Designing Context-Aware Chatbots for Product Configuration

Tom Niederer[1], Daniel Schloss[1(✉)], and Noemi Christensen[2]

[1] Karlsruhe Institute of Technology, Kaiserstraße 12, 76131 Karlsruhe, Germany
daniel.schloss@kit.edu
[2] CAS Software AG, CAS-Weg 1 - 5, 76131 Karlsruhe, Germany

Abstract. Product configurators provide an interface for customizing complex products. However, large form-based configurators overwhelm many end users and are often considered expert tools. This paper therefore addresses the problem of the complexity of current product configurators. Since chatbots can respond flexibly to queries and offer a natural language interface, they have the potential to simplify the configuration process. In this paper, we present a chatbot for product configuration that we developed using the design science research approach and in collaboration with an industrial partner. We derive design principles for configurator chatbots from user interviews that relate in particular to the flexibility of the chatbot compared to a static process. These design principles were implemented in our chatbot artifact which was evaluated in an online experiment (N = 12) and compared to a baseline chatbot with an inflexible configuration process. Our results indicate that the proposed design increased dependability and configuration performance, and overall had positive effects on participants' engagement. Thus, this study contributes prescriptive knowledge on the design of context-aware chatbots for product configuration and a novel artifact in the form of a context-aware configurator chatbot prototype.

Keywords: Chatbots · Product configuration · Context-awareness

1 Introduction

Product configurators emerged in the course of digitalization, when the demand for customized products grew, and "Mass Customization" became a phenomenon [1]. For companies, product configurators offer a good opportunity to organize the increasing demand for tailored products. The industry partner of the research project presented in this paper, CAS Software AG, offers product configurators for this purpose. Although CAS has been successfully developing Configure, Price, Quote (CPQ) solutions for midsize and large customers for 35 years, product configurators present a major design challenge due to the high number of configuration items and variants. Due to their specificity, software-based product configurators are often considered expert tools and cannot be easily used by customers and sales staff. The term "Mass Confusion" [2] describes a major issue occurring in complex customization settings. Customers can be overwhelmed by the number of available options and the complexity of the product structure. Information overload can deter users from choosing at all. Even small-scale configuration problems often have complex knowledge bases, e.g. when a product consists of components that can have sub-components or similar. Component interdependencies, restrictions, and rules introduce additional complexity [3]. In such settings, users are exposed to a multi-layered configuration process that involves many steps, many options to select from, and hard-to-track product domain restrictions. Researchers from the domain of product configuration as well as the CAS Software AG are constantly looking for ways to reduce complexity for end-users.

Studies in various information system domains have shown that natural language interaction can reduce complexity for end-users (e.g. [4, 5]). Natural language interfaces provide even inexperienced users with the capabilities to formulate requests in areas where they lack proficiency. They can reduce the need for experts to translate queries into domain-specific technical terms. For these reasons, natural language interfaces are also of interest for the special application of product configuration where managing complexity is the central challenge. Due to their ability to engage in natural language, chatbots could be a suitable tool to facilitate complex and lengthy processes like product configuration for the user.

However, while a configuration process is usually mapped to a stringent and linear scenario, a conversation is not. It can consist of contextual queries, surprising turns, interim questions, and even uncooperative behavior. Within a configuration dialogue, several scenarios might occur where the bot needs context to react appropriately. A customer might ask "What are my options?" or "Why not green?" instead of specifying the desired color. Here the proposed chatbot should be able to connect this follow-up question to the narrow context. End-users might also reference the broad context or change their mind: "Actually, I prefer red" at any moment during the configuration. Finally, modern chatbots possess the ability to extract multiple pieces of information within a single request [6]. The process could thus be accelerated, if users know their preferences beforehand: "I want a blue sportscar with high-end hi-fi interior". Due to the outlined characteristics of product configuration and natural language communication, a chatbot as a product configurator must have a high degree of flexibility.

Context-aware chatbots are intelligent agents that can consider contextual knowledge to simplify the human-machine interaction. This work sees the concept of context-awareness as a tool to provide the required adaptability and a fit between task and information presentation. A context-aware chatbot holds the potential to ultimately lead to a higher task performance for users [7, 8]. This work therefore investigates the design of a context-aware chatbot for product configuration. Our research project, conducted as a Design Science Research (DSR) project, aims to answer the following research questions:

- RQ1: How to design a context-aware chatbot for product configuration to facilitate the configuration process for the user?
- RQ2: How does a context-aware configuration chatbot compare to a basic question-answer chatbot regarding usefulness, ease of use, and cognitive load?

2 Theoretical Background and Related Work

2.1 Product Configurators

Since product configurators offer valuable opportunities for companies on the one hand, but also pose challenges, product configuration research has been devoted to the problems of configurators and addresses them with design proposals. To address the "Mass Confusion" [2] concern, product configurator research suggests teaching the customer about product attributes and their mapping to design parameters [6]. They also stress the importance of identifying customer needs. Customers might not be interested in exploiting all possible alternatives [6]. Furthermore, customers differ in their knowledge about the product to configure. For these reasons the configuration process can lead to a cognitive overload of information. Since a too high cognitive load can negatively influence task performance, in this case on the configuration task [7], an adequate representation format is crucial for the cognitive fit and the performance in a configuration process [2].

So far, only few technologies for assisting a configuration process have been tested. Most software based-configuration interfaces are available to customers as web forms [3, 5, 8]. However, they come with problems like limited flexibility and intuitiveness as well as complexity, which will be discussed in more detail in the subsequent section. For this reason, we are testing a chatbot as a new technology for product configuration, which reflects chatbot-specific but also configuration-specific design principles, as we outline in the following sections.

2.2 Conversational Interfaces for Complexity Reduction

Chatbots are a popular, but novel technological possibility to display a configuration process [9]. Due to the intuitiveness and possible efficiency of natural language, chatbots hold the potential to solve and facilitate complex tasks like product configuration. Therefore, the technology has already been tested in related areas with high complexity like e-commerce and recommendations, with the focus on assisting users in purchase decisions. Several studies use chatbots to query product databases based on communicated user preferences [10–12]. Natural language interfaces have also been applied to complex data scenarios. In recommender systems, conversational interfaces have been used to avoid information overload [5, 13]. Since it is very common for people to ask other people for recommendations (e.g. restaurants, movies) they are very familiar with formulating recommendation requests in natural language [13]. This also holds true for configuration, as the most intuitive process would be a product configuration in form of a consultation with a product expert or a sales representative.

2.3 Context-Awareness in Chatbots

A simple question-answer form would not be sufficient to facilitate configuration as this approach solely moves cognitive efforts from clicking to typing. The provision of an intuitive configuration interface is achieved by equipping the chatbot with context-aware capabilities. Context is needed where a statement relates not only to a single question but to the entire dialogue. Context-awareness enables a more natural conversational flow, which supports the call for integrating more social features in conversational agents [14]. Product configuration is context-dependent per se, since a single step is meaningless without reference to the others. In chatbot research context has been defined as extracted information from the conversation between user and chatbot [15]. There are also other derivations known, e.g. when someone does not provide requested information and context is needed to react appropriately: Corrections, references to the broad context, and references to the narrow context [16].

For a configuration bot to offer contextual knowledge, it must be connected to a memory and knowledge base. This is the case with our chatbot, which is connected to the configuration interface of CAS in order to map the hierarchical configuration steps and the respective configurations. Furthermore, modeled configuration restrictions can be used to inform the user about available options. Modeled additional information on specific product characteristics can be retrieved to answer follow-up questions. Furthermore, the bot could offer the possibility to skip certain options and apply default configuration values. Finally, product experts could formulate several desired configuration properties within a single request, while less sophisticated users are guided step by step through the process.

3 Research Approach

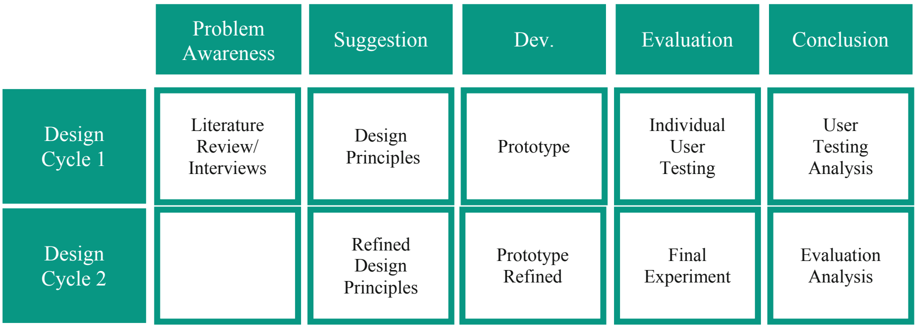

Fig. 1. Design science research approach for this project.

We consider the DSR approach to be particularly suited for gaining insights about the design of context-aware chatbots for product configuration as it involves iterative development and evaluation phases to ensure pertinence and validity [17]. Our DSR project follows the five established phases of problem awareness, suggestion, development, evaluation, and conclusion [18]. The following sections give an overview of the main steps of the implemented procedure, while Sect. 4 describes each step in detail. Our research project consisted of two cycles including an interim evaluation with first qualitative feedback and a more extensive final evaluation. The involved steps of the research project are illustrated in Fig. 1. As the interim evaluation demonstrated that the initial problem was sufficiently understood but indicated a slight adjustment of the design principles (DPs), the second iteration started with the suggestion phase after the interim evaluation.

Awareness of the Problem: To improve problem understanding, company employees with access to customer feedback (e.g. sales representatives, product managers) were interviewed to identify actual problems end-users are facing using product configuration interfaces. For this, seven semi-structured interviews were conducted. In total, seven employees (2 female, 5 male; average age = 34.14 years, SD = 6.87; working experience = 15.57 years, SD = 7.69; product configuration experience = 5.71 years, SD = 2.77) were interviewed. For the interviews, a semi-structured interview catalog was developed consisting of questions regarding demographics and professional background, as well as twelve pre-formulated questions concerning product configuration. The interview ranged from questions addressing the status quo of product configuration (e.g. "How often is the product configurator used by which user groups?") to questions addressing problems with product configuration (e.g. "Why do users cancel the configuration?") to customer requirements (e.g. "What are commonly expressed customer requirements?"). The interviews were transcribed using Microsoft Teams Live-Transcription and lasted on average 23,39 min (SD = 2,77). We hand-labelled sentences from the interviews and grouped them by topic. Agreement on the final clusters and classification of labelled information was reached in discussions. As we illustrate in the upcoming Sect. 4.2, the expert interview results were in line with related literature on product configuration focusing on requirements from a user perspective.

Suggestion: Based on the interviews and reviews, key end-user goals and requirements regarding product configuration interfaces were identified. Additionally, theoretical best practices and descriptive design knowledge from related fields and chatbot research were reflected. The insights were mapped to prescriptive DPs. The suggestion of Design Cycle 2 yielded refinements for the scope and realization of the DPs.

Development: The identified design principles were instantiated in a prototype. The second development phase included the implementation of measures to improve chatbot responses in terms of content and formatting. Additionally, product related information was added.

Evaluation: The interim evaluation was conducted in the form of unmoderated remote usability testing [19]. The final evaluation in Design Cycle 2 investigated whether the chatbot was able to facilitate the configuration process for end-users compared to a baseline chatbot, which only offered a step-by-step inflexible configuration process. Central evaluation criteria for the final evaluation were usefulness, ease of use, and cognitive load. These were measured by observing a user's configuration task performance and conducting a post-experiment survey.

4 Designing Context-Aware Chatbots for Product Configuration

4.1 Awareness of the Problem

The transcribed documents were analyzed to extract information addressing problems of existing product configurators, as well as the goals and requirements of end-users.

The analysis of interviews and literature resulted in five major groups of identified issues which are described in the subsequent paragraphs.

Limited Flexibility: The heterogeneity of consumers using product configuration interfaces is addressed by several authors [3, 20]. However, usually product configurators offer a single standard form for the customization experience. Users with extensive product knowledge go through the same process steps as novice customers and are exposed to the same level of detail and information [6]. Additionally, standard forms emphasize a strict order of the configuration which may differ substantially from actual user preferences: "I am relatively bound to the order, while that does not necessarily correspond to the things that are important to me" (Interviewee 7). Every selection of product characteristics can limit further selectable options. As a result, following the strict linear order of the configuration process can easily lead to suboptimal results.

Insufficient Information: Lack of information becomes apparent concerning several stages within a configuration process. First, many customers might not have detailed technical product domain knowledge at their disposal [3]. Therefore, they are not able to select specific characteristics to fulfill their needs [6]. This can be due to very domain-specific terminology. For this reason the major challenge of designing product configuration interfaces often also is conflict resolution [21]. Configuration conflicts appear when components selected by the customer do not fit together. Interviewee 7 reports frustrated customers, who "[…] could not click on what they wanted and also did not understand why".

Complexity/Confusion: High variant products are complex by nature and the central challenge of a product configuration interface is the reduction of complexity. For example at many car manufacturers often only a few product experts have a full overview of the product range [21]. Configuration interfaces presenting too many options at once do overwhelm customers [2, 3]. Interview participant 5 also reports that "[…] you simply don't see through how to do what […]".

Limited Intuitivity/Guidance: Limited intuitivity is reported by several interview participants, as the configuration is often "[…] not self-explanatory" (Interviewee 4). Even if user guidance in the form of explanatory texts exists, it is still perceived as "problematic" (Interviewee 2). Users also fail "[…] to click on the information to find out by myself" (Interviewee 7) due to limited intuitiveness of the information representation.

Duration: The process of configuring a customized product is a "cognitively challenging task" [6]. Often customers are "not [..] interested in fully exploiting the potential of customization" [6]. Such users have some preferences in mind and after those are fulfilled further options are not particularly relevant. However, often it still "[…] takes a lot of clicks to even get to a result" (Interviewee 5).

4.2 Suggestion

Deriving Design Requirements: Having gained a deeper understanding of problems associated with product configuration interfaces, in the next step, we derived requirements addressing those issues. First, the interface should help both novice and experienced users. Therefore, the chatbot must have a flexibility that ensures the right level of efficiency and assistance for experts as well as novice users. Customers should be able to decide on their own which level of detail they need, and in which order they configure their product. Furthermore, economic and psychological studies also show that human preferences change depending on the alternatives available [22]. Thus, a product configuration interface must always support easy and flexible changes during the customization procedure. The resulting design requirement (DR) was stated as: a product configurator must offer flexibility (DR1) to support customers differing vastly in their goals, knowledge, and configuration procedure.

Second, the interface must overcome the gap of missing information. It needs to support customers who use the configurator as a tool to research what is possible with the given product. Thus, the proposed design must allow one to learn more about the product, as well as its features and characteristics. Allowing for this case comes with a high level of transparency: What happens to my configuration if I choose a certain option? How does this affect my end product regarding my preferences? Are certain options combinable? In short, a product configurator must allow exploration (DR2).

Thirdly, a product configurator must address the issue of complexity and confusion. Therefore, a crucial challenge lies in the provision of an adequate amount of information at the right place at the right time and not all at once. Interviewees suggest reducing the required mental effort by employing intuitivity and facilitation: The product configurator should be clear, easy to navigate and as easy to understand as possible. Respectively, the proposed solution should be as self-explanatory and as easy to learn as possible. Thus, it must reduce cognitive effort (DR3).

Customers usually do not want to spend hours configuring the end product, they want to configure their desired products "as quickly […] as possible" (Interviewee 3). Therefore, a suitable configuration tool should implement mechanisms to configure quickly (DR4).

Translating Design Requirements into Design Principles: To define guidelines of how the DRs can be fulfilled in a chatbot interface, they were translated into DPs. The mapping is explained in the subsequent paragraphs and depicted in Fig. 2.

To increase flexibility (DR1) and opportunities for exploration (DR2), the chatbot design must provide conversational flexibility (DP1). The design must reflect that natural language input is of much higher variety than input in graphical user interfaces [23, 24]. The desired goal is to understand the needs of users and how they are best served [24]. It must offer users a way to directly formulate their preferences in natural language, as well as to start a configuration process by querying about the process itself for specific product properties. Furthermore, the chatbot needs to provide flexibility at any time during the configuration - it must allow for (contextual) queries, corrections, undoing of previous steps, and deviations from a standard configuration proceeding.

To increase opportunities for exploration (DR2) and to decrease required cognitive effort (DR3), the chatbot must provide relevant information before, during, and after the configuration process (DP2). Relevant information does include general (static) information about the product, the process, possibilities, and limitations. During the configuration, dynamic (context-dependent) information provision becomes particularly relevant. Furthermore, the chatbot must make use of NLU capabilities during the configuration to support users with vague or unclear request formulations. Context-dependent information is needed during the configuration: The chatbot should be able to name conflicting features and provides solutions about how a conflicting characteristic can be selected and what effects the selection would have. Ideally, the chatbot is also able to explain why different options cannot be combined. During the configuration, relevant information must also be available in the form of intermediate states and transparency regarding changes in the configuration.

To decrease required cognitive effort (DR3) and accelerate the configuration procedure (DR4) the chatbot must offer a clear structure (DP3). The design must reflect that contents and features of a text-based interface are to a much greater degree hidden from the user compared to a graphical interface [24]. Several authors implicate the necessity to reveal the system's capabilities throughout and during the interaction to form expectations and provide guidance [23–26]. Researchers have found that conversational guidance can be achieved by proposing users' responses [23], providing clickable buttons to generate text [27], and clarifying conversational flow using instructional messages [28]. For the domain of product configuration, the chatbot must present cues on how customers can reach their goals. Guidance can be enhanced by offering next steps during the configuration process. Finally, a guided mode could be offered where the chatbot asks questions, e.g. what characteristic of a certain feature the user wants to select. The implementation of DP3, however, must take DP1 into account. The clear structure must be an optional offer, that does not force users into a mechanical procedure.

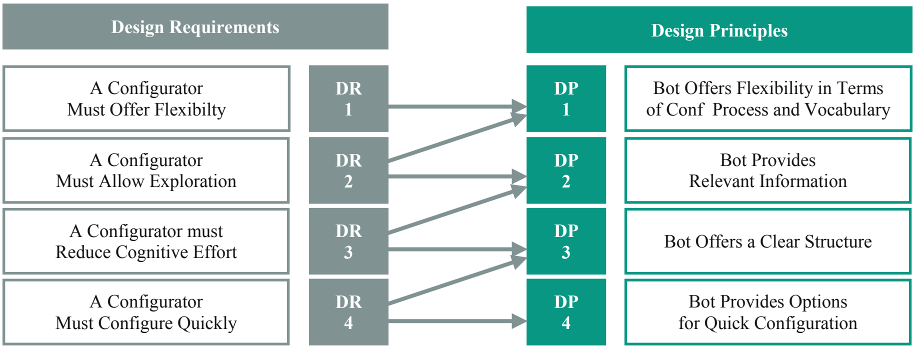

Fig. 2. Deriving design principles from design requirements

Acceleration mechanisms can be simple, for example by using default values (e.g. always choose the most often selected characteristic of available options). Configuration researchers also extensively investigate options to incorporate recommendations into product configuration [3, 29, 30]. Another suggestion from the interviews (Interviewee

5, 7) was to optimize the end-product according to a goal specified by the user (e.g. price or performance.).

4.3 Development

For chatbot development the conversational AI framework RASA was used. The primary criterion for selecting the framework was to satisfy technical requirements regarding the instantiation of the DPs. RASA is available open-source and performs well in comparison to other NLU services for chatbot development [31]. It contains NLU components for intent classification, entity extraction, and response retrieval, as well as a dialogue management component, deciding on the next action the chatbot should perform. For the implementation of the context-aware chatbot the component pipeline included the whitespace tokenizer, RASA's built-in RegexFeaturizer, LexicalSyntacticFeaturizer, its CountVectorsFeaturizer, as well as RASA's Dual Intent Entity Transformer (DIET) and RegexEntityExtractor. The Regex Entity Extractor was used to extract all the defined characteristics and features, the DIET Entity Extractor is able to identify entities that are not explicitly defined in the training data by using machine learning techniques. Furthermore, it offers a scalable architecture with easy integration of APIs and databases. Figure 3 shows the principal architecture of the chatbot:

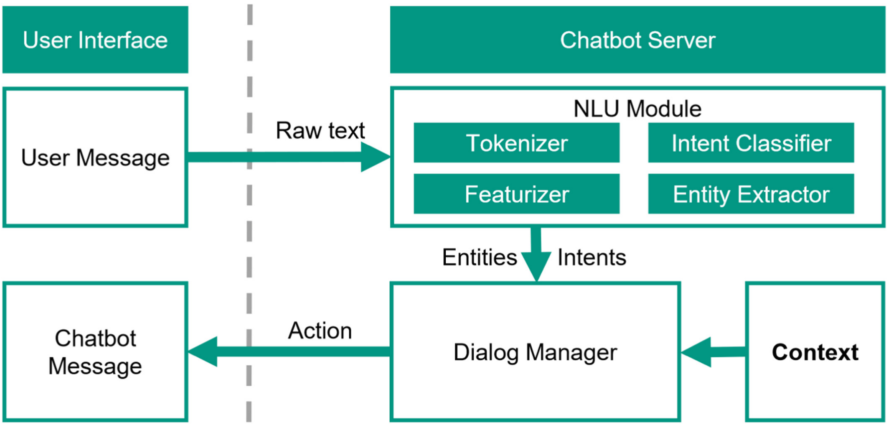

Fig. 3. Overview of the chatbot architecture

Two versions of the configuration chatbot were developed. Both versions were connected to the company's configuration backend. It determines feasible product variants, i.e. determines which components can be combined in which way. Thus, every customer request must be checked by calling the product domain database. Moreover, the backend can be consulted to retrieve additional information about the product, its properties, and their relationship to each other.

The baseline chatbot is based on simple question-answer mechanisms to configure step by step, as also realized in the standard web form. During the configuration process, the chatbot follows the same arranged order of features, and users are asked feature by feature for desired characteristics. Users can select characteristics by typing the string of the desired one or by the number. It can only provide context-independent information and does not implement the proposed design. When users with the baseline design query for specific features, the chatbot always answers with the same static response indicating which characteristics the feature contains. The proposed design, depicted on the right in Fig. 4, however, generates the response dependent on previous steps (e.g. highlighting selectable characteristics based on previous configuration steps). The context-aware chatbot was designed according to the suggested guidelines.

Context-awareness was realized using entity recognition to populate the chatbot's memory of the conversation. The conversational memory was used to provide dynamic, context-dependent responses, based on previous user utterances. The chatbot memory consists of slots storing information about all selected product characteristics and the procedural configuration stage. It was used to resolve and communicate configuration conflicts, provide information and assistance depending on the stage of configuration, as well as to assist with autocompletion resulting in a valid configuration. To react flexibly (DP1) to multiple directions of dialog the chatbot makes use of NLU modules to understand various user intents in all stages of the configuration process. Important user intents are for example to configure (i.e. to select characteristics), to request information, to request more guidance, or to ask for autocompletion.

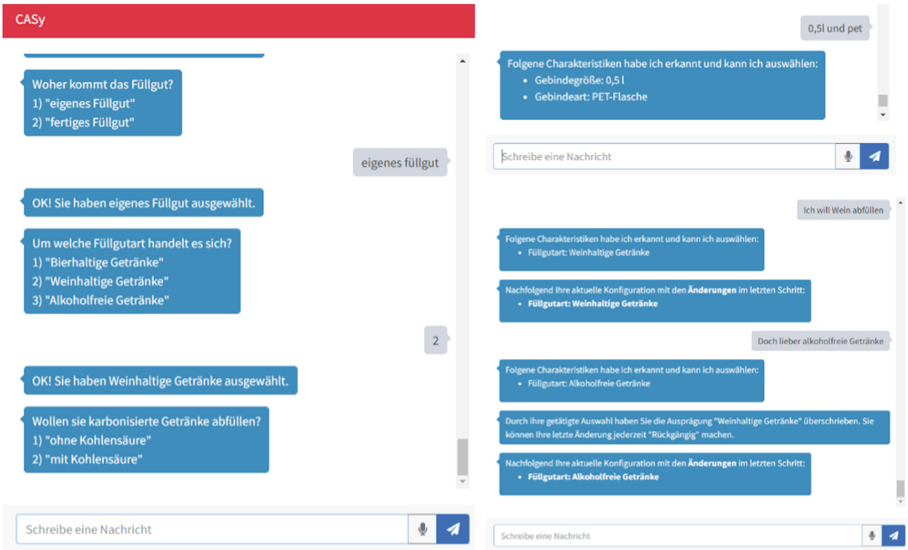

Fig. 4. Basic chatbot configuring step by step (left) and context-aware chatbot recognizing two characteristics at once (top right) or resolving conflicts (bottom right)

The results of the interim evaluation mainly addressed shortcomings of the instantiation of providing relevant information (DP2) and offering a clear structure (DP3). Thus, the second development phase added product and process related information to the chatbot responses (e.g. how many features have to be configured, what are the next configuration steps). To enhance structure and clarity the usage of emojis and markdown formatting to demarcate information was applied. Finally, to reveal the chatbot capabilities more transparently, the chatbot provides example utterances for all its functionalities. Figure 5 portrays exemplarily how the warning sign emoji is used to indicate conflicts, the light bulb icon to mark instructional information, the arrow icon to offer next steps, quotation marks to point out preformulated responses and bold formatting to highlight configuration changes.

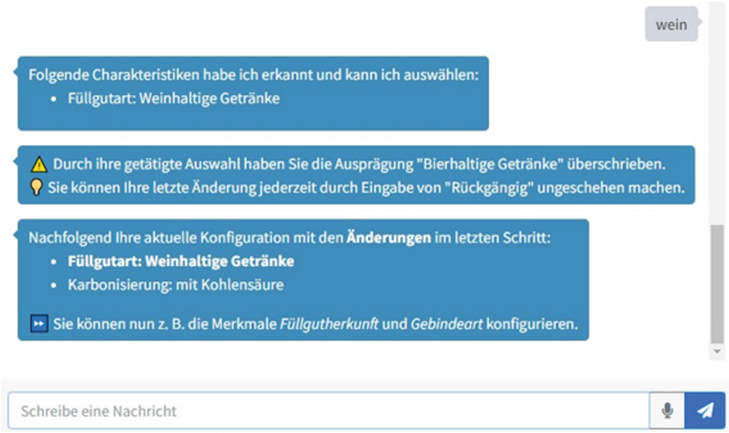

Fig. 5. Implementation of advices for the communication of configuration conflicts

5 Evaluation

5.1 Experimental Design

To evaluate the proposed design, the context-aware chatbot was compared to the baseline design in an online experiment. The evaluation investigated whether the formulated design principles could be instantiated in an artifact that did facilitate the configuration process for end-users. The artifact is assessed regarding its capabilities to address the outlined problems and to achieve added value to end-users. Based on established constructs of technology acceptance [32], the first two hypotheses were formulated as:

H1: The proposed design leads to a higher level of perceived ease of use.
H2: The proposed design leads to a higher level of (perceived) usefulness.

Based on the cognitive load theory, the third hypothesis was stated as:

H3: The proposed design leads to a reduction in perceived cognitive load.

To evaluate the proposed design, a controlled between-subject design online experiment was conducted. On the one hand, a control group of users performed a specific product configuration task with the baseline question-answer chatbot. The treatment group worked on the same task using the context-aware chatbot implementing the proposed design principles. Both groups answered a survey after the practical execution. The tasks included querying product related information and configuring according to desired product attributes. During the configuration task, participants had to resolve conflicts and to apply corrections.

Participants were mainly company employees from a different domain than product configuration who might have used a configurator lately (e.g. to configure a car). Those are considered potential end-users. Table 3 in the appendix overviews the demographics and controls of the participants.

5.2 Measurement Instruments

Perceived Ease of Use (PEOU) has been captured using a variety of subjective measures such as questionnaires [33–36]. Research has found PEOU to be related to the usability criteria perspicuity and dependability [36]. Perspicuity captures the degree to which it is easy to familiarize yourself with the product and to learn how to use it. A user feeling in control of the interaction experiences a high degree of dependability. Due to its ease of application and proven expressiveness, the User Experience Questionnaire (UEQ) is used to assess those dimensions [37, 38]. The questionnaire consists of 26 items in the form of semantic differentials (i.e. adjectives with the opposite meaning). It also captures the overall attractiveness, efficiency, and hedonic attributes (stimulation and novelty) which are associated directly or indirectly with the behavioral intention to use a system [36]. Perceived Usefulness (PU) evaluates an information system on a performance and output level [34]. To subjectively assess usefulness, scales were defined on the relevant task domains, which are information acquisition and configuration within the use case [34, 35]. As prescribed by the DRs, the main steps involved in a configuration process are information retrieval and customizing a product. Thus, the questionnaire contains two items where participants are asked to rate the prototype's usefulness regarding each process step. Applied user-related constructs are listed in Table 4.

The subjective assessment is complemented by measuring the user's task performance. The user performance was objectively assessed by logging the user's behavior during the interaction and by evaluating the participants' responses to the information retrieval task. In the appendix, Table 5 provides an overview of the obtained measurements. The *durations* and *number of messages* indicate the depth of the interaction and allow quantitative inferences regarding participants' proceedings and their investment in the task. Based on the responses, an *information retrieval score* was calculated, which is the share of correctly answered questions. From the logs, the performance criteria *interaction time* and the *number of messages* sent to the chatbot could be read out. As the configuration task specified several desired characteristics uniquely, a *configuration performance score* was calculated. The score was calculated as the share of correctly selected characteristics of all uniquely specified characteristics. Consequential errors (e.g. selecting a wrong characteristic which prevented the selection of further characteristics) were excluded from the score. Finally, the logs made traceable whether users adjusted incorrect characteristics and how much time they invested to do so.

This work hypothesizes that compared to a basic chatbot, a context-sensitive chatbot achieves higher task performances, higher PU, and higher PEOU due to a reduction in cognitive load and a higher fit between problem representation and task. Accordingly, measures to assess PU, PEOU, and cognitive load were defined and included in the experiment.

5.3 Evaluation Results

Users of the context-aware chatbot invested more time overall (on average 163.5 s more). Participants of both treatment groups invested about half of their time in the configuration task. To acquire the requested information, about the same amount of time was invested. To perform the configuration task users with the context-aware chatbot spent on average 111.5s more, which is significant (df = 10, t = 1.855, p = 0.047). To do a correction required significantly less time (on average about a minute) with the context-aware chatbot (df = 8, t = 2.845, p = 0.011). Two participants using the baseline chatbot did not perform the correction and were excluded from the configuration time measurement. The findings of more interaction investment with the context-aware design are supported by the observed number of sent messages. On average users of the baseline bot sent 37.5 messages (SD = 4.93) to the chatbot, while the test group sent 44.5 (SD = 13.28) messages on average.

Perceived Ease of Use: Participants reported a higher perspicuity, as well as higher dependability if they were using the context-aware chatbot (see Table 1). Despite the small group size (6 each) of our exploratory experiment, a significant difference between baseline and context-aware chatbot was already found for dependability.

Table 1. Descriptive statistics for perspicuity and dependability

Treatment	Baseline (n = 6)	Context-aware (n = 6)
Perspicuity		
Mean	1.00	1.67
SD	1.11	0.96
Dependability		
Mean	1.08	1.58
SD	0.52	0.41

Perceived Usefulness: The context-aware chatbot was on average perceived as more useful both for information retrieval (Mean context-aware = 2.17 (SD = 1.17) vs. Mean baseline = 1.17 (SD = 1.47) and configuration (Mean context-aware = 2.00 (SD = 0.63) vs. Mean baseline = 1.50 (SD = 1.64). In Table 2 the results for task performances are illustrated. For the information retrieval task, differences in performance scores are marginal, while the configuration scores differ significantly. The small variances in both tasks allow for the observation of behavior patterns that can be verified in the chat logs.

Table 2. Descriptive statistics for task performance

Treatment	Baseline (n = 6)	Context-aware (n = 6)
Inf.retrieval score		
Mean	0.67	0.71
SD	0.13	0.10
Conf. score		
Mean	0.79	0.93
SD	0.52	0.41

Cognitive Load: The measures on cognitive load suggest that participants experience on average rather low cognitive load, as both measured means are positive. Participants in the baseline design experienced less overall cognitive load (Mean context-aware = 0.82 (SD = 1.97) vs. Mean baseline = 1.33 (SD = 0.82)), while there is a high variance in the findings corresponding to the context-aware chatbot. The second scale, focusing rather on the medium to solve the task ("how easy or difficult was it to solve the tasks using the chatbot") favors the proposed design (Mean context-aware = 2.17 (SD = 10.75) vs. Mean baseline = 1.67 (SD = 0.82)).

Qualitative Results: Qualitatively, participants in the baseline design did miss flexibility: "It would also be nice if you didn't have to undo each step individually but jump to the desired point" (Participant 5). Participant 3 found it obstructive to "have to think about what is the next best thing". Five out of six participants using the baseline design found the step-by-step approach difficult to do corrections during the configuration.

Participants with the context-aware design criticized that not all their questions could be answered satisfactorily and they "had to rephrase the questions" (Participant 6). Here, the chatbot can be improved regarding its natural language understanding, as well as regarding the information it can provide (e.g. additional information about specific features).

Most participants highlighted the ease and intuitivity of selecting characteristics for both designs. For the baseline, the selection of characteristics was very fast due to the possibility of just typing in a number. However, the linear mechanic configuration comes at the cost of flexibility and control. All participants using the context-aware chatbot found it easy to configure. They acknowledged that typing in "buzzwords" (Participant 4) led to meaningful information or was sufficient to select characteristics. Participant 10 also appreciated the auto-complete option.

6 Discussion

6.1 Contribution

This work presents a novel approach addressing the known challenge of designing product configuration interfaces. Founded on prescriptive knowledge from product configuration and chatbot research, this work aims to facilitate the configuration process for end-users by providing an intuitive and easy to use interface. It synthesizes expert interviews and existing literature to address the main issues of existing solutions which are limited adaptiveness [6], information overload [2], and limited transparency [3, 6]. The formulated DRs are independent of the specific design of the interface and can guide the design of systems addressing the limitations of current solutions. Derived DPs reflect chatbot capabilities in terms of perceiving the conversation as the object of design. Developers can use the prescriptive knowledge to conceptualize chatbots as product configuration interfaces. The derived DPs of a chatbot for product configuration show the need for context-awareness for these kind of chatbots in order to provide flexibility (DP1), relevant information (DP2), structure (DP3), and quick configuration options (DP4). These principles can serve as the basis of a design theory for configuration bots.

From a practical perspective, the main deliverables are two chatbot prototypes (baseline and context-aware), as well as the evaluation insights. The results of the evaluation suggest that the proposed DPs did have an impact on participants' configuration experience and configuration outcome. Although the sample size was limited to 12 test persons, participants experienced significantly higher dependability and performed significantly better in the configuration with the context-aware chatbot. On average the evaluation showed that participants with the proposed design invested more time and sent more messages to the chatbot. By doing so, they used context-aware features to resolve configuration conflicts, select and reselect flexibly, and complete their configuration. They configured rather according to their preferences than to the order suggested by the bot. Although participants configuring with the baseline design knew better-suited alternatives exist and conflicts are part of the task, they made no effort to revise already selected characteristics. Participants stated qualitatively that the chatbot step-by-step approach made it hard to do corrections. Especially, the possibility to perform corrections flexibly yielded value to the test group. Participants in the test group were more successful and quicker in applying changes to their configuration. The control group confirmed this observation, as five out of six participants requested more flexibility for doing corrections.

For both designs, the self-reported overall mental effort was rather low, while it was even lower with the baseline design. The chat logs and evaluation results suggest that this low perceived mental workload is a result of the unsuitability of the baseline chatbot for the task at hand, as it did not lead the participants to reconsider their decisions. Accordingly, a reduction in perceived cognitive load (H3) could not be verified. However, the chatlogs suggest that the proposed design lead to higher engagement (e.g. more messages sent, more time invested to configure). The results indicate that the context-aware chatbot was perceived as easier to use than the baseline (H1), while the results were significant for the dimension dependability.

Significant differences in task performance results indicate that the proposed design was more useful than the baseline (H2), which was also inclined by queried perceived usefulness. A reduction in cognitive load due to the treatment could not be verified. Participants with the proposed design invested more mental effort, which was needed to complete the task at hand successfully. In conclusion, the context-aware chatbot is particularly suitable in an iterative configuration process involving changes and exploration.

The baseline can serve as a fallback when users want to quickly select characteristics manually.

6.2 Limitations and Future Works

This research project comes with five main limitations which offer opportunities for further studies on chatbots for product configuration. First, only the company's employees took part in the expert interviews. Future works could integrate a more diverse set of end-users. However, the interviewed experts did have access to customer feedback, some of them over years. Thus, they were able to produce generalizable insights from different product domains, while feedback from end-users might be subject to individual experience and the individual product domain.

Second, the evaluation was based on a sample size of 12 participants. Mainly participants without extensive product configuration experience were recruited, who were considered potential end-users. The evaluation results could be strengthened by increasing the sample size and targeting real end-users from the domain of application.

Third, the task participants had to perform during the evaluation was designed according to the DRs. To measure performance in terms of correctly selected characteristics objectively, the task's scenario specified preferences. Configuration conflicts were integrated into the task design, as their presence in configuration processes was confirmed by the interviewees and literature [21]. Further research must examine the proposed design's impact on the configuration experience of real end-users with their preferences. However, such an approach makes it difficult to apply objective performance measures, as the results are subject to individual preferences.

Fourth, the measures of perceived usefulness and cognitive load consisted of two items each. The small sample size and high variance in those measures do not permit the deduction of generally valid statements that have been quantitatively proven. However, the measurements allow to observe trends that could be reinforced by observing the participants' configuration proceedings, as well as their qualitative responses. Further research can extend evaluation metrics and sample sizes.

Fifth, by delimitation in the stated research questions, the proposed design principles are evaluated against a baseline without context-aware abilities. A comparison to a classical web form was not in the scope of this project. Due to their ability to reduce complexity for end-users and to provide flexibility in the configuration process, this work attributes added value to the usage of natural language, which has not been evaluated in a between-subjects comparison. On the one hand, participants qualitatively acknowledged the ease of selecting and changing product characteristics flexibly. On the other hand, web forms offer more flexibility regarding information representation than a chat interface. Further research can build on our contribution and compare different configuration modes.

7 Conclusion

Chatbots can reduce complexity and facilitate request formulation for end-users by allowing them to interact in natural language. Product configuration is an application where customers can be overwhelmed by the mass and complexity of the product features in current solutions. Existing interfaces often do not differentiate between novice and expert users and emphasize an order of configuration for end-users.

As we could show in a small-scale evaluation, a context-aware chatbot for product configuration provides flexibility in terms of possible conversation paths and vocabulary used. Drawing from expert interviews and literature, DRs were compiled from which general DPs were formulated. Those can serve as a blueprint to guide the development of chatbots or natural interfaces for product configuration. The instantiation of the proposed design was evaluated against a baseline design in a between-subjects comparison. Users with the proposed design invested on average more time and messages for the configuration and ended up with a better output in terms of task performance. Furthermore, participants attributed higher usefulness and higher usability to the proposed design. For both designs, the reported mental workload was rather low, indicating low perceived complexity. The instantiation of the DPs proved to be especially beneficial for quickly selecting and revising product attributes in an iterative process, flexibly in terms of order and vocabulary used. In use-cases with a higher number of variable product attributes, the visual possibilities offered by a chat window appear to be limited and might be supported by a web-form representation.

Further researchers are invited to apply, evaluate, and extend the proposed design and design theory on a chatbot for product configuration facilitating the user configuration process. Since the results of this project suggests high potential for combined approaches, a combination of a chatbot and a classical configuration interface could be an idea for future work to build on.

Appendix

Questions Asked in the Semi-structured Interviews

Status Quo

- "How often is the product configurator used by which user groups?"
- "What are the goals of a customer when configuring the product?"
- "What proportion of the configurations that have been started will be completed?"

Problem identification

- "For what reasons do users cancel a product configuration?"
- "How is the feedback on the product configurator?"
- "What complaints or negative feedback about the configurator are there?"
- "What are the hurdles in the current configuration process?"

Requirements
Abstract requirements

- "What are frequently expressed customer requirements?"
- "What are the relevant properties for you that a configurator has to implement?"

- “How can intuitive operation or a pleasant process flow be achieved?”

Concrete requirements for configuration chatbot

- “In your opinion, how would a text-based chatbot have to proceed in order to to enable a pleasant configuration process?”
- “What abilities of the chatbot would be desirable”
- Opt.: negative questions in cases of insufficient feedback

(translated from German).

Final Evaluation

Table 3. Final experiment groups

Condition	N	Age	Gender	Product configuration experience*	Chatbot usage**
Control (baseline design)	6	Mean = 29.5 (SD = 4.14)	Female = 4 Male = 2	Non-Expert = 4 Expert = 2	Mean = 3.00 (SD = 1.41)
Context-Aware (proposed design)	6	Mean = 30.17 (SD = 10.34)	Female = 2 Male = 4	Non-Expert = 4 Expert = 2	Mean = 3.33 (SD = 1.21)

*Measured on a five-point Likert scale
**Measured on a seven-point Likert scale

Table 4. User-related constructs

Construct	Reference	Measurement	Items
User experience questionnaire	Laugwitz et al. (2008)	7-point likert scale	26
Perceived usefulness	Lund (2001)	7-point likert scale	2
Cognitive load	Paas (1992), Eysink et al. (2009)	7-point likert scale	2

Table 5. Performance related metrics

Measurement	Description
Interaction duration	Duration to complete all tasks
Information retrieval duration	Duration to complete task one
Configuration duration	Duration to complete task two
Correction retrieval duration	Duration to complete task three
Number of messages 0000000	Total number of messages sent to the chatbot
Information retrieval score	Performance score in the information retrieval task
Configuration score	Performance score in the configuration task

References

1. Pine, B.J., II., Victor, B., Boynton, A.C.: Making mass customization work. Harv. Bus. Rev. **71**, 108–111 (1993)
2. Huffman, C., Kahn, B.E.: Variety for sale: mass customization or mass confusion? J. Retail. **74**, 491–513 (1998). https://doi.org/10.1016/S0022-4359(99)80105-5
3. Felfernig, A. (ed.): Knowledge-Based Configuration from Research to Business Cases. Morgan Kaufmann is an Imprint of Elsevier, Amsterdam (2014)
4. Li, F., Jagadish, H.V.: Constructing an interactive natural language interface for relational databases. Proc. VLDB Endow. **8**, 73–84 (2014). https://doi.org/10.14778/2735461.2735468
5. Sun, Y., Zhang, Y.: Conversational recommender system. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. ACM, New York (2018)
6. Randall, T., Terwiesch, C., Ulrich, K.T.: Principles for user design of customized products. Calif. Manag. Rev. **47**, 68–85 (2005). https://doi.org/10.2307/41166317
7. Sweller, J.: Cognitive load during problem solving: effects on learning. Cogn. Sci. **12**, 257–285 (1988). https://doi.org/10.1016/0364-0213(88)90023-7
8. Blecker, T., Abdelkafi, N., Kreutler, G., et al.: Product configuration systems: state of the art, conceptualization and extensions (2004)
9. Dale, R.: The return of the chatbots. Nat. Lang. Eng. **22**, 811–817 (2016). https://doi.org/10.1017/s1351324916000243
10. Gupta, S., Borkar, D., de Mello, C., et al.: An e-commerce website based chatbot. Int. J. Comput. Sci. Inf. Technol. **6**, 1483–1485 (2015)
11. Cui, L., Huang, S., Wei, F., et al.: SuperAgent: a customer service chatbot for e-commerce websites. In: Proceedings of ACL 2017, System Demonstrations, Stroudsburg, PA, USA. Association for Computational Linguistics (2017)
12. Nica, I., Tazl, O.A., Wotawa, F.: Chatbot-based tourist recommendations using model-based reasoning. In: ConfWS, pp. 25–30 (2018)
13. Christakopoulou, K., Radlinski, F., Hofmann, K.: Towards conversational recommender systems. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, New York (2016)
14. Clark, L., Pantida, N., Cooney, O., et al.: What makes a good conversation? Challenges in designing truly conversational agents. In: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (2019)

15. Jain, M., Kumar, P., Kota, R., et al.: Evaluating and informing the design of chatbots. In: Proceedings of the 2018 Designing Interactive Systems Conference. ACM, New York (2018)
16. Vlasov, V., Drissner-Schmid, A., Nichol, A.: Few-shot generalization across dialogue tasks (2018)
17. Hevner, A.R.: A three cycle view of design science research. Scand. J. Inf. Syst. **19**, 4 (2007)
18. Kuechler, B., Vaishnavi, V.: On theory development in design science research: anatomy of a research project. Eur. J. Inf. Syst. **17**, 489–504 (2008). https://doi.org/10.1057/ejis.2008.40
19. Barnum, C.M.: Usability Testing Essentials: Ready, Set… Test!, 2nd edn. Morgan Kaufmann, Amsterdam (2020)
20. Ardissono, L., Felfernig, A., Friedrich, G., et al.: A framework for the development of personalized, distributed web-based configuration systems. AI Mag. **24**, 93 (2003). https://doi.org/10.1609/aimag.v24i3.1721
21. Drews, M.: Interaction Patterns für Produktkonfiguratoren. In: Mensch & Computer 2008, pp. 367–376. Oldenbourg Wissenschaftsverlag GmbH (2008)
22. Kahneman, D., Tversky, A.: Prospect theory: an analysis of decision under risk. In: Handbook of the Fundamentals of Financial Decision Making, pp. 99–127. World Scientific (2013)
23. Diederich, S., Brendel, A.B., Kolbe, L.M.: Designing anthropomorphic enterprise conversational agents. Bus. Inf. Syst. Eng. **62**(3), 193–209 (2020). https://doi.org/10.1007/s12599-020-00639-y
24. Følstad, A., Brandtzæg, P.B.: Chatbots and the new world of HCI. Interactions **24**, 38–42 (2017). https://doi.org/10.1145/3085558
25. Dzindolet, M.T., Peterson, S.A., Pomranky, R.A., et al.: The role of trust in automation reliance. Int. J. Hum. Comput. Stud. **58**, 697–718 (2003). https://doi.org/10.1016/S1071-5819(03)00038-7
26. Luger, E., Sellen, A.: "Like having a really bad PA" the gulf between user expectation and experience of conversational agents. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing System, pp. 5286–5297 (2016)
27. Jain, M., Kota, R., Kumar, P., et al.: Convey: exploring the use of a context view for chatbots. In: Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, pp. 1–6 (2018)
28. Gnewuch, U., Morana, S., Maedche, A.: Towards designing cooperative and social conversational agents for customer service (2017)
29. Cöster, R., Gustavsson, A., Olsson, T., et al.: Enhancing web-based configuration with recommendations and cluster-based help. In: 2002 Workshop on Recommendation and Personalization in eCommerce (2002)
30. Tiihonen, J., Felfernig, A.: An introduction to personalization and mass customization. J. Intell. Inf. Syst. **49**(1), 1–7 (2017). https://doi.org/10.1007/s10844-017-0465-4
31. Braun, D., Hernandez-Mendez, A., Matthes, F., et al.: Evaluating natural language understanding services for conversational question answering systems. In: Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, Stroudsburg, PA, USA. Association for Computational Linguistics (2017)
32. Davis, F.D., Bagozzi, R.P., Warshaw, P.R.: User acceptance of computer technology: a comparison of two theoretical models. Manag. Sci. **35**, 982–1003 (1989). https://doi.org/10.1287/mnsc.35.8.982
33. Aladwani, A.M.: The development of two tools for measuring the easiness and usefulness of transactional Web sites. Eur. J. Inf. Syst. **11**, 223–234 (2002). https://doi.org/10.1057/palgrave.ejis.3000432
34. Freeman, L.A., Jessup, L.M.: The power and benefits of concept mapping: measuring use, usefulness, ease of use, and satisfaction. Int. J. Sci. Educ. **26**, 151–169 (2004). https://doi.org/10.1080/0950069032000097361

35. Lund, A.: Measuring usability with the USE questionnaire. Usability Interface **8**(2), 3–6 (2001)
36. Mlekus, L., Bentler, D., Paruzel, A., et al.: How to raise technology acceptance: user experience characteristics as technology-inherent determinants. Gr. Interakt. Org. **51,** 273–283 (2020). https://doi.org/10.1007/s11612-020-00529-7
37. Laugwitz, B., Held, T., Schrepp, M.: Construction and evaluation of a user experience questionnaire. In: Holzinger, A. (ed.) USAB 2008. LNCS, vol. 5298, pp. 63–76. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89350-9_6
38. Schrepp, M., Hinderks, A., Thomaschewski, J.: Construction of a Benchmark for the User Experience Questionnaire (UEQ), vol. 4, p. 40 (2017) https://doi.org/10.9781/ijimai.2017.445