging more users to interact with the chatbot *SOCIO*, especially we will aim at English native speakers. Finally, we would like to enhance *SOCIO* with speech recognition, to enable design workshops using conversation, in the style of [14][16].

Acknowledgements

Work funded by the Spanish Ministry of Science (project MASSIVE, RTI2018-095255-B-I00) and the R&D programme of Madrid (project FORTE, P2018/TCS-4314).

REFERENCES

[1] Michael Borenstein, Larry V. Hedges, Julian P. T. Higgins, and Hannah R. Rothstein. 2009. *Introduction to meta-analysis*. West Sussex, Wiley, UK.

[2] Nelson Baloian, Gustavo Zurita, Flávia Maria Santoro, Renata Mendes de Araujo, S. Wolfgan, D. Machado, and José A. Pino. 2011. A collaborative mobile approach for business process elicitation. In *Proc. 2011 15th Int. Conf. Comp. Supp. Coop. W. Design (CSCWD0'11)*. Lausanne, Switzerland, 473-480.

[3] John Brooke. 1996. *SUS-a quick and dirty usability scale*. In P. W. Jordan, B. Thomas, B. A. Weerdmeester, & A. L. McClelland (Eds.). Usability Evaluation in Industry, Chapter 21, 189-194.

[4] Amy Cheng, Vaishnavi Raghavaraju, Jayanth Kanugo, Yohanes P. Handrianto, and Yi Shang. 2018. Development and evaluation of a healthy coping voice interface application using the google home for elderly patients with type 2 diabetes. In *Proc. 15th IEEE Ann. Consumer Communic. & Netw. Conf. (CCNC'18)*. Las Vegas, NV, USA, 1-5.

[5] Mei-Ling Chen and Hao-Chuan Wang. 2018. How Personal Experience and Technical Knowledge Affect Using Conversational Agents. In *Proc. 23rd Int. Conf. Intell. U. Interf. Comp. (IUI'18 - Comp)*. ACM. Tokyo, Japan. Article 53.

[6] Simon Forster, Jakob Pinggera, and Barbara Weber. 2013. Toward an understanding of the collaborative process of process modeling. *CAiSE Forum* 2013, 98-105.

[7] Mirco Franzago, Davide Di Ruscio, Ivano Malavolta, and Henry Muccini. 2018. Collaborative Model-Driven Software Engineering: A Classification Framework and a Research Map. *IEEE Transact. Softw. Engin.* 44, 1146-1175.

[8] Jesús Gallardo, Crescencio Bravo, and Miguel A. Redondo. 2012. A model-driven development method for collaborative modeling tools. *J. Network and Comp. Applic.* 35(3), 1086-1105.

[9] Gartner. 2011. CRM Strategies and Technologies to Understand, Grow and Manage Customer Experiences. Gartner Customer 360 Summit 2011. Gartner, Los Angeles, CA.

[10] Fáber D. Giraldo, Sergio España, William Giraldo Orozco, and Oscar Pastor. 2018. Evaluating the quality of a set of modelling languages used in combination: A method and a tool. *Information Systems* 77, 48-70.

[11] Julian P. T. Higgins and Sally Green. 2011. *Cochrane handbook for systematic reviews of interventions*, (vol. 4). John Wiley & Sons.

[12] ISO/IEC 25010. 2011. Systems and Software Engineering - Systems and Software Quality Requirements and Evaluation (SQuaRE) - System and Software Quality Models. ISO. Geneva, Italy.

[13] Mohit Jain, Ramachandra Kota, Pratyush Kumar, and Shwetak N. Patel. 2018. Convey: Exploring the Use of a Context View for Chatbots. In *Proc. 2018 CHI Conf. Hum. Fact. Comp. Syst. (CHI'18)*. ACM, New York, USA, Paper 468.

[14] Rodi Jolak, Boban Vesin, Michel R. V. Chaudron. 2017. Using Voice Commands for UML Modelling Support on Interactive Whiteboards: Insights and Experiences. In *Proc. Iber. Conf. Soft. Eng* (*CIbSE'17*). Bs. As., Argentina, 85-98.

[15] Patrick W. Jordan, Bruce Thomas, Ian Lyall McClelland, and Bernard Weerdmeester. 1996. *Usability Evaluation in Industry* (1st. ed.). CRC Press Taylor & Francis Group.

[16] Samuel Lahtinen, Jari Peltonen. 2005. Adding speech recognition support to UML tools. *Journal of Visual Lang. Comput* 16(1-2), 85-118.

[17] Carlene Lebeuf, Margaret-Anne D. Storey, and Alexey Zagalsky. 2018. Software bots. *IEEE Software* 35(1), 18-23.

[18] Irene Lopatovska, Katrina Rink, Ian Knight, Kieran Raines, Kevin Cosenza, Harriet Williams, Perachya Sorsche, David Hirsch, Qi Li, and Adrianna Martinez. 2019. Talk to me: Exploring user interactions with the amazon alexa. *J. Librarianship and Information Science* 51(4), 984-997.

[19] Quynh N. Nguyen and Anna Sidorova. 2018. Understanding user interactions with a chatbot: A self-determination theory approach. In *Proc. Americas Conf. Inform. Syst. 2018: Digital Disrupt. (AMCIS'18).* New Orleans, LA, USA. 1-5.

[20] Sara Pérez-Soler, Esther Guerra, and Juan de Lara. 2018. Collaborative modeling and group decision making using chatbots in social networks. *IEEE Software* 35(6), 48-54.

[21] Sara Pérez-Soler, Esther Guerra, Juan de Lara, and Francisco Jurado. 2017. The rise of the (modelling) bots: towards assisted modelling via social networks. In *Proc. 32nd IEEE/ACM Int. Conf. Autom. Softw. Eng. (ASE'17)*. IEEE Press, Piscataway, NJ, USA, 723-728.

[22] Joaquín Pérez, Yanet Sánchez, Francisco J. Serón, and Eva Cerezo. 2017. Interacting with a Semantic Affective ECA. In *Proc. Int. Conf. Int. Virt. Agents (IVA'17).* Lecture Notes in Computer Science, vol 10498. Springer. 374-384.

[23] Ranci Ren, John W. Castro, Silvia T. Acuña, and Juan de Lara. 2019. Usability of chatbots: A systematic mapping study. In *Proc. 31st Int. Conf. Soft. Eng. and Knowledge Eng. (SEKE'19)*. Lisbon, Portugal, 479-484.

[24] Julia Saenz, Walker Burgess, Elizabeth Gustitis, Andres Mena, and Farzan Sasangohar. 2017. The usability analysis of chatbot technologies for internal personnel communications. In *Proc. 67th Ann. Conf. and Expo of the Instit. of Industrial Engineers*. Pittsburgh, United States, 1357-1362.

[25] Claudia Sinoo, Sylvia van der Pal, Olivier A. Blanson Henkemans, Anouk Keizer, Bert P. B. Bierman, Rosemarijn Looije, and Mark A.Neerincx. 2018. Friendship with a robot: Children's perception of similarity between a robot's physical and virtual embodiment that supports diabetes self-management. *Patient Education and Counseling* 101(7), 1248-1255.

[26] Sam Suthar. 2019. 11 Chatbot Trends to Help Grow your Business in 2019. Retrieved November 11, 2019, from https://acquire.io/blog/chatbots-trends.

[27] The Interaction Design Foundation. 2019. What is usability? Retrieved November 11, 2019, from https://www.interaction-design.org/literature/topics/usability.

[28] Myrthe L. Tielman, Mark A. Neerincx, Rafael Bidarra, Ben Kybartas, and Willem-Paul Brinkman. 2017. A Therapy System for Post-Traumatic Stress Disorder Using a Virtual Agent and Virtual Storytelling to Reconstruct Traumatic Memories. *J. Medical Systems* 41, 125.

[29] Sira Vegas, Cecilia Apa, and Natalia Juristo. 2016. Crossover designs in softw. eng. experiments: Benefits and perils. *IEEE Trans. Soft. Eng.* 42(2), 120-135.

[30] Jim Whitehead, Ivan Mistrík, John Grundy, and André van der Hoek. 2010. Collaborative software engineering: Concepts and techniques. Collaborative Software Engineering. In: Mistrík I., Grundy J., Hoek A., Whitehead J. (eds). *Collaborative Software Engineering* (pp. 1-30). Springer, Berlin, Heidelberg.