





Using a Data-Driven Context Model to Support the Elicitation of Context-Aware Functionalities – A Controlled Experiment

Rodrigo Falcão¹ , Marcus Trapp¹, Vaninha Vieira²,
and Alberto Vianna Dias da Silva^{3,4} 

¹ Fraunhofer Institute for Experimental Software Engineering IESE,
Kaiserslautern, Germany

`rodrigo.falcao@iese.fraunhofer.de`

² Institute of Computing, Federal University of Bahia, Salvador, Brazil

³ Computer Science Graduate Program, Federal University of Bahia, Salvador, Brazil

⁴ Federal Institute of Bahia, Salvador, Brazil

Abstract. Background: Context modeling to support the elicitation of context-aware functionalities has been overlooked due to its high complexity. To help overcome this, we have implemented a data-driven process that analyzes contextual data and generates data-driven context models. Objective: We aim at investigating to which extent a data-driven context model supports the identification of more complex contexts (i.e., contexts that combine several contextual elements) and unexpected context-aware functionalities. Method: We used a one factor with two treatments randomized design with 13 experienced software engineers. Given a specific system-supported user task, the participants were asked to come up with requirements that describe context-aware functionalities to improve the user task. Results: Use of the data-driven context model increased the average number of contextual elements used to describe requirements from 1.77 to 4.23. No participant from the control group was able to identify by themselves any of the contexts included in the model. All comparisons between groups had sufficient effect size and power. The participants regarded the data-driven context model as a useful tool to support the elicitation of context-aware functionalities. Conclusion: The data-driven context model has shown potential to support the identification of relevant contexts for given user tasks.

Keywords: Context awareness · Data-driven · Model · Requirements · Experiment

1 Introduction

Computers are part of everyday life and, in recent decades, have become increasingly ubiquitous. The number of software-based solutions that surround us constantly increases. Just to mention the mobile world, millions of apps are readily available in a market where competitors yearn to release new features to their users as fast as possible

This work has been partially supported by CNPq, Brazil.

© Springer Nature Switzerland AG 2021

L. Ardito et al. (Eds.): PROFES 2021, LNCS 13126, pp. 119–135, 2021.

https://doi.org/10.1007/978-3-030-91452-3_8

– but not just any feature. Ideally, competitors want to deliver *delightful* features, which amaze and capture their audience. These features fall into the category of unconscious requirements [16]; as such, they are hard to elicit.

Context-aware functionalities are perceived as a way to delight users (e.g., [7, 8, 15]). They consider context to produce a certain system behavior, typically a recommendation or an adaptation. These types of features can be mapped to Dey’s definitions of the context-aware features “presentation” and “execution” [5], respectively. The elicitation of context-aware functionalities, in turn, demands context modeling, which involves an analysis of the relevance of contextual elements (CEs) (e.g., [4, 21]) and an analysis of combinations of CEs (e.g., [4, 10]) for a given user task (e.g., [5, 6]). According to practitioners, these are challenging steps and have been overlooked due to their high complexity: In a scenario with dozens of CEs, identifying which CEs influence a given user task, either individually or in combination with others, is time-consuming, non-intuitive, and error-prone [6].

Data-driven approaches have been named as promising to improve requirements engineering (RE) in general [14] and context modeling in particular [22]. Therefore, we formulate the following research question (RQ1): To which extent does a data-driven context model support the identification of more complex contexts and unexpected context-aware functionalities? We implemented a data-driven context modeling process to identify relevant contexts and create context models to support practitioners in the elicitation of context-aware functionalities. We used our implementation to generate a context model for the system-supported user task “*create a comment*” of DorfFunk¹, a communication app with characteristics of a social network that has approximately 25,000 active users. DorfFunk was developed and is maintained by Fraunhofer IESE. In this paper, we report on a controlled experiment carried out to verify to which extent the data-driven context model supports the identification of more complex contexts (i.e., contexts combining several CEs) and unexpected context-aware functionalities from contexts that, without the data-driven context model, would require more time to be identified.

This paper is organized according to the guidelines for reporting experiments in software engineering proposed by Jedlitschka et al. [12]. It is structured as follows: Sect. 2 discusses related work and summarizes the data-driven context modeling process; Sect. 3 contains the plan for the experiment; Sect. 4 describes the execution; Sect. 5 contains the analysis; Sect. 6 discusses the results; and Sect. 7 concludes the paper.

2 Background

Proposals for representing context models concerning RE are diverse; however, the challenge of identifying the relevant contexts remains high, independent of the chosen representation, especially when the context modeling activity is performed by humans. Alegre et al. [1], for example, surveyed existing RE modeling techniques for context-aware systems and mention no data-driven approach. Data-driven approaches have been used to improve context-aware systems, though: Saputri and Seok-Won [18] reviewed the use of machine-learning techniques (which are data-based approaches) in self-adaptive systems and found that in 41% of them, the purpose was to support modeling.

¹ <https://www.digitale-doerfer.de/unsere-loesungen/dorffunk/>.

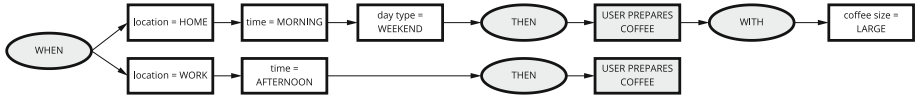


Fig. 1. Example of a data-driven context model.

Among these papers, only the work by Rodrigues et al. [17] is related to RE. In their work, data mining was employed to identify relevant contexts for dependable systems, which were later mapped manually to a contextual goal model [2]. Nonetheless, ad-hoc context modeling has been the state of the practice [6].

In our approach, we implemented a data-driven context modeling process introduced by Falcão [7] that creates context models to support the elicitation of context-aware functionalities. These models contain contexts that were found to influence a given user task. We define “context” as a set of instantiated CEs. For example, given a CE “time” and another CE “location”, examples of contexts include “afternoon” (1 CE: Time), “home” (1 CE: Location), and “afternoon at home” (combination of 2 CEs: Time and location). The more CEs we have in a context, the more complex the context is. In the data-driven process, contextual data is collected from available sources based on the user task in focus, and then processed to generate a context model. The data collection is manual and the data processing (including the model generation) is automated. We classified the CEs as continuous or categorical, and used statistical methods to search for correlations among them. The outcomes were transferred to a diagram referred to as “data-driven context model”, which is a directed acyclic graph with one root node. Each path from the root node towards a leaf describes how a context influences a user task of interest. Figure 1 shows an example. Consider that the user task is “Prepare a coffee”. Each of the two paths contains a set of instantiated CEs (white boxes) that, together, were found to influence the task (e.g., “When location = WORK and time = AFTERNOON then user prepares coffee”, i.e., the context “location = WORK and time = AFTERNOON” influences the user task “Prepare a coffee”, according to the model).

3 Experiment Planning

3.1 Goals

From RQ1, we derived the following study goals using the GQM template [3]:

- Goal 1** Increase the ability of practitioners to identify complex contexts to describe context-aware functionalities from the point of view of the researcher in the context of a controlled experiment with practitioners using a data-driven context model based on DorfFunk data.
- Goal 2** Improve the efficiency of the identification of relevant contexts from the researcher’s point of view in the context of a controlled experiment with practitioners using a data-driven context model based on DorfFunk data.
- Goal 3** Improve the effectiveness of the identification of relevant contexts from the researcher’s point of view in the context of a controlled experiment with practitioners using a data-driven context model based on DorfFunk data.

- Goal 4** Verify the usefulness of the data-driven context model from the point of view of practitioners in the context of a controlled experiment with practitioners using a data-driven context model based on DorfFunk data.

3.2 Design

The controlled experiment had a one factor with two treatments randomized design [23]. The primary factor was the context modeling technique. The participants of the treatment group were assigned to the data-driven context model, whereas the participants of the control group received the list of available CEs.

3.3 Participants

In practice, elicitation of context-aware functionalities is an activity that can be performed by a large range of professionals, including requirements engineers, UX designers, software architects, and developers, among others [6]. Therefore, we generalized our population as *software engineers with experience in information systems*. No prior experience in context awareness was required. We drew a convenient (non-probabilistic) sample of it in the Information Systems Division at Fraunhofer IESE. We invited all 34 professionals in the division through a corporate email list. Participation was voluntary. In order to motivate the invitees, they were informed that in the experiment they would have the opportunity to learn about requirements elicitation of context-aware functionalities and to participate in a practical activity about it. The informed consent form was attached to the email. We had 21 volunteers, which was our initial sample size (it was later reduced to 13 participants due to a deviation in the execution – see Sect. 4.2).

3.4 Participants' Task

The participants were asked to create requirements in written form, in English, to describe context-aware functionalities for one given system-supported user task of the app DorfFunk. The control group participants were told that they were constrained to stick to the list of CEs they had received to elaborate the requirements, whereas the treatment group participants were asked to stick to using the data-driven context model to elaborate theirs. Note that for the treatment group participants, using the data-driven context model implied that they would be constrained by the same CEs available to the control group participants. All participants were informed that there was no minimum number of requirements they should create.

3.5 Hypotheses and Variables

- **H1: Use of the data-driven context model influences the ability of individuals to elaborate requirements with more complex contexts.** The contexts used in the requirements varied regarding the number of CEs they combined. We wanted to check whether the requirements elaborated by the treatment group participants were more (or less) complex, with complexity measured by the number of CEs combined

to describe the context used in the requirement (see [Goal 1](#)). The independent variable was the context modeling technique (*ad-hoc* or data-driven), and the dependent variable was the number of CEs combined to describe the contexts of each requirement created by the individuals. The null and alternative hypotheses were formulated as follows: $H_{0_1} : \mu_{1\ control} = \mu_{1\ treatment}$ and $H_{1_1} : \mu_{1\ control} \neq \mu_{1\ treatment}$.

- **H2: Use of the data-driven context model influences the efficiency of individuals to identify the relevant contexts included in the data-driven context model.** We wanted to check whether the data-driven context model included relevant contexts that, within a limited amount of time, would not be found by the participants of either group (see [Goal 2](#)). The independent variable was the context modeling technique (*ad-hoc* or data-driven), and the dependent variable was the percentage of relevant data-driven contexts found by the individuals. The null and alternative hypotheses were formulated as follows: $H_{0_2} : \mu_{2\ control} = \mu_{2\ treatment}$ and $H_{1_2} : \mu_{2\ control} \neq \mu_{2\ treatment}$.
- **H3: Use of the data-driven context model influences the effectiveness of individuals to identify the relevant contexts included in the data-driven context model.** We wanted to verify whether the data-driven context model included relevant contexts that, within a limited amount of time, would be used more often by individuals of either group (see [Goal 3](#)). The null and alternative hypotheses were formulated as follows: $H_{0_3} : \mu_{3\ control} = \mu_{3\ treatment}$ and $H_{1_3} : \mu_{3\ control} \neq \mu_{3\ treatment}$.
- **H4: The data-driven context model is perceived by individuals as a useful instrument to support the elicitation of context-aware functionalities.** The data-driven context model is a new artifact aimed at improving the way individuals elicit context-aware functionalities. We wanted to verify how much the participants valued it (see [Goal 4](#)).

3.6 Experimental Materials

Informed consent: The informed consent form contained partial disclosure of the experiment in order to prevent undesirable change of behavior by the participants [23]².

Briefing questionnaire: This contained three questions to characterize their professional experience. They were asked about the number of years of professional experience (less than 5 years, or 5 years or more), whether their professional experience included requirements elicitation (yes/no), and to which role most of their professional experience was related.

Instructions about the data-driven context model: The treatment group participants received in advance a PDF file introducing the syntax and semantics of the data-driven context model they would use during execution of the experiment.

Introductory presentation: A set of slides introducing the participants to the experiment was presented. The slides contained information about the concept of context awareness, examples of context-aware functionalities in well-known applications, a description of context modeling activities performed to support the elicitation of context-aware functionalities, and a presentation of the participants' task in the experiment. It included a video introducing the app DorfFunk, for which the participants were to create requirements.

Data-driven context model: During the execution

² All materials are available at <https://doi.org/10.5281/zenodo.5090748>.

of the experiment, the treatment group participants received a PDF file containing the data-driven context model generated based on the analysis of contextual data of the app DorfFunk, with focus on a specific user task. **List of CEs:** During the execution of the experiment, the control group participants received a PDF file containing the list of CEs they could use to elaborate context-aware functionalities. There were 15 CEs. **Debriefing questionnaire:** Right after the participants had performed their task in the experiment, they were asked to answer a debriefing questionnaire about the experience. Three questions were posed to all participants: whether their task in the experiment was clearly explained; whether the time they had to participate was adequate; and whether they perceived their task as easy or not. They were also asked about how well they knew the app DorfFunk, because we regarded the amount of previous knowledge about the app as a possible confounding factor. The treatment group participants were asked additional questions about the data-driven context model they received to support the task. For this purpose, we employed the UTAUT (Unified Theory of Acceptance and Use of Technology [20]), tailoring the items to our case. **Requirements validation checklist:** We defined and used a checklist to guide the validation of the requirements generated by the participants in the analysis phase (see Sect. 5.2).

3.7 Procedure

The execution of the experiment was subdivided into three parts. In the first part, they were introduced to the concepts of context awareness, elicitation of context-aware features, and context modeling using the “introductory presentation”. Then, they received a step-by-step introduction to the tool they would use to provide data during the execution of the experiment. At the end of the introduction, they were informed that the specific system-supported user task they should try to improve using context awareness was “*create a comment*”, and a story board illustrating the as-is situation of the system-supported user task was shown. The user task was revealed only at the end of the introduction in order to prevent the participants from thinking about possible context-aware functionalities in advance. In the second part, the participants had 30 min to perform their task (see Sect. 3.4). When the time was over, they were informed and proceeded to the last part, the debriefing questionnaire. At the end of the debriefing questionnaire, the control group participants were told that some participants received different artifacts to perform the task, and were given the opportunity to download these artifacts, namely the instructions about the data-driven context model and the data-driven context model. Due to the COVID-19 pandemic [24], all sessions were performed online via MS Teams³. In order to ensure that the same instructions would be provided to all participants across the sessions, the instructions were written down before and read at the time of the execution. After the completion of the questionnaire, they were asked to keep the participation confidential, since other participants would join different sessions in different date/times.

³ <https://www.microsoft.com/en-us/microsoft-teams/group-chat-software>.

4 Execution

4.1 Preparation

Before the execution of the experiment, the participants were asked via email to answer the briefing questionnaire about their professional experience (see Sect. 3.6). The input was used to block participants before the randomization procedure. There were two blocks: those with less than 5 years of professional experience ($N = 6$), and those with 5 years or more of experience ($N = 15$). The other questions (experience with elicitation and major role) were not used to block the participants because they led to very small odd blocks in some cases, which would compromise the randomization. Within the blocks, the participants were randomly assigned to either the control group ($N = 11$) or the treatment group ($N = 10$). The treatment group participants received an email with a set of slides containing instructions about the data-driven context model, so they would know how to use it in the experiment. They were asked to read the material before the execution and to keep all information confidential.

Next, all participants were asked to provide their availability to participate in the experiment. Multiple time slots were offered over a period of two weeks. Based on their responses, we prepared and executed eight sessions between 18 and 27 May 2021.

4.2 Deviations

On the first day (18 May 18 2021), there were two sessions, with 9 participants ($p_1 - p_9$) in total, 5 of them belonging to the treatment group. At the end of the day, we performed a data validation to check the input provided by the participants. We noticed that the treatment group participants had apparently deviated from their task, as the results indicated that they did not use the data-driven context model as intended. Furthermore, there was a question to be answered in the debriefing questionnaire on a 5-point Likert-like rating scale about how clearly their task in the experiment was explained, and 60% of the treatment group participants did *not* agree that their task was clearly explained. In fact, one participant contacted the moderator via private chat during the execution to ask how they should use the data-driven context model. In addition to that, we had the chance to talk to one participant after the experiment and they confirmed that they did not quite understand how they should use the data-driven context model to perform their activity. For this reason, we changed the procedure of the experiment for the future treatment group participants: They would be explicitly introduced to the material “Instructions about the data-driven context model” (see Sect. 3.6), which they received before the experiment to read by themselves. For the control group participants, the procedure was not changed. In none of the remaining six sessions, participants from both groups took part, so we could change the procedure for the treatment group participants without having to reschedule sessions. As a consequence of the early data validation, where some participants were found to have misunderstood the task, the data collected should be considered invalid [23]. As we changed the instructions for the next treatment group participants, we also removed from the analysis all 5 subjects who had received the original instructions. Of these, 3 belonged to the less experienced block. The other 3 less experienced participants belonged to the control group, and were therefore also

removed from the analysis; otherwise we would have had less experienced professionals only in the control group. For that reason, we were left with 13 participants (p_4 who participated in the first session, and $p_{10} - p_{21}$ who participated in the later sessions), all of them professionals with 5 years or more of experience.

5 Analysis

We used quantitative methods to analyze the data. For the briefing and debriefing questionnaires, we used central tendency and data visualization measures. The contexts used in each valid requirement were extracted. If these contexts contained any of those included in the data-driven context model, the corresponding contexts included in the models were classified as *relevant data-driven contexts*. Once the contexts were extracted, we were able to calculate the average number of CEs in the context of each group, as well as the effectiveness and efficiency of each participant in identifying the relevant data-driven contexts. We compared both groups using statistical tests and verified the effect size using Hedges's g , a *d family* effect size measure recommended for small sample sizes [13].

5.1 Descriptive Statistics

Among the 13 participants, 12 (92.3%) reported that their professional experience included requirements elicitation. Figure 2a shows the participants' main professional role, organized by group. There was a prevalence of RE-related professionals (requirements engineers, UX designers) in the treatment group. With respect to the participants' previous knowledge about the app DorfFunk, Fig. 2b shows that the control group participants were more familiar with it (more subjects were users of the app or even participated in the development team).

We asked the participants to rate their agreement with the statement “My task in this experiment was clearly explained”. For this purpose, we used a 5-point Likert-like scale (Strongly disagree, Disagree, Neutral, Agree, and Strongly agree) and found that 11 participants (84.6%) strongly agreed and 2 (15.4%) agreed (*median = mode = “Strongly agree”*). Similarly, another statement was formulated with respect to the time the participants had to perform their task (see Fig. 2c). The mode and the median for the control group participants was “Strongly agree”, whereas for the treatment group participants, both mode and median were “Neutral/Agree”. Finally, a statement was formulated to get the participants' perception of the ease of their task. As can be seen in Fig. 2d, the mode and the median for both groups were “Neutral”; however, for the control group participants, the distribution of responses was spread wider, varying from “Strongly disagree” to “Strongly agree”, whereas for the treatment group, it varied from “Disagree” to “Agree”.

5.2 Data Set Preparation

The participants generated a total of 105 requirements; 55 (52.4%) came from the control group and 50 (47.6%) from the treatment group. Before carrying out hypothesis

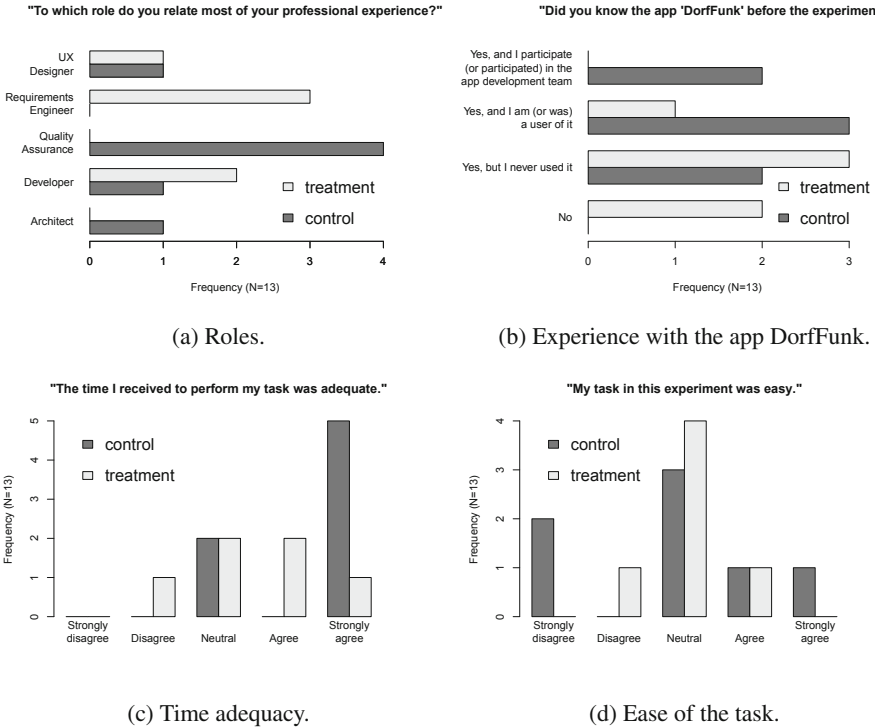


Fig. 2. Frequency of responses in the debriefing questionnaire.

testing, it was necessary to validate the requirements. As validation technique, we used a checklist with quality criteria for requirements as presented by Pohl and Rupp [16]. According to them, this technique improves the reproducibility of the validation. Our checklist covered the following quality criteria (derived from [16] and [11]), which were applied sequentially:

1. **Agreed:** A requirement should be directly related to the system-supported user task in focus. Moreover, it should describe a context-aware feature – in our case, either a recommendation or adaptation based on the context. In total, 24 requirements (58.3% of them from the treatment group) failed in this criterion, e.g., “The system should ask the user to turn on the notification of DorfFunk when the user is in the home network after 17:00” (no direct connection to the user task “Create a comment”).
2. **Feasible:** A requirement should only use CEs available in the model; otherwise they would be technically unfeasible. In total, 5 requirements (80% of them from the treatment group) failed in this criterion, e.g., “User in the role of suggestion worker should have predefined answers based on *the content of the suggestion*.” (The content of the suggestion was not available).
3. **Necessary:** A requirement must be currently applicable to the app DorfFunk. In total, 15 requirements (53.3% of them from the treatment group) failed in this crite-

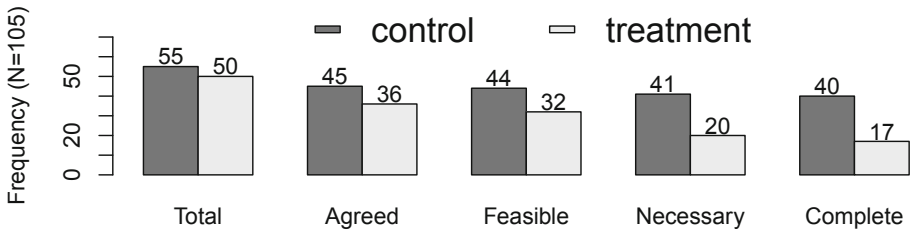


Fig. 3. Frequency of requirements after each validation step, by group.

rior, e.g., “If user is the author of a post, the system should set the type of post to clarification when the user writes a new post for the same event, but allow the user to change it if necessary”. (There is no post type “clarification”).

4. **Complete:** A requirement is described in such a way that no additional information is needed to understand it. In total, 4 requirements (75% of them from the treatment group) failed in this criterion, e.g., “The system should give a suggestion for content type when posting a comment on an event”. (There is no information about what criteria should be used to give the suggestion).

In the introduction of the experiment, the participants were indirectly informed about the criteria 1, 2, and 4. After the validation steps, we had 57 valid requirements, 29.9% of them from the treatment group (see Fig. 3). In this process, participant p_{21} (treatment group) was removed because none of their 6 requirements were valid (two were not feasible and four were not necessary). The validation was performed by the first author, who is an experienced software engineer specializing in context awareness and with a broad understanding of the app DorfFunk. Whenever doubts came up concerning the criterion “necessary”, the opinion of the product owner was heard. Next, we extracted the contexts of these 57 requirements and counted for each how many CEs were used to compose the context. Then we checked which of these contexts were included in the data-driven context model in order to define a list of relevant data-driven contexts according to the participants. The number of CEs used to describe contexts as well as the list of relevant data-driven contexts were used to support hypothesis testing. Due to space limitations, the raw data is not included here⁴.

5.3 Hypothesis Testing

H1 (“more complex contexts”). The mean number of CEs that the control group participants used to describe their 40 contexts was $\mu_{1\ control} = 1.77$ ($SD = 0.80$, $min = 1$, $max = 5$, $median = 2$), whereas the mean from the treatment group (17 contexts) was $\mu_{1\ treatment} = 4.23$ ($SD = 0.97$, $min = 3$, $max = 5$, $median = 5$)⁵. We applied the Shapiro-Wilk normality test to the distributions and rejected the hypothesis of normality (control: $W = 0.73036$, $p\text{-value} = 3.183e - 07$; treatment:

⁴ The anonymized raw data is available at <https://doi.org/10.5281/zenodo.5090748>.

⁵ For H1, H2 and H3, we used $\alpha = 0.05$ as significance level and $\beta = 0.2$.

$W = 0.66011$, $p - value = 4.226e - 05$). Then we applied the non-parametric Wilcoxon rank sum test to compare the distributions and found that the difference between the means was significant ($p - value = 7.071e - 09$); consequently, H_{0_1} can be rejected. The effect size was large (Hedges's $g = 2.84$, 95% confidence interval: 2.06 to 3.62). We did a post-hoc power analysis and found power $1 - \beta = 0.9880658$; therefore, we can accept H_{1_1} . Conclusion: Individuals who had the model used more complex contexts ($\mu_{1\ treatment} > \mu_{1\ control}$) to elaborate their requirements.

H2 and H3 (“efficiency” and “effectiveness”). In total, 6 contexts included in the data-driven context model were used by participants to describe their context-aware functionalities. None of these contexts were found by participants of the control group, so $\mu_{2\ control} = \mu_{3\ control} = 0\%$. The efficiency and the effectiveness of the treatment group participants is shown in Table 1. The mean efficiency of the treatment group in identifying the relevant data-driven context was $\mu_{2\ treatment} = 36.6\%$ ($SD = 0.217$), whereas their effectiveness was $\mu_{3\ treatment} = 74.6\%$ ($SD = 0.347$). As we cannot assume normal distribution from the control group, we again used the Wilcoxon rank sum test to compare the differences and found that H_{0_2} and H_{0_3} were rejected ($p - value = 0.002033$ and $p - value = 0.001645$, respectively). In both cases, the effect size was large (efficiency: Hedges's $g = 2.46$, 95% confidence interval: 0.87 to 4.05; effectiveness: Hedges's $g = 1.74$, 95% confidence interval: 0.33 to 3.14). We did a post-hoc power analysis and found for efficiency power $1 - \beta = 0.9556588$, and for effectiveness power $1 - \beta = 0.7392546$; therefore, we can accept H_{1_2} and cannot accept H_{1_3} . Conclusion: Individuals who had the model were more efficient ($\mu_{3\ treatment} > \mu_{3\ control}$). No conclusion can be drawn about effectiveness.

H4 (“a useful instrument”). With respect to the usefulness of the data-driven context model, the participants provided their perception regarding 18 statements adapted from UTAUT [20], covering the following aspects: performance expectancy (PE, 3 questions), effort expectancy (EE, 4 questions), attitude toward using technology (AT, 4 questions), self-efficacy (SE, 4 questions), and anxiety (AX, 3 questions). We coded the answer options numerically (1 to 5). We reversed the code in item AT.1 of aspect AT, and in all items of aspect AX, for they were stated in a negative way. Figure 4 shows the distributions of the responses of each aspect.

Table 1. Efficiency and effectiveness of the treatment group participants.

Participant ID	Modeled contexts used	Total contexts used	Efficiency	Effectiveness
p16	1	3	16.7%	33.3%
p17	3	3	50%	100%
p18	1	1	16.7%	100%
p19	2	5	33.3%	40%
p20	4	4	66.6%	100%

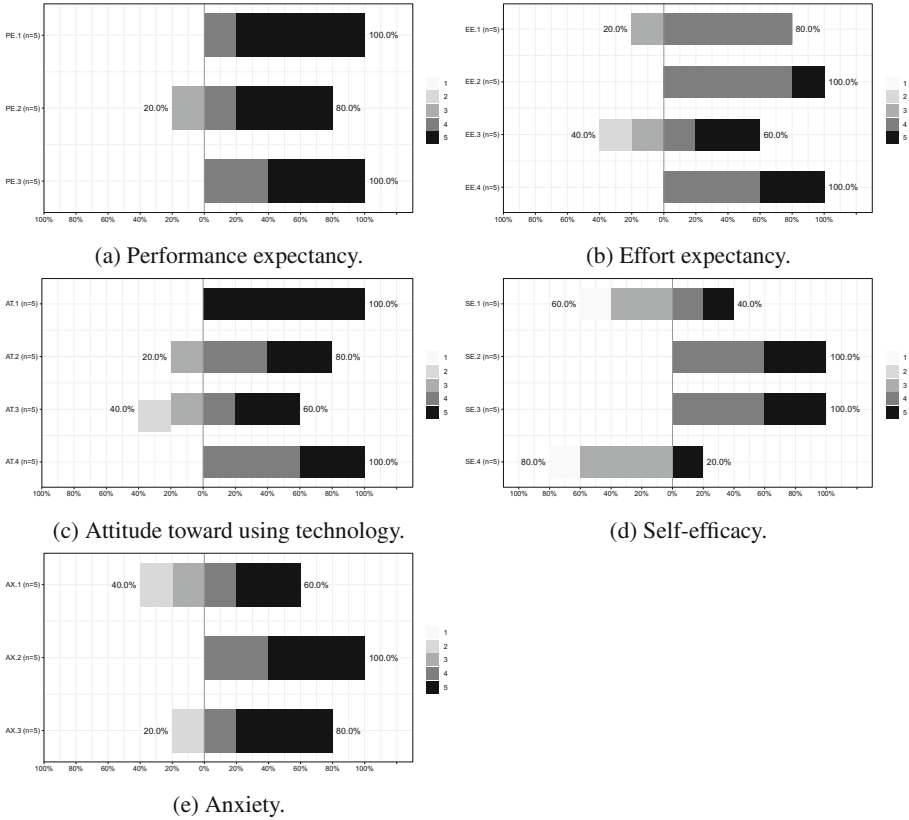


Fig. 4. Participants’ assessment with respect to usefulness of model.

Table 2 shows the scores of each aspect and their reliability. The item scores were calculated based on the mean value of the participants’ ratings; the aspect score was calculated as the mean of the item scores composing each aspect. To verify the reliability of the scores, we calculated their Cronbach’s alpha, which was higher than 0.7 for PE, AT, SE, and AX, hence acceptable [9], whereas for EE it was not acceptable. Conclusion: Apart from the aspect Self-efficacy (SE), all aspects had high scores (> 0.4), meaning that the participants evaluated the model positively.

Table 2. The five investigated aspects of usefulness of the data-driven context model.

Aspect	Item	Item score	Aspect score	Cronbach's alpha
PE - Performance expectancy	PE.1	4.8	4.6	0.778
	PE.2	4.4		
	PE.3	4.6		
EE - Effort expectancy	EE.1	3.8	4.05	0.468
	EE.2	4.2		
	EE.3	3.8		
	EE.4	4.4		
AT - Attitude toward using technology	AT.1	5	4.34	0.804
	AT.2	4.2		
	AT.3	3.8		
	AT.4	4.4		
SE - Self-efficacy	SE.1	3.2	3.75	0.752
	SE.2	4.4		
	SE.3	4.4		
	SE.4	3		
AX - Anxiety	AX.1	3.8	4.2	0.903
	AX.2	4.6		
	AX.3	4.2		

6 Discussion

6.1 Evaluation of Results and Implications

The data-driven context model (and the process that creates it) is not about completeness: There is no claim suggesting that *all* relevant contexts for the user task in focus would be included in the model. Furthermore, these contexts were not said to be necessarily *better* than others. The point is that the data-driven context model is able to provide shortcuts in the solution space of a complex activity: context modeling. We confirmed in the experiment that, with its support, individuals were able to find some *unexpected* contexts – and to do so *faster*, for with unlimited time, we expect that individuals would eventually find these contexts anyway (which is a hypothesis to be investigated). As participants who used the data-driven context model were able to elaborate requirements with more complex contexts, we accepted H1. Our findings back the results of a previous empirical study performed by Falcão et al. [6], where they found that practitioners regard the analysis of combinations of CEs as a highly complex activity. We observed this in our experiment, as a significantly lower number of CEs were found in the contexts of the control group. In addition, this shows that the data-driven context model contained meaningful contexts with a larger number of CEs.

We measured the participants' efficiency and effectiveness in finding relevant data-driven context models included in the model. Although we had expected that the treatment group participants would perform better since they received the data-driven context model, we assumed that some of the contexts found in the data-driven context model would be "too obvious", and therefore participants of the control group would be able to identify them despite not having received the data-driven context model (and this expectation would be even higher if we knew in advance that the control group participants had better background knowledge about the app DorfFunk – see Sect. 5.1). For this reason, it was unexpected for us that not a single control group participant was able to identify by themselves any of the contexts included in the data-driven context model, which led to zero efficiency and zero effectiveness in identifying such contexts. It is worth noting that the result would be the same if we had considered all requirements generated by the control group before reducing the data set as presented in Sect. 5.2. Therefore, the treatment group participants were more efficient in finding relevant contexts included in the data-driven context model, which speaks (at least partially) in favor of the representation of the model and its content. We also found higher effectiveness, but without statistical power, which we believe was due to the sample size, since the effect size was large.

In fact, in all tests performed (H1, H2, and H3), the effect size can be considered large (according to Kampenes et al. [13]); however, for effectiveness (H3), it is worth noting that the 95% confidence interval ranged from small ($g \leq 0.376$) to large ($g \geq 1.002$) effect sizes, and we could not accept H_{13} . This may have been caused by participant p_{16} , who created 3 valid requirements, but 2 of them could not be directly mapped to the data-driven context model, even though they were clearly inspired by contexts presented in the model. This added to the practical utility of the model, but is not captured by the pragmatic evaluation we chose (e.g., in one case, p_{16} cleverly inverted a context described in the model to describe a context-aware functionality). In order to be consistent with our analysis procedure and give less room to subjectivity, we did not include such cases in the baseline of relevant data-driven contexts.

We generally found a positive trend towards acceptance of the data-driven context model in all aspects investigated, as can be seen in Fig. 4. One exception could be the aspect "self-efficacy", which had the lowest score. To a certain extent, it reflects our need to adjust the procedure after the first two sessions of the experiment (see Sect. 4.2), as we realized that the participants needed additional explicit instructions to understand how to use the data-driven context model. With respect to effort expectancy, the aspect whose reliability was not acceptable (see Sect. 5), we found the cause in item EE.1, which referred to how clear and understandable the usage of the data-driven context model would be. If EE.1 was removed, the aspect score would be 4.13 and the reliability would be acceptable (0.733).

Regarding RQ1, we conclude that the approach supports the identification of more complex contexts as well as unexpected functionalities. In a practical setting, we think the data-driven context model should be used to support creativity group sessions, providing individuals with triggers and input that can be copied as-is, combined, or transformed (the basic elements of creativity [19]) to generate new ideas – as happened, for example, with participant p_{16} . Working together, individuals will be in a better position

to evaluate the meaning of contexts, judge their value, discard what is irrelevant, and leverage the shortcuts to insightful contexts provided by the model. Although, on the one hand, the automated process helps people identify some relevant contexts faster, on the other hand human participation remains a fundamental piece in the validation and in the creativity steps that follow context modeling. Moreover, we expected the outcomes to be more sound when contextual data is collected with context modeling in mind (in the experiment, we used available data for the sake of convenience) and when more CEs are included in the analysis, which is expected in a smart scenario [6,7].

6.2 Threats to Validity

Construct validity: *Mono-operation bias:* All participants performed the same task within the context of the same application, meaning that the results can reflect the particularities of the specific setting. *Confounding constructs or level of constructs:* The data-driven context model was created based on available CEs. A lower effect is expected, as fewer CEs were involved, whereas the results may be more pronounced, as more CEs were included in the analysis. *Restrict generalizability across constructs:* The process of generating the data-driven context model may have had an impact on the overall efficiency of the elicitation process. **Internal validity:** *Interaction with selections:* In the random assignment of participants to groups, those with major background experience in RE-related roles were concentrated in the treatment group, which may have benefited the group. On the other hand, the control group had participants with more prior knowledge about DorfFunk, which could also have been advantageous for the group. **External validity:** *Interaction between selection and treatment:* It is possible that the selection was not representative and the results are not generalizable, given the number of participants and the sample strategy. Moreover, if participants with particular expertise in the elicitation of context-aware functionalities had been involved, the outcomes might be different. However, such professionals belong to a hard-to-spot population [6]. **Conclusion validity:** *Reliability of measures:* The participants provided written requirements, and there was a manual activity to prepare the data set (see Sect. 5.2), extract contexts from the requirements, and count the number of CEs of each context. Mistakes in these manual steps, especially in the preparation of the data set, may have compromised the conclusions. *Experimental setting:* The experiment was performed in online sessions via the Internet. In such settings, it is harder to check the participants' compliance with the procedure, in particular to ensure that they focus exclusively on the activity during the experiment.

7 Conclusion and Future Work

Context modeling to support the elicitation of context-aware functionalities is a needed but rather overlooked activity due to its complexity, especially concerning the analysis of relevance and combinations of contextual elements. We implemented a semi-automated data-driven approach to analyze contextual data and generate context models that revealed relevant contexts that influence a given user task of interest. We used data from an app in use to create a data-driven context model. In this paper, we reported

its evaluation in the context of a controlled experiment with experienced software engineers. The results showed that participants using the data-driven context model were able to describe requirements with more complex contexts, while participants without the context model were not able to identify any of the relevant contexts included in the model. Moreover, participants using the data-driven context model regarded it as a useful instrument to support the elicitation of context-aware functionalities. To the best of our knowledge, this paper contributes the first controlled experiment evaluating the usage of data-driven context modeling on the elicitation of context-aware functionalities.

A limitation of the data-driven approach is that it can only draw conclusions from existing data, and the universe of relevant combinations of CEs is much larger than what can be inferred via data analysis. Therefore, it must be clear that the data-driven context model reveals *some* relevant contexts, but never *all* of them. The data-driven context model anticipates the identification of relevant contexts for a given user task, meaning that part of the time-consuming work of analyzing relevant combinations can be skipped. Nonetheless, implementing the data-driven context modeling approach introduces effort as well; however, this is operational effort, which takes place in polynomial time, taking the list of CEs as input, whereas the analysis of relevant combinations is creative work requiring exponential time with the same input.

We want to run the experiment again with less experienced participants in order to check whether it can attenuate the experience factor in the elicitation of context-aware functionalities. We also consider it essential to generate the data-driven context model using a different application in order to mitigate the mono-operation bias. Furthermore, we want to perform a case study where the participants would work together to create context-aware functionalities based on the data-driven context model. Finally, we plan to modify the data processor using different algorithms and evaluate which strategy reveals more relevant contexts – which might lead to a multi-strategy approach.

References

1. Alegre, U., Augusto, J.C., Clark, T.: Engineering context-aware systems and applications: a survey. *JSS* **117**, 55–83 (2016)
2. Ali, R., Dalpiaz, F., Giorgini, P.: A goal-based framework for contextual requirements modeling and analysis. *Requirements Eng.* **15**(4), 439–458 (2010)
3. Basili, V., Caldiera, G., Rombach, H.D.: The goal question metric approach. In: *Encyclopedia of Software Engineering*, pp. 528–532 (1994)
4. Bauer, C., Dey, A.K.: Considering context in the design of intelligent systems: current practices and suggestions for improvement. *JSS* **112**, 26–47 (2016)
5. Dey, A.K.: Understanding and using context. *PUC* **5**(1), 4–7 (2001). <https://doi.org/10.1007/s007790170019>
6. Falcão, R., Villela, K., Vieira, V., Trapp, M., Faria, I.: The practical role of context modeling in the elicitation of context-aware functionalities: a survey. In: *RE 2021*. IEEE (2021)
7. Falcão, R.: Improving the elicitation of delightful context-aware features: a data-based approach. In: *RE 2017*, pp. 562–567. IEEE (2017)
8. Google: Google Awareness API (2021). <https://bit.ly/3wmoF56>. Accessed 08 July 2021
9. Hair, J.F., Black, W., Babin, B., Anderson, R.: *Multivariate Data Analysis*. Pearson (2009)

10. Henriksen, K.: A framework for context-aware pervasive computing applications. Ph.D. thesis, The University of Queensland (2003)
11. ISO/IEC/IEEE 29148 - Systems and Software Engineering - Life cycle processes - Requirements engineering. Standard, ISO (2018)
12. Jedlitschka, A., Ciolkowski, M., Pfahl, D.: Reporting experiments in software engineering. In: Shull, F., Singer, J., Sjøberg, D.I.K. (eds.) *Guide to Advanced Empirical Software Engineering*, pp. 201–228. Springer, London (2008). https://doi.org/10.1007/978-1-84800-044-5_8
13. Kampenes, V.B., Dybå, T., Hannay, J.E., Sjøberg, D.I.: A systematic review of effect size in software engineering experiments. *IST* **49**(11–12), 1073–1086 (2007)
14. Maalej, W., Nayebi, M., Johann, T., Ruhe, G.: Toward data-driven requirements engineering. *IEEE Softw.* **33**(1), 48–54 (2016)
15. Olsson, T., Lagerstam, E., Kärkkäinen, T., Väänänen-Vainio-Mattila, K.: Expected user experience of mobile augmented reality services: a user study in the context of shopping centres. *PUC* **17**(2), 287–304 (2013). <https://doi.org/10.1007/s00779-011-0494-x>
16. Pohl, K., Rupp, C.: *Requirements Engineering: Fundamentals, Principles, and Techniques*, 2nd edn. Rocky Nook, San Rafael (2015)
17. Rodrigues, A., Rodrigues, G.N., Knauss, A., Ali, R., Andrade, H.: Enhancing context specifications for dependable adaptive systems: a data mining approach. *IST* **112**, 115–131 (2019)
18. Saputri, T., Lee, S.W.: The application of machine learning in self-adaptive systems: a systematic literature review. *IEEE Access* **8**, 205948–205967 (2020)
19. Trapp, M.: Creative people are great thieves with lousy dealers (2020). Proceedings <http://ceur-ws.org> ISSN 1613, 0073
20. Venkatesh, V., Morris, M.G., Davis, G.B., Davis, F.D.: User acceptance of information technology: Toward a unified view. *MIS Q.* **27**, 425–478 (2003)
21. Vieira, V., Tedesco, P., Salgado, A.C., Brézillon, P.: Investigating the specifics of contextual elements management: the CEManTIKA approach. In: Kokinov, B., Richardson, D.C., Roth-Berghofer, T.R., Vieu, L. (eds.) *CONTEXT 2007*. LNCS (LNAI), vol. 4635, pp. 493–506. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74255-5_37
22. Villela, K., et al.: Towards ubiquitous RE: a perspective on requirements engineering in the era of digital transformation. In: *RE 2018*, pp. 205–216. IEEE (2018)
23. Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A.: *Experimentation in Software Engineering*. Springer, Heidelberg (2012). <https://doi.org/10.1007/978-3-642-29044-2>
24. World Health Organization: COVID 19 - GLOBAL. <https://bit.ly/3AI5RR0>. Accessed 21 June 2021