Information Systems and e-Business Management (2023) 21:37–79
https://doi.org/10.1007/s10257-022-00570-7

ORIGINAL ARTICLE

How can I help you? Design principles for task-oriented speech dialog systems in customer service

Thuy Duong Oesterreich[1] · Eduard Anton[1] · Julian Schuir[1] · Alexander Brehm[1] · Frank Teuteberg[1]

Received: 5 July 2021 / Revised: 16 June 2022 / Accepted: 17 August 2022 / Published online: 6 December 2022

Abstract

Organizations are increasingly delegating customer inquiries to speech dialog systems (SDSs) to save personnel resources. However, customers often report frustration when interacting with SDSs due to poorly designed solutions. Despite these issues, design knowledge for SDSs in customer service remains elusive. To address this research gap, we employ the design science approach and devise a design theory for SDSs in customer service. The design theory, including 14 requirements and five design principles, draws on the principles of dialog theory and undergoes validation in three iterations using five hypotheses. A summative evaluation comprising a two-phase experiment with 205 participants yields positive results regarding the user experience of the artifact. This study contributes to design knowledge for SDSs in customer service and supports practitioners striving to implement similar systems in their organizations.

Keywords Speech dialog system · Conversational agent · Design science research · Design principles · Customer service · Experiment

✉ Thuy Duong Oesterreich
toesterreich@uni-osnabrueck.de

Eduard Anton
eduard.anton@uni-osnabrueck.de

Julian Schuir
julian.schuir@uni-osnabrueck.de

Alexander Brehm
albrehm@uni-osnabrueck.de

Frank Teuteberg
frank.teuteberg@uni-osnabrueck.de

[1] Department of Accounting and Information Systems, Osnabrück University, Katharinenstr. 3, 49074 Osnabrück, Germany

1 Introduction

In recent years, technological advances in artificial intelligence (AI) and natural language processing (NLP) have accelerated the proliferation of speech-based dialog systems (SDSs) in customer service (Rzepka et al. 2020). Organizations leverage SDSs as a cost-efficient alternative to human operators by performing routine support tasks such as answering frequently asked questions, authenticating customers, or transmitting process-relevant information (Jusoh 2018; Doherty and Curran 2019). Businesses can benefit from the use of SDSs by saving personnel costs while satisfying customers with 24/7 availability and reducing hold times in phone queues (Jusoh 2018; Kaczorowska-Spychalska 2019). Given these capabilities, SDSs have the potential to create competitive advantages such as increasing customer loyalty, net promoter scores (i.e., the likelihood to recommend a company), and sales conversion rates (Deloitte 2019).

However, customers often report frustration when interacting with SDSs due to poorly designed solutions (Walsh et al. 2018; zendesk 2019). Aside from technical issues such as limited natural language understanding capabilities, SDSs in business practice frequently exhibit deficiencies in their capacity to engage in convincing, human-like, and goal-oriented conversations (Forrester 2017). These deficiencies are related to the design of the dialog strategy employed by an SDS. For instance, many SDSs employ closed dialog strategies that allow a navigation along predefined paths (Dale 2016). This feature can be perceived as unsatisfactory by users, as listening to long instructions and predefined menu options can be tedious (Walsh et al. 2018). Similarly, SDSs, which are less path-oriented and more open to direct customer queries, fail to meet customer expectations raised by their human-like imitation, as they are incapable of responding to users' individual intents in a convincing manner, e.g., due to the variety of possible inputs (Forrester 2017; Kirkpatrick 2017). Taken together, these issues demonstrate that reaping the touted benefits of SDSs requires careful dialog design to create a satisfying user experience.

Following the technological advances in the realms of AI and NLP, related studies in the fields of human–computer interaction and information systems (IS) have devoted growing attention to the design of dialog systems in recent years. However, the vast majority of these studies address the design of text-based dialog systems (Diederich et al. 2022). For instance, Gnewuch et al. (2017) introduce four design principles for social and cooperative chatbots in customer service. Aside from suggesting the integration of social cues to provide a human-like dialog, the authors propose the implementation of informative opening messages and conversational breakdown recovery strategies to ensure a goal-oriented interaction. Design cues for speech-based interaction address socio-phonetic design (Schmitt et al. 2021) or anthropomorphic features (Pfeuffer et al. 2019) without focusing on dialog strategies. Moreover, studies addressing SDSs frequently adopt a behavioral science perspective to synthesize evidence-based recommendations for the design of dialog systems. Thereof, several studies have examined the empirical comparison of user experience of different dialog strategies (e.g., Jurafsky 2000; Chu et al. 2005; Merdivan et al. 2019). Accordingly, the existing body of knowledge is lacking in design principles targeted at operationalizing dialog strategies for SDS in customer service.

On a conceptual level, dialog strategies can be operationalized through a frame-based or a finite-state dialog strategy. SDSs that follow the finite-state approach are characterized by a system-guided dialog based on predefined menu options (also referred to as closed dialog strategies). On the contrary, frame-based SDSs offer users the possibility to freely express their concerns based on open questions in a human-like conversation (also denoted as open dialog strategies) (Griol et al. 2017). However, the particular dialog strategy that is appropriate for providing a satisfying user experience in customer service settings is not evident (Meng et al. 2003; Savcheva and Foster 2018). To provide guidance to scholars and practitioners on this design uncertainty and address the lack of design knowledge related to dialog strategies for SDSs, our overarching objective in this study is to systematically develop, theoretically ground, and justify design knowledge in terms of a design theory. The design theory draws on Bunt's (2000) dialog theory and comprises both requirements and design principles (DPs) for SDS dialog strategies in customer service. We empirically evaluate the instantiations of the DPs in terms of user experience through a two-phase experiment with 205 participants concerning the proposed design theory.

The remainder of this study is organized into several sections. In Sect. 2, we outline the theoretical background and related work in the area of SDSs. In Sect. 3, we explain our multi-method design science-oriented research approach. In Sect. 4, the requirements and DPs are elaborated and modified in three iterations. Additionally, we describe the development of the SDS prototypes based on the elaborated requirements and DPs. In Sect. 5, we describe the evaluation of the artifact. We thereafter discuss the findings of our research in Sect. 6 by highlighting the main implications for research and practice and outlining the limitations of our study. Concluding remarks are provided in Sect. 7.

2 Theoretical background

2.1 Speech Dialog Systems in customer service

To date, a wealth of terms is frequently used for different kinds of dialog systems, including digital assistants, chatbots, conversational agents, and machine conversation systems (McTear et al. 2016, p. 39; Luger and Sellen 2016; Diederich et al. 2019a). Dialog systems can process concerns and inquiries from customers based on text- or speech-based inputs. Our focus is on SDSs used in phone-based customer service. Speech as an interaction modality in customer service remains very popular with customers (zendesk 2019). An SDS thereby serves as the machine-based interface to customer service, which resembles human support personnel (Cho et al. 2019); furthermore, as several studies indicate, an SDS can be convenient for customers because issues can often be explained more quickly orally than in writing (Ruan et al. 2018; Pfeuffer et al. 2019; Schmitt et al. 2021). This is especially true for the elderly who are not as familiar with typing (Pfeuffer et al. 2019; Gupta 2021).

SDSs can be differentiated in task-oriented and non-task-oriented systems (Hussain et al. 2019; Mairittha et al. 2019). Task-oriented systems are designed to assist users in performing basic tasks in short dialogs, such as booking a flight or purchasing a product, whereas non-task-oriented systems are configured to simulate a natural conversation that

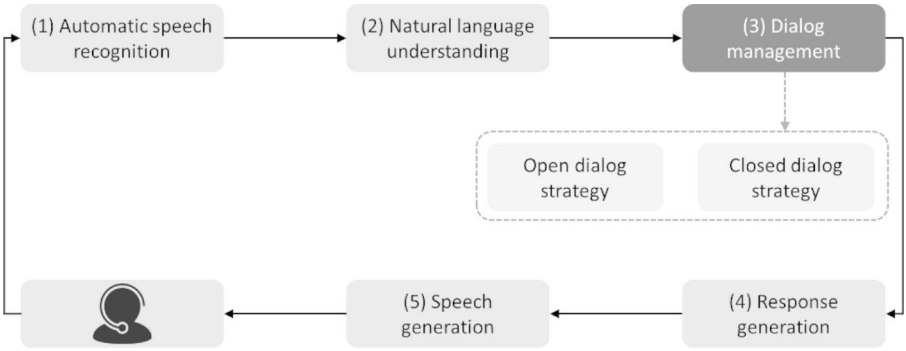

Fig. 1 Schematic structure of an SDS, adapted from Merdivan et al. (2019)

resembles human-to-human interactions (Hussain et al. 2019). The focus of our study is on task-oriented SDS, as we explore customer service, which is generally about solving a specific request or concern.

As schematically illustrated in Fig. 1, an SDS consists of five central modules: (1) automatic speech recognition, (2) natural language understanding, (3) dialog management, (4) response generation, and (5) speech generation (Merdivan et al. 2019).

When contacting customer service, usually via a telephone, the system provides an introduction to which the customer responds with spoken words. These prompts are picked up by the microphone of the user device and transmitted to an automatic speech recognition module, which in turn converts the speech into text for further processing. The outputted text serves as input for an NLP engine, which interprets the text by extracting semantic information such as dialog acts, entities, and intents (Firdaus et al. 2021). In linguistic terms, a dialog act is a functional tag of an utterance (e.g., question, statement, conversation opener), entities are proper names (places, times, customer numbers, personal names), and intents relate to the user goal (Chen et al. 2018; Firdaus et al. 2021). To understand the semantics of utterances, the utterances, entities, or intents are classified according to predefined classification schemes (Firdaus et al. 2021). The central module of an SDS is the dialog manager that fulfills several functions, namely providing and updating the dialog context, coordinating external modules, and deciding what information is needed and when this information should be extracted (Traum and Larsson 2003). Thus, dialog management can be understood as the component of an SDS that is responsible for controlling dialog flows and making context-based decisions (McTear et al. 2016, p. 210; Zhao et al. 2019). In addition, dialog management defines how incorrect, unforeseen, or unclear information is handled. Coordinating the dialog flow has a major impact on user satisfaction and consequently requires an ample amount of attention during development (McTear et al. 2016, p. 210).

Depending on the utterance of the customer and the selected dialog strategy, the system response is generated by the response generation module (Klüwer 2011). The most widespread method is the use of response templates with so-called slots or placeholders filled with the entities from dialog management (Singh and Arora 2020). In the last step, the generated response is reproduced in natural language by the speech generation module that synthetically generates speech (Burgoon et al. 2017, p. 257). Depending on

the dialog flow, multiple conversation turns may be necessary to fulfill the customer's inquiry (Merdivan et al. 2019).

2.2 Dialog strategies for Speech Dialog Systems

The underlying strategy of a dialog system steers the flow of the conversation. This conversational flow can be directed by following the finite-state approach, such that the user navigates through the system by using predefined menu options (McTear 2017). With this approach, the system initiative is generally high, offering possible paths of the conversational direction, whereas user initiative is limited to selecting the preferred path by command (Chu et al. 2005). Thus, the underlying closed dialog strategy provides system-guided support, which aims to collect relevant data successively through a fixed sequence of questions (McTear 2002). Accordingly, the main goal of the closed dialog strategy is successful task realization through system-controlled guidance, thereby providing structure for all menu and error correction options and narrowing down the possible utterances (Lee et al. 2017). By contrast, the frame-based approach (open dialog strategy) merely determines the boundaries of the conversation and offers users the possibility to freely express their concerns (Torres et al. 2019). Instead of detailed menu prompts, open questions convey a natural conversation, thus imitating human dialogs (Griol et al. 2017). This feature allows users to directly name multiple entities to be captured and, if necessary, supplemented by specific system questions to obtain the required information (slot filling) (Singh and Arora 2020). Consequently, in contrast to the finite-state system, this approach is characterized by low system initiative and low level of system support. However, this conversational flexibility also has shortcomings. Due to the wider range of expressions to be considered in model training, such systems are more error-prone (Lee et al. 2017).

Although closed dialog strategies predominate in business practice (Dale 2016), researchers have been interested in the differences and comparisons between dialog strategies since the 1990s. Delogu et al. (1998) examine the first forms of interactive voice response systems based on natural speech input and compare them with closed dialog technologies such as dual-tone-multi-frequency, which allows the user to interact with the system via a closed menu using phone keys. Similarly, a number of more recent studies compare open with closed SDSs, which yield rather contradictory results. For example, Meng et al. (2003) demonstrate that open dialog systems are superior to closed dialog systems in terms of performance accuracy and error rates when used in a simple foreign exchange domain. On the contrary, the study by Savcheva and Foster (2018) shows that open dialog systems do not provide higher customer satisfaction, but the more human-like interaction has led to a somewhat more efficient interaction in terms of errors encountered. As these studies indicate (Meng et al. 2003; Savcheva and Foster 2018), we do not think that a comparison to determine whether an open or a closed dialog strategy is generally preferable to the other is beneficial, as both strategies have their strengths and weaknesses, depending on the use scenario. Rather, dialog strategies must be adapted to the specific use case to ensure a satisfactory user experience (Kvale et al. 2021). Therefore, our study focuses on devising a design theory that integrates design principles that apply to both strategies and aim for a satisfactory user experience in customer service. We instantiate an SDS with an open and a closed dialog strategy to

evaluate the utility and effectiveness of the devised design theory for both dialog strategies and to highlight the strengths and weaknesses in the user experience in the context of a task-oriented use case.

3 Design Science Research Approach

Our primary purpose is to devise and evaluate a design theory that integrates the design principles of an open and closed dialog strategy for an SDS in customer service. Therefore, we rely on the design science research (DSR) paradigm for guiding the entire research process based on the wealth of guidelines and principles proposed by various IS scholars (Walls et al. 1992; Hevner et al. 2004; Gregor and Jones 2007; Peffers et al. 2007; Gregor and Hevner 2013). Similar to several other DSR studies from the past (Markus et al. 2002; Abbasi and Chen 2008; Meth et al. 2015; Diederich et al. 2020), we follow the approach of Gregor and Jones (2007) and Walls et al. (1992) and propose a design theory for SDSs for a class of artifacts, with the aim of addressing a class of problems rather than a single artifact. In doing so, we consequently strive for a solid theoretical foundation and the empirical validation of our design theory throughout the research process.

With this purpose in mind, we draw on the DSR methodological approach of Peffers et al. (2007), which provides a structured development process with several continuous design and evaluation cycles. Although Peffers et al. (2007) adopt a different view toward the role of artifacts and design theory, their multi-step approach supports the basic principles of the design theory development process as proposed by Gregor and Jones (2007) and Walls et al. (1992); therefore, this multi-step approach is considered appropriate for guiding the research process of the current study. The development and evaluation of the SDS design theory takes place in three iteration rounds, as illustrated in Fig. 2.

The development process is initiated by a brief problem identification and motivation (Activity 1) to justify the value of a solution for the problem. The objectives for a solution are subsequently defined (Activity 2). Through a systematic literature review according to vom Brocke et al. (2009) and Webster and Watson (2002), we acquire knowledge about the research problem as well as justificatory knowledge that can be used for informing the design of our artifacts in the sense of kernel theory (Walls et al. 1992; Gregor and Jones 2007; Gregor and Hevner 2013). Accordingly, the systematic literature review benefits not only the initial steps of problem identification and motivation (Sect. 1, Introduction) and the definition of objectives for a solution (Sect. 2, Theoretical Background) but also the subsequent design and evaluation (Activity 3) (Sect. 4 Design Theory, and Sect. 5, Evaluation).

Due to the interdisciplinary nature of the research topic located at the interfaces between IS, computer science, human–computer interaction, and other related fields (Gnewuch et al. 2017), we conduct a broad literature search in several interdisciplinary

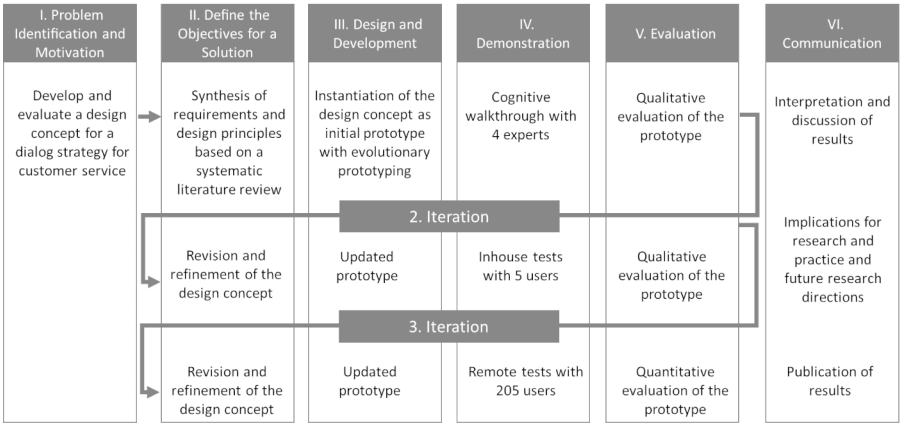

Fig. 2 Design science research approach according to Peffers et al. (2007)

databases. The literature search yields a sample of 74 articles that can be considered relevant for the objectives of our study.[1]

Based on this body of academic and business knowledge, we start the design of the artifact (Activity 3) by identifying a set of 14 requirements to help us to address the class of goals to be achieved. Guided by these requirements, we explicate five DPs following the principles of Gregor et al. (2020) to meet the requirements for designing a dialog strategy for SDSs in customer service. Central to each design theory is a set of hypotheses for testing the question of whether the proposed DPs meet the requirements (Walls et al. 1992; Gregor and Jones 2007). Thus, we initially develope a system architecture based on the five DPs, which serves as a foundation for the subsequent development of the prototypes using the evolutionary prototyping approach. Consistent with the principles of the DSR approach, the evolutionary prototyping method is characterized by a process of constant revision, refinement, and testing of an artifact (Davis 1992). This method enables us to develop, test, and redesign the SDS in several iterations until we meet the requirements. The prototypes are iteratively tested by potential users and modified based on the users' feedback (Carter et al. 2001). Only when the SDS is considered to meet the requirements, the evolutionary prototyping of the respective DSR iterations is completed and the next activity can begin (Activity 4). Given the significance of a diligent evaluation process, it is considered essential to every DSR project (Hevner et al. 2004; Peffers et al. 2007). Similarly, Walls et al. (1992) and Gregor and Jones (2007) posit that design theories must be subject to a thorough empirical investigation to test the hypotheses concerning the proposed design theory.

We follow the framework of Venable et al. (2016) to develop an appropriate evaluation strategy comprising three evaluation rounds in a naturalistic setting (Activity 5). Naturalistic evaluation allows us to study the performance of our artifacts in a real business environment and to increase the internal validity and rigor of the assessment process (Pries-Heje et al. 2008; Venable et al. 2016). After the first prototyping phase, our

[1] More details on the literature search and selection process as well as a complete overview of the selected literature sample are provided in Appendices A.1–A.3.

initial SDS prototypes are evaluated by four SDS experts with several years of professional experience in the field of dialog systems and user-experience design. The experts analyze the systems in terms of usability and feasibility through the cognitive walkthrough method, an effective approach for evaluating the design of user interfaces in early prototyping phases based on cognitive theory (Rieman et al. 1995). In the second iteration round, we subject the revised prototypes to further testing by five potential users to ensure that the tasks in the dialog system could be mastered without prior experience or further assistance. We use the feedback of the experts and users from both iterations to revise and refine our prototypes. Finally, in the third iteration, we conduct a two-phase experiment with 205 participants to empirically validate the developed prototypes. For this purpose, we invite the participants to take part in a remote test in which two tasks have to be completed using the prototypes. The users subsequently fill out the user-experience questionnaire based on their experiences with the closed and open SDS. The questionnaire for the user-experience survey is informed by the Subjective Assessment of Speech System Interfaces framework established for evaluating SDSs (Hone and Graham 2000). With the completion of the user survey, the design and evaluation cycle is completed. Finally, as the final communication step of the DSR process, we interpret and present the results and key findings (Activity 6).

4 Design theory of the Speech Dialog Systems

In this section, we focus on the development of the design theory according to the model proposed by Walls et al. (1992) and Abbasi and Chen (2008), which encompasses four main design components of a design theory (cf. Table 1). To ensure a consistent design theory, we draw on the body of knowledge in the research field of SDSs for a theoretical foundation of the development process. We thereby use dialog theory (Bunt 2000) as justificatory knowledge (kernel theories) to identify the requirements as a major precondition for deriving the corresponding DPs that can be adapted to the dialog management of the SDSs. However, several other theories are equally suited for guiding the socio-technical design of speech dialog systems, such as task–technology fit theory (Goodhue 1995), according to which a match between task characteristics and technology characteristics leads to improved user performance, or social response theory (Nass and Moon 2000; Moon 2000) and the embodied social presence theory (Mennecke et al. 2011), which consider technologies such as SDSs as social actors that should be designed as human-like as possible. In our research, we rely on dialog theory as kernel theory because it is ideally suited for guiding the design of dialog systems that are to assist users with simple tasks and short dialogs, thus aligning with our focus as described in Sect. 2. Furthermore, dialog theory provides guidelines on the socio-technical design of the dialog systems, for example on the communicative behavior of the agents (Bunt 2000).

The requirements represent a set of main goals and requisites, which specifies the functions of an SDS. The DPs, in turn, form a set of corresponding principles devised from the requirements. Guided by the main features of frame-based (open) and finite-state (closed) dialog strategies as presented in Sect. 2.2, we identify five main categories of requirements and DPs: prompt design, menu design, persona design, confirmation

Table 1 Main components of the design theory of an SDS dialog strategy

Component		Description
1.	Kernel theory	Dialog theory (Bunt 2000)
2.	Requirements	Main goals and requirements specifying the functions of an SDS dialog strategy, devised from justificatory knowledge (kernel theory)
3.	Design principles	Corresponding principles that are hypothesized to meet the requirements concerning the functions of prompt design, menu design, persona design, confirmation strategy, error management, and functional design
4.	Testable hypotheses	Qualitative and quantitative evaluations of the prototypes to empirically validate the value claims of the open SDS versus the closed SDS. The hypotheses (H1–H5) are constructed based on the requirements and DPs and provided in Sect. 5.

strategy, error management, and functional design. As shown in Table 1, we also include five hypotheses (H1–H5) that serve as a foundation for a subsequent qualitative and quantitative evaluations of the prototypes to empirically validate the value claims of an open SDS compared to the value of a closed SDS. We proceed to develop the requirements and DPs based on justificatory knowledge. We subsequently present the empirical results of our qualitative and quantitative evaluations.

4.1 Requirements and Design Principles for the Speech Dialog System

According to Walls et al. (1992), a design theory includes prescriptive instructions for how to realize more effective and feasible design and use. With regard to our design theory for an SDS, we must therefore identify the main requirements and DPs to help us to achieve these goals. According to Bunt's dialog theory (2000, p. 2), an SDS consists of *"structures of goals, beliefs, preferences, expectations, and other types of information, plus memory and processing capabilities"* that dynamically change during communicative acts as a reaction to other acts. In task-oriented dialogs in customer service, the goal of users is to express their concerns and inquiries in natural language to ensure that their requests are effectively handled. To this end, we identify the requirements related to DP prompt design, menu design, persona design, confirmation strategy, error management, and functional design. These requirements are essential to support the user through the dialog and to achieve the desired objective.

The first category of requirements is concerned with the design of system prompts as one major design aspect of SDSs. In this context, dialog theory assumes that communicative agents strive for rationality in reaching their goals (Bunt 2000), which in turn requires an effective design of an SDS. In recent years, an increasing number of studies on the prompt design of dialog systems have been published (Robertson et al. 2016; Jha 2019; Przegalinska et al. 2019). Overall, the academic literature agrees that system responses should be kept short because long messages would confuse the user (Delogu et al. 1998; McTear et al. 2016, p. 64). Moreover, direct and precise expressions and a strong task orientation should guide the answers of users (*R1*) (McTear et al. 2016, p. 64; Robertson et al. 2016; Jain et al. 2018). Additionally, for the sake of comprehensibility,

Lewis (2016, p. 222) advises using simple expressions and low variation of technical terms. The SDS should convey competence within the context of the application while remaining comprehensible (*R2*) (Verhagen et al. 2014). Based on the aforementioned, the following DP can be explicated:

DP1 *For SDS designers to shape an efficient dialog between customers and an SDS, ensure that the SDS employs brief but precise and goal-oriented prompts; such a design facilitates the understandability of the dialog while conveying competence in addressing the customer's goal and task (*Verhagen et al. 2014).

Aside from the task-oriented acts, dialog control acts are considered important for a smooth and successful communication according to dialog theory (Bunt 2000). Dialog control acts comprise social acts and behaviors for natural communication purposes. In this regard, another crucial stream of human–computer interaction research currently deals with anthropomorphism to examine the impact of human-like characteristics and design elements of conversational agents, the so-called "social cues" on user perception (Araujo 2018; Pfeuffer et al. 2019; Diederich et al. 2020). Anecdotal evidence has shown that anthropomorphic characteristics are not necessarily related to a higher trustworthiness of a system; instead, their impacts depend on the specific context. When the system is intended to replace a human expert (e.g., for customer support), human-like characteristics are considered beneficial for generating familiarity and trust with the agent. By contrast, the humanness of a system is not considered helpful when it is designed to substitute an existing computer system given the "automation bias" (Diederich et al. 2020). Guided by these findings, we adopt the view of human characteristics being positively related to the trustworthiness of conversational agents for deriving the requirements and DPs for our design theory in the context of customer support.

In customer service, customer satisfaction depends not only on measurable criteria (i.e., the time required to process the request) but also on social factors such as the feelings of users (Hudson et al. 2017). Therefore, a major aim is to create high-quality conversations that resemble human interaction in terms of not only expression but also the emotions generated (Lee and Choi 2017). According to the academic literature, users desire certain human characteristics when interacting with dialog systems. First, a dialog system should be honest and authentic (Przegalinska et al. 2019), that is, it should neither deny its status as a machine nor behave like one (Luo et al. 2019). The positive associations of an efficiently and rationally acting machine should be combined with the communication characteristics of a human interlocutor (R3) (Portela and Granell-Canut 2017). Nonetheless, the SDS should admit mistakes without making the user feel responsible for them to maintain user trust (R4) (Branham and Mukkath Roy 2019).

According to dialog theory, the communicative behavior of the agents, including communicative acts such as greetings, apologies, gratitude, agreement, should conform to the social norms and conventions of the specific context and application area (Bunt 2000). Following this recommendation, the mode of expression should correspond to the specific context of an application (Gnewuch et al. 2017). If the context of application allows informal language, users might want small talk, humor, sarcasm, and playfulness (Hill et al. 2015; Jain et al. 2018). However, the SDS should not show negative character traits such as being rude or offensive. Users prefer a friendly dialog partner without this

friendly behavior appearing artificial (R5) (Verhagen et al. 2014). With regard to the voice, there is no prevailing opinion on whether the voice of an SDS should generally be female or male (Luo et al. 2019). Nonetheless, Eyssel et al. (2012) state that users prefer the voice of their own gender. To summarize, the following DP should be considered when designing an SDS character:

DP2 *For SDS designers to enable customers having a human-like dialog with an SDS, prompts should be responsive to errors and their expressions should be appropriate to the customer service context, using natural and friendly phrases coupled with social cues for a more comfortable and trusting interaction (*Lee and Choi 2017; Gnewuch et al. 2017).

The third category of requirements is concerned with confirmation and error management strategies, which are recognized as major components of an SDS (Gnewuch et al. 2017). Confirmation strategies check whether the system has correctly captured the variables once the customer responds to a question or makes a request (R6). A distinction is made between explicit and implicit confirmation strategies (McTear et al. 2016, p. 214). Explicit strategies prompt users to actively confirm their input, whereas implicit strategies only require passive confirmation (Lee et al. 2010). In the latter case, the system repeats the mentioned inputs in connection with a new question. If the user answers this question, the system automatically confirms the variables (McTear et al. 2016, p. 214). If the variables formulated do not apply, the user can point to this and the system initiates the correction process (R7) (Lee et al. 2010; Mané and Levin 2008) conclude that users prefer implicit strategies; by contrast, McTear et al. (2016) consider implicit strategies as beneficial to a more efficient conversation. To assess these finding in more detail, we explore and evaluate whether an implicit or explicit confirmation strategy is preferable. The DP regarding the confirmation strategy is as follows:

DP3 *For SDS designers to ensure that an SDS has correctly captured all the required information during a conversation turn, a confirmation strategy should be implemented to guide customers in providing required and missing values for a structured and effective conversation (*McTear et al. 2016, p. *214).*

SDS error management similarly requires special attention (McTear et al. 2016, p. 266). Errors and dialog breakdowns can occur when user statements cannot be assigned to an intent (Uchida et al. 2019), which can create negative experiences and consequently reduce user trust in the SDS (Begany et al. 2016). To prevent the interruption of a conversation, an error management strategy is required (Opfermann and Pitsch 2017). Additionally, the error prompt should be based on the type of error. For example, the SDS must react differently to misunderstandings of the automatic speech recognition module than to unrecognized intents (R8) (Opfermann and Pitsch 2017). In the case that further errors occur despite error prompts, the SDS must react in a differentiated way to boost the chance of problem resolution (R9). Multi-stage error recovery strategies increase the level of assistance when the system repeatedly misunderstands the user. In particular, the error recovery strategies “ask” and “solve” according to Benner et al. (2021) are employed. The “ask” strategy includes the options for the customer to make another

request at any stage of the dialog and to rephrase the request or sentence after repeating the input options, whereas the "solve" strategy aims to actively provide solutions for avoiding the dialog breakdown. Overall, the following DP should be considered for a consistent error management strategy:

DP4 *For SDS designers to equip the SDS to handle errors (e.g., unrecognized intents, wrong navigation turns) and dialog aborts without interrupting the conversation for customers, a multi-stage error recovery strategy should provide customers with context-sensitive support to successfully communicate their requests (*Begany et al. 2016).

The final important set of requirements is related to the functional design of the dialog flow, which defines the rules for the entire dialog course and thus describes the users' different action alternatives (Handoyo et al. 2018). A logical dialog structure should enable the effortless handling of the system by incorporating the user perspective and using available information (Gardner-Bonneau and Blanchard 2007). The automatic verification and completion of user input ensures a more efficient dialog flow (Jain et al. 2018). For example, incomplete addresses can be completed with the help of the Google Maps API to enable a more efficient dialog (Vaira et al. 2018). As another example, the integration of mathematical checksums can help to validate credit card or customer numbers (Pearl 2016).

To summarize, on the one hand, all necessary user options should be integrated into the dialog flow to ensure completeness (R10) (McTear et al. 2016, p. 63); on the other hand, the number of functions of a task-oriented SDS should be limited, as an oversupply of functions can result in a higher development effort, an increasing error rate, and dissatisfied users (Michiels 2017). The functions should be designed to meet the expectations of users but avoid complex tasks that dissatisfy them (R11) (Kiseleva et al. 2016). To meet these requirements, the following DP should apply:

DP5 *For SDS designers to provide customers with a functional range that adds value to customer service, only domain-specific functions that meet user expectations should be included, but the functions should be limited to the essential ones to achieve customer objectives and avoid overwhelming customers with options (*Michiels 2017*).*

As stated in the methodology section, we conduct an empirical study to highlight the strengths and weaknesses in the user experience in the context of a task-oriented use case (Walls et al. 1992; Gregor and Jones 2007). To compare the effects of the open and closed dialog strategies on user experiences in detail, comparability between the systems is required. The main difference between the two strategies can be found in the menu-oriented structure of the closed dialog system. Menu design is a major category of requirements of the closed dialog strategy. Menu prompts belong to the category of system prompts, and they should also fulfill the requirements of being efficient, precise, and understandable (Robertson et al. 2016). Thus, we consider menu design as the equivalent category of requirements and DP for the "prompt design" of the open dialog strategy.

Menus are considered important when informing the user about the possible dialog paths, but an excessive number of menu options can overwhelm users (Bigot et al.

2013). In this context, anecdotal evidence has shown that the human memory is capable of remembering five to nine menu options (Miller 1956). With each additional option, the ability to remember is negatively affected (Bigot et al. 2013). Thus, the recommendation is to limit the number of menu options (R12) (Robertson et al. 2016) to a maximum of five (Bigot et al. 2013). In a similar vein, the arrangement of the options should be properly designed. In this case, the primacy-recency effect must be taken into account, according to which the information that is named first (primacy effect) or last (recency effect) is better remembered (Murdock 1962). Consequently, important or frequently requested menu options should be placed at the beginning or end to prevent errors and time-outs. Furthermore, the listing of the options to choose from should not be followed by additional information because such structure impairs the ability to remember the previously mentioned options (R13) (Bigot et al. 2013). In addition, Gardner-Bonneau and Blanchard (2007) recommend a strong distinction between the wording of individual menu options and commands (R14). Overall, the following DP should be considered when designing menus:

DP6 *To enable SDS designers to facilitate a menu-driven conversation between the customer and the SDS, the SDS should be equipped with a menu of up to five differentiated options within a conversation turn, with important to frequently requested menu options placed at the beginning or end of the dialog to allow for a goal-oriented dialog (*Lee et al. 2017).

The outlined requirements and DPs as well as the conceptual link between them are summarized in Table 2. As proposed by Gregor et al. (2020), DPs specify the mechanisms that SDS developers must implement to satisfy a particular set of requirements.

4.2 System Design

To allow for a naturalistic design, we collaborate with a German IT consulting company from Lower Saxony, for whom we build two SDS prototypes: one with an integrated open dialog strategy and another with a closed dialog strategy. We apply the design theory to an adventure booking portal (Adventure Guru) that is aligned with the business logic specifications provided by our cooperation partner. The design instantiations are built and tested in each iteration. Within the first iteration, the effectiveness of the prototype design is evaluated via cognitive walkthroughs with two user-experience designers, a creative technologist, and an expert for digital business. All experts have distinguished professional experience in the field of SDS and work for the IT consulting firm of our cooperation partner. In the second iteration, the refined design instantiations are subjected to user tests (comprising researchers and practitioners) to validate that the SDS variants can be used for booking, editing, or cancelling adventures. Following DP5, the SDS instances are equipped with a limited number of functions that allow users to book 12 different adventures (e.g., bungee jumping), edit bookings, cancel bookings, or obtain answers to frequently asked questions. To capture customer intents, we use Google's Dialogflow phone gateway as the customer interface, along with the underlying natural language understanding engine. Although we add some specific training sentences as well as alternative wordings and statements for model training, we are otherwise able

Table 2 Requirements and design principles for the design theory

Category	Requirement	Design Principle
Prompt design	**R1**: Short and precise prompts that contain only the most necessary information **R2**: Comprehensible prompts with a simple mode of expression and a small variation of technical terms	**DP1**: For SDS designers to shape an efficient dialog between customers and an SDS, ensure that the SDS employs brief but precise and goal-oriented prompts; such a design facilitates the understandability of the dialog while conveying competence in addressing the customer's goal and task (Verhagen et al. 2014).
Persona design	**R3**: Incorporation of social cues **R4**: Capability to admit mistakes without giving the user the feeling of being responsible for errors **R5**: Natural and friendly mode of expression that is aligned with the specific context of the field of application	**DP2**: For SDS designers to enable customers having a human-like dialog with an SDS, prompts should be responsive to errors and their expressions should be appropriate to the customer service context, using natural and friendly phrases coupled with social cues for a more comfortable and trusting interaction (Lee and Choi 2017; Gnewuch et al. 2017).
Confirmation and error management	**R6**: Prompt customers to provide the required and missing values **R7**: Implicit or explicit confirmation of the captured variables	**DP3**: For SDS designers to ensure that an SDS has correctly captured all the required information during a conversation turn, a confirmation strategy should be implemented to guide customers in providing required and missing values for a structured and effective conversation (McTear et al. 2016, p. 214).
	R8: Different reactions of the SDS depending on the type of error **R9**: Multi-stage error recovery strategy for an increased level of assistance when errors repeatedly occur	**DP4**: For SDS designers to equip the SDS to handle errors (e.g., unrecognized intents, wrong navigation turns) and dialog aborts without interrupting the conversation for customers, a multi-stage error recovery strategy should provide customers with context-sensitive support to successfully communicate their requests (Begany et al. 2016).
Functional design	**R10**: Integration of task- and support-oriented functions **R11**: Provision of a limited number of functions to avoid an oversupply of information	**DP5**: For SDS designers to provide customers with a functional range that adds value to customer service, only domain-specific functions that meet user expectations should be included, but the functions should be limited to the essential ones to achieve customer objectives and avoid overwhelming customers with options (Michiels 2017).
Menu design	**R12**: Limited number of menu options (up to five menu options) **R13**: Prevention of errors and time-outs by placing important or frequently requested menu options at the beginning or end of a dialog **R14**: Strong distinction between the formulations of the individual menu options and commands	**DP6**: To enable SDS designers to facilitate a menu-driven conversation between the customer and the SDS, the SDS should be equipped with a menu of up to five differentiated options within a conversation turn, with important to frequently requested menu options placed at the beginning or end of the dialog to allow for a goal-oriented dialog (Lee et al. 2017).

to rely on an already well-functioning natural language understanding engine. The extra training phrases are not fed into the model by speech-to-text conversion, but rather as plain text. We integrate and utilize the Parloa development platform for the design and management of dialog flows. Furthermore, we design the SDS prompts for the instances

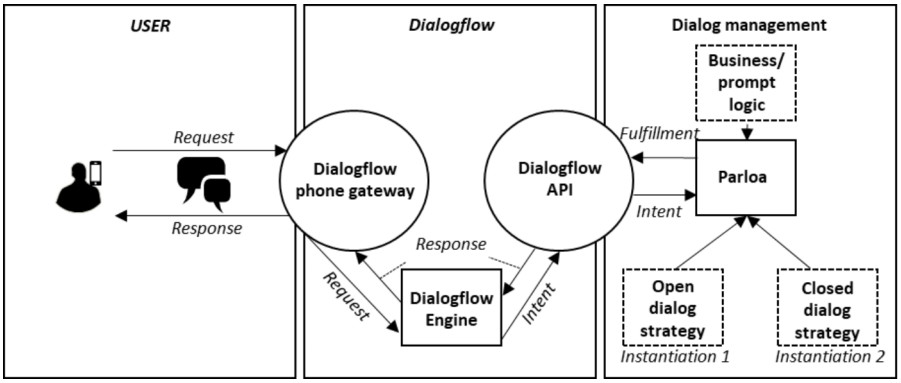

Fig. 3 System architecture

to deal with customer booking inquiries and to handle customer conversations based on the business logic specifications and the devised DPs (cf. Figure 3).

Open SDS instantiation – DP1: The welcome prompt of the open SDS instantiation welcomes and invites the user to start the conversation with the open question: "How can I help you?" (cf. Figure 4, right). The user decides on the further course of the conversation by either posing a question or placing a booking. However, the lack of clear instructions also causes considerable uncertainty, which is intercepted if the user does not react within 4 s, after which the system automatically informs the user about possible central functions. This interception does not constitute instructions for action as is the case in the closed SDS, but it is intended to provide information about the available functions. Instead of enumerating individual menu items and querying individual variables, the open SDS allows the user to input several variables within a single statement. For example, the selected experience, the number of people participating, and the date can be recorded within one statement. However, if the user specifies only one variable, the system will proactively ask for the remaining input to complete the process step. Even if the system asks for a specific variable, the user can still name additional information concerning several variables.

Furthermore, the SDS is capable of telling jokes and engaging in simple small talk; nonetheless, to ensure that the system does not lose task orientation, the prompts always end with the question of the respective process step (cf. Figure 5). If the user deviates too far from the actual task so that the system cannot interpret the statement, an error prompt occurs. In such cases, the multi-level error recovery strategy enables the user to correct errors by intervening with statements such as "this is not correct; the booking should actually be made for the [date]." If the correction results in another error, the system provides an example statement and, if necessary, refers to the corresponding help intent. The level of assistance only increases after the second error. However, the error prompts remain short and rely on the user's initiative to independently correct the error. The correction within the open SDS can be realized through a single user turn (DP4).

Closed SDS instantiation – DP6: The closed SDS starts by welcoming the user, naming the menu options, and offering navigation hints (cf. Figure 4, left). Booking experiences constitutes the central functionality of the SDS; thus, it is the first named main

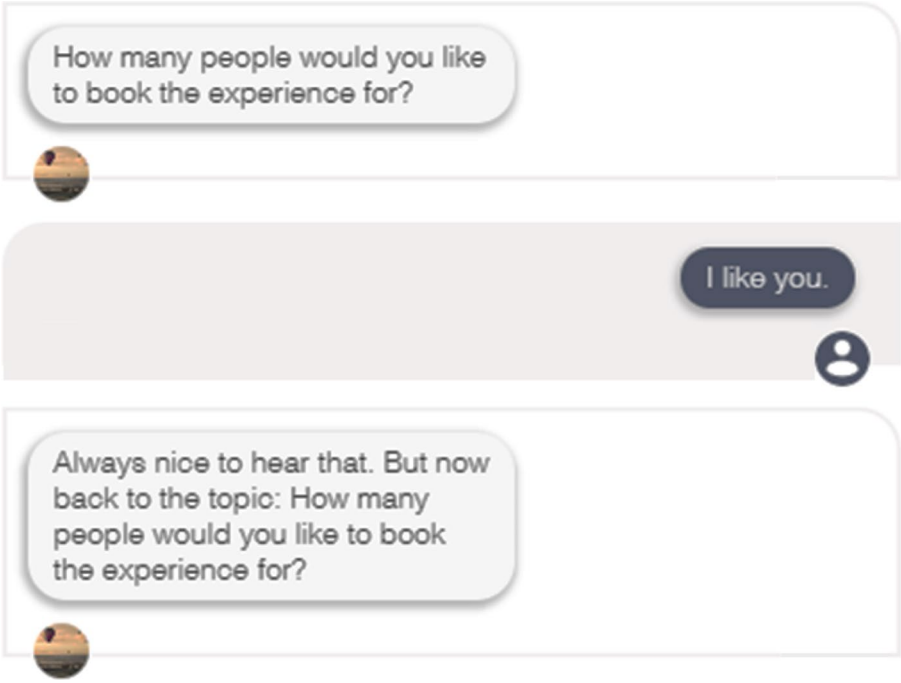

Fig. 4 The humorous and anthropogenic side of the SDS (DP2)

menu option. We set the frequently asked questions last as a form of assistance in case of uncertainties.

In addition to the menu options, the closed SDS lists the command options to help the user in understanding how to operate the SDS. Due to the tree navigation structure of the closed dialog strategy, an incorporated "return" command ensures easy and quick navigation corrections during the booking process. Additionally, by selecting the "main menu" command, the user can cancel ongoing processes and return to the main menu, where only the welcome prompt is repeated, as the user should still be familiar with the command options. After selecting a menu path (e.g., "book experience"), the menu items of the next navigation level are listed and necessary input variables such as experience category, experience, number of participants, and date are successively captured. If errors occur despite the coherent closed dialog strategy, the SDS responds with the prompt "sorry, I'm probably hearing particularly badly today. Could you please repeat that?" The SDS admits its mistake in a funny and friendly way and asks the user to repeat the statement. Different responses of the error prompt ensure that the SDS does not repeat itself in the course of the dialog (DP2). The available options explicitly express that the user should repeat and not rephrase the input. If the user still fails to select the desired option, the system assistance is increased. For example, the system advises to follow the exact wording of the menu options before repeating them afterwards (DP4).

In the formative evaluation cycles, the idea that implicit confirmation is not always understood as data entry confirmation has become apparent; hence, we implement the explicit confirmation after each process step, although such approach lengthens the dia-

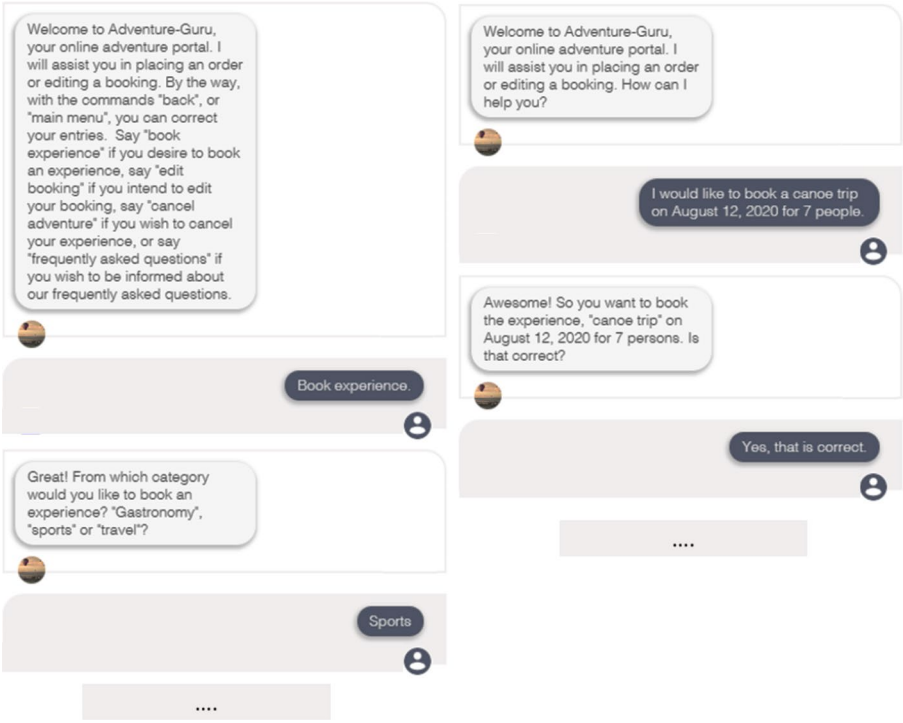

Fig. 5 Welcome prompt – closed SDS (left) and open SDS (right)

log. Thus, successful user inputs are explicitly confirmed by "all right," "understood," and "OK" in both instantiations. After the explicit confirmation, the user confirms the repeated variables, receives a booking number (at the end of the booking process), or returns to the main menu (DP3).

To explicitly summarize the described technical realization of our DPs, we outline the corresponding implemented design features in Table 3. These design features reflect a series of specific design choices that instantiate each DP (Meth et al. 2015; Schoormann et al. 2021).

5 Evaluation of the Speech Dialog Systems

For the evaluation, we follow the framework of Venable et al. (2016) to ensure alignment between our research goals and framework settings, demonstrate design utility, and generate implications for research and practice. The applied framework comprises four steps: (1) explicate the goals of the evaluation, (2) select the evaluation strategy, (3) determine the properties to evaluate, and (4) design the individual evaluation episodes.

(1) Given our need to analyze the utility and efficacy of both SDS dialog strategies with respect to achieving a specific goal (Venable 2006), our evaluation aims to test the rigor of both designs by assessing their functional effectiveness (Venable et al. 2012). Furthermore, we aim to outline the strengths and weaknesses of both dialog strategies

Table 3 Design features

DPs	Design Features
DP1	• Welcome prompt providing necessary information to interact with the SDS • Short prompts of no more than two sentences (unrelated to the enumeration of menu options) • Enable the input of multiple variables within a single statement • Proactive requests of any missing information if it is insufficient for the process to continue • Always end a prompt with reference to the next process step
DP2	• Dialogflow phone gateway as application programming interface to the customer • Welcome prompt with greetings and introductions • Humorous responses to statements such as "I like you," "I love you" (love_intent), "I hate you," and "this sucks" (insult_intent) • Error prompts equipped with funny and friendly statements in which errors are acknowledged • Multiple responses to the error prompt to ensure that the SDS does not repeat itself (random selection of responses)
DP3	• Implicit or explicit confirmation after each process step
DP4	• If no response is obtained for 4 s, a prompt with information about the available functions is activated • Users can correct an input at any stage of the dialog • After two successive errors, a prompt is provided to help with the input options ("ask" and "solve" strategies according to Benner et al. (2021))
DP5	• Book adventures (e.g., bungee jumping) • Edit bookings • Cancel bookings • Answers to frequently asked questions
DP6	• Enumeration of menu options and navigation hints in the welcome prompt • Include the booking experience option as the first menu option • Include the frequently asked questions as the last menu option • "Return" command to ensure easy and quick navigation corrections during the booking process • "Main menu" command to revert to the beginning of the booking process, without repeating the command options • Tree structure to successively query experience category, experience, number of participants, and date

by empirically testing the user experience in customer service, thereby reducing design uncertainty and risk.

(2) According to Venable et al. (2016), the "human risk and effectiveness" strategy is suitable for problem spaces where user design is paramount (Gnewuch et al. 2017; Diederich et al. 2020). Therefore, with our focus on the design risks related to the interaction between users and the SDS, we also follow this strategy.

(3), (4) By applying formative qualitative evaluation methods during the first two iterations and a summative quantitative evaluation method at the end of the third iteration, we operationalize the human risk and effectiveness strategy in a naturalistic framework.

The goal of the formative evaluation cycles is to improve design and implementation to ensure effective instantiations; by contrast, the purpose of summative evaluation is to capture the usability of the final design. After the completion of the first prototyping phase, the initial SDS prototypes are evaluated by four SDS experts who analyze the systems in terms of usability and feasibility via the cognitive walkthrough method. In human–computer interaction research, cognitive walkthrough represents an effective method for evaluating the design of a user interface in early prototyping phases based on cognitive theory (Rieman et al. 1995). For the open SDS, the results of these cognitive walkthroughs particularly related to issues of unrecognized intents and prompt wording; for the closed SDS, the results pertain to the number of options and prompt length as well as the categorization and order of prompts. The idea that the implicit confirmation strategy poses the most challenging issue for the experts becomes apparent, as it is not always understood as data entry confirmation (cf. DP3). Based on the experts' feedback, we refine the prototypes in the second design cycle and correct major pitfalls (e.g., change to an explicit confirmation strategy, addition of test phrases for the model training, improvement of prompt design, categorization of the service offerings). By subsequently conducting in-house user tests with five potential users, we aim to ensure that users can master the tasks in the dialog systems without prior experience and further assistance. We record, transcribe, and analyze the conducted user tests through a qualitative content analysis (Mayring 2001). The results from the user tests reveal rather minor issues (wording of the prompts, isolated intent detection issues), which are resolved by the further refinement of the instances.

After the second iteration, the instantiations are prepared for the final evaluation, a two-phase experiment with 205 participants. The properties to be evaluated for the comprehensive summative evaluation are captured in the hypotheses in Table 4. The hypotheses represent "statements required to test whether the design satisfies the requirements" (Gregor and Jones 2007, p. 319).

5.1 Experimental design

To test the hypotheses, we conduct a two-phase experiment. In the first phase, the participants are asked to familiarize themselves with both the open and the closed SDS by performing two tasks in each instantiation:

- Task 1: "*Call Adventure Guru to book the canoeing experience, on August 12, for seven people.*"
- Task 2: "*In the same call, you want to edit your booking and change the number to five people*."

The tasks provide a clear and comprehensible use case for the interaction with the SDSs and allowed for comparability across participants. We log the user activities (i.e., completion time), errors made (number of corrections), and number of dialog steps required to complete the tasks. The log file is automatically created by an integrated function of

Dialogflow as soon as the participants begin their task by calling the Adventure Guru. In the second phase of the experiment, the participants complete an online survey that captures the user experience with both instantiations. The constructs and items operationalizing the survey constitute existing validated measures (cf. Appendix A.4). We use the construct of perceived humanness from Gnewuch et al. (2017) to test H1 (resp. DP2, with the aim of enabling customers to have a human-like dialog with an SDS). Furthermore, we employ the constructs of the Subjective Assessment of Speech System Interfaces framework (Hone and Graham 2000); this framework is a standardized user-experience questionnaire for conversational interfaces, which features a broad selection of user-experience dimensions (Kocaballi et al. 2019). To test H2 and thereby examine the DP3 design, we use the construct of habitability, which refers to "the extent to which the user knows what to do and knows what the system is doing" (Hone and Graham 2000, p. 23). In addition, the construct of system response accuracy is utilized to test H3, which examines the DP4 design of error handling, and the construct of likability is used for testing H4 by assessing preferences between an open (or DP1) and a closed menu design (or DP6). We utilize a five-point Likert scale to measure all constructs. We further conduct a small-scale preliminary study to test the comprehensibility of the items and ensure validity and reliability by refining the measurement instrument (Straub et al. 2004). To strengthen and extend the testing of the hypotheses, we include the results from the log file analysis to support the subjective assessment of the participants with objective information about the system tests, providing additional validation of DP4, DP1, and DP6 and testing the functional effectiveness anchored in DP5. In doing so, hypotheses H1–H5 allow us to test the aims of DPs implicitly through the implementation of the SDS designers (implementers) and explicitly through customers (user). An overview of the hypotheses and the corresponding measures is presented in Table 4.

To recruit participants, we use several social media groups and the public news hub of our cooperation partner. Of the 214 survey participants, nine have to be sorted out due to incomplete data (e.g., no registered call in the system). The descriptive statistics of the remaining 205 participants are shown in Table 5.

5.2 Experiment results

Following the approach of Diederich et al. (2020), we analyze our data by using descriptive statistics and conducting statistical hypothesis tests. Descriptive statistics from the logging information show that participants are able to complete tasks more efficiently with the open SDS with an average of 9.66 dialog steps and nearly 50 s less time than with the closed SDS, in which an average of 11.72 dialog steps are required. However, navigation errors occur more frequently with the open SDS (average 1.46) than with the closed SDS (average 1.07). Moreover, the success rate in fulfilling both tasks for the closed system is a convincing 96.10%, compared to 91.22% for the open SDS. The survey data are validated for the internal consistency reliability of our latent constructs by calculating the Cronbach's alpha (α) and the composite reliability that exceeds the recommended limit of 0.7 (Nunnally and Bernstein 1994). Descriptive statistics reveal higher subjective average scores using the open SDS in the area of perceived humanness, system response accuracy, and likability, whereas the closed SDS is more convincing in habitability.

Table 4 Hypotheses and corresponding measures

Hypotheses	Corresponding Design Principle	Log File Measure (Phase 1)	Survey Measure (Phase 2)
H1: If the human-like SDS follows a strategy of open dialog, it is perceived as more human-like than if the SDS follows a strategy of closed dialog.	DP2	-	Perceived humanness (Gnewuch et al. 2018)
H2: If the human-like SDS follows a strategy of open dialog, it is perceived as more habitable/comprehensible than if the SDS follows a strategy of closed dialog.	DP3	-	Habitability (Hone and Graham 2000)
H3a: If the human-like SDS follows a strategy of open dialog, the system response accuracy is perceived as higher than if the SDS follows a strategy of closed dialog.	DP4		System response accuracy (Hone and Graham 2000)
H3b: If the human-like SDS follows a strategy of open dialog, fewer errors occur than if the SDS follows a strategy of closed dialog.	DP4	Number of errors	-
H4: The human-like (open) SDS contributes to an improved user experience in task-oriented settings of customer service compared to those of the closed SDS.	DP1, DP6	-	Likability (Hone and Graham 2000)
H5: The human-like (open) SDS shows a higher functional effectiveness and performance in task-oriented settings of customer service than the closed SDS.	DP1, DP5, DP6	Task completion success rate, Duration, Number of dialog steps	-

To assess the significance of the difference between the systems for each of the examined variables, we first analyze our continuous survey data and the logging variables for univariate normality by applying the Kolmogorov–Smirnov test, which shows significant results ($p<.01$) for all continuous variables, indicating that the sample distribution do not follow a normal distribution (Field 2009). Based on these pre-tests, we use the non-parametric Wilcoxon signed-rank test to conduct the hypothesis tests of our related samples (Wilcoxon 1992). Next to the Wilcoxon signed-rank test, we use a chi-square test for examining the difference between the task success rates due to the dichotomous nature of the variable (1 = successful task completion of both tasks, 0 = unsuccessful task

Table 5 Descriptive statistics of the participants' demographical data (n = 205)

Demographical data		Absolute	Relative
Gender	Male	80	39%
	Female	125	61%
Age	Generation Z (18–24 years)	72	35.12%
	Generation Y (25–44 years)	109	53.17%
	Generation X (45–65 years)	24	11.71%
Education level	High school	94	54.15%
	College degree	111	45.85%
SDS experience	No experience	44	21.46%
	Some experience, but no regular usage	120	58.54%
	Regular user	41	20.00%

completion of both tasks). The descriptive statistics and the results of the hypothesis tests are highlighted in Table 6 (Wilcoxon signed-rank test) and Table 7 (chi-square test).

We can confirm H1 (T+, Z = -5.545, $p < .01$), denoting that the open SDS design strategy is significantly perceived as more human-like than the closed one. We reject H2 (T-, -2.450, $p = .014$), as the closed SDS is perceived as more comprehensible than the SDS that follows an open dialog strategy. We identify significant differences in perceived system response accuracy in favor of the open SDS instance (H3a, T+, -2.234, $p = .025$), but the number of errors is significantly higher with this strategy (H3b, T+, -2.763, $p = .006$). Overall, the user experience with the open SDS is perceived as significantly more likeable than with its closed counterpart (H4, T+, -5.033, $p = .006$). With regard to functional effectiveness (H5), we obtain mixed results depending on the definition. In terms of the time (T-, -10.344, $p = .000$) and the dialog steps (T-, -8.027, $p = .000$) required to successfully complete the assigned tasks, we observe significant advantages for the open SDS. However, when considering functional effectiveness as the total number of successfully completed tasks, we note significant advantages for the closed SDS ($\chi^2 = 2.734$, df = 1, $p > .05$).

The logging data are clearly objective in nature. To substantiate the associated hypotheses (H3b, H5) from a subjective point of view, we conduct a qualitative content analysis of the open-ended answers in the survey, which allows the participants to optionally report what they like and dislike about the two SDS variants. Two researchers inductively code the occurring patterns in the text fields. The coding is validated using Krippendorff's alpha ($\alpha = 0.83$) (Krippendorff 1989). We count the subjective positive and negative sentiments in the text fields per variable (number of errors, dialog steps and duration to task completion). We summarize the counts of the examined variables that are assigned to H5 because the statements often refer to the effectiveness and efficiency of the task fulfilment and therefore cannot be clearly distinguished. The results from the qualitative content analysis support the findings from the hypothesis tests (cf. Table 8).

Considering the demographic data, we perform a non-parametric pendant to the one-way ANOVA, the Kruskal–Wallis H test (Kruskal and Wallis 1952), to evaluate the differences between the group distributions. The Kruskal–Wallis H test shows strong evidence of intergenerational differences in the perceptions of humanness (H = 15.921,

Table 6 Results from the Wilcoxon signed-rank test (n=205)

Construct	α	Mean	Median	SD	Sum of Ranks (T)	Z Score	*p*-value	Hypotheses
Perceived humanness								
Open SDS	0.812	3.431	3.333	1.039	10821.50 [a]	-5.545 [c]	0.000**	H1 supported
Closed SDS	0.702	2.881	3.000	0.904	3713.50 [b]			
Habitability								
Open SDS	0.737	3.286	3.250	0.956	7460.50 [a]	-2.450 [d]	0.014*	H2 rejected
Closed SDS	0.801	3.540	3.500	1.049	11260.50 [b]			
System response accuracy								
Open SDS	0.778	3.627	3.750	0.979	10018.15 [a]	-2.234 [c]	0.025*.	H3a supported
Closed SDS	0.750	3.383	3.500	0.971	6817.50 [b]			
Number of errors								
Open SDS	-	1.460	1.000	1.468	6323.50 [a]	-2.763 [c]	0.006**.	H3b rejected
Closed SDS	-	1.070	1.000	1.219	3687.50 [b]			
Likability								
Open SDS	0.869	3.905	4.000	0.905	12888.50[a]	-5.033[c]	0.000**	H4 supported
Closed SDS	0.832	3.452	3.400	0.879	5256.50[b]			
Duration for task completion (in sec.)								
Open SDS	-	134.532	125	36.995	1123.00 [a]	-10.344 [d]	0.000**	H5 partially supported (considering the results in Table 7)
Closed SDS	-	184.085	178	23.656	16455.00 [b]			
Dialog steps to achieve task completion								
Open SDS	-	9.666	9.000	2.813	2168.50 [a]	-8.027 [d]	0.000**	
Closed SDS	-	11.720	11.000	1.648	12537.50 [b]			

a. Closed SDS < Open SDS

b. Closed SDS > Open SDS

c. Based on positive ranks (T+)

d. Based on negative ranks (T-)

** $p < .01$, * $p < .05$, not significant (n.s.) for $p > .05$

Table 7 Results from the chi-square test (n=205)

	n	Task 1 Completion (Success Rate)	Task 2 Completion (Success Rate)	Tasks 1 and 2 Completion (Success Rate)	Hypotheses
Open SDS	205	197 (96.1%)	187 (91.22%)	187 (91.22%)	
Closed SDS	205	201 (98.05%)	197 (96.1%)	197 (96.10%)	

Chi-square test

$\chi^2 = 2.734$, df = 1, $p > .05$

H5 partially supported (considering the results from Table 6)

df=2, $p<.01$), habitability (H=22.582, df=2, $p<.01$), system response accuracy (H=26.279, df=2, $p<.01$), and likability (H=22.394, df=2, $p<.01$) in the closed SDS.

Table 8 Results from the qualitative content analysis of the open-ended questions

Hypotheses Results (Objective Data)		Coding Results (Corresponding Subjective Impression)	
		Positive Sentiment	**Negative Sentiment**
H3b: Number of errors			
Open SDS	Open SDS > Closed SDS	12	37
Closed SDS		20	21
H5: Functional effectiveness			
Open SDS	Dialog steps to achieve task completion:	74	3
Closed SDS	Open SDS < Closed SDS Duration for task completion: Open SDS < Closed SDS Tasks 1 and 2 completion rate: Open SDS < Closed SDS	9	47

A pairwise comparison using post-hoc (Dunn–Bonferroni) tests reveals that this result is predominantly due to the difference between Generations Z and X as well as Y and X, with the older generation showing a stronger bias in all respects toward closed SDS and the effect of the difference increasing with age difference. We also observe statistically significant differences between Generations Z and Y in terms of habitability, which can be interpreted as the older the user is, the more comprehensibly closed SDS are perceived (cf. Table 9).

These findings are also confirmed when analyzing the preferences of users according to the survey results (cf. Table 10). Overall, 68.78% of users prefer the open SDS, whereas 31.22% consider the closed SDS as more preferable. The younger user groups of 18–24 and 25–44 years old prefer the open SDS, whereas the older user group of 45–65 years old clearly opt for the closed SDS.

6 Discussion

Guided by the DSR paradigm, the primary purpose of this study is to devise and evaluate a design theory for an SDS dialog strategy in customer service. The proposed design theory including 14 requirements and five DPs is informed by the principles of dialog theory (Bunt 2000) and related work in prior conversational agent and SDS research; it is also empirically validated in three iteration rounds through five hypotheses. In doing so, we contribute to research and practice in several ways. First, we enrich the body of knowledge by proposing a design theory that provides codifying design knowledge for a class of artifacts (SDS dialog strategies) to address a class of problems according to Walls et al. (1992) and Gregor and Jones (2007). This type of knowledge can be referred to as "nascent design theory," which provides "knowledge as operational principles/

Table 9 Results from the pairwise intergenerational comparison (post-hoc)

Closed SDS	Generation	Mean	SD	Pairwise Z-Y	Pairwise Z-X	Pairwise Y-X
				Z-Score	**Z-Score**	**Z-Score**
Perceived humanness	Generation Z	2.796	0.768	0.217 $^{n.s}$	-3.897** †0.865	-3.588** †0.654
	Generation Y	2.768	0.899			
	Generation X	3.653	0.960			
Habitability	Generation Z	3.163	0.994	-3.023** †0.225	-4.552** †1.050	-2.722* †0.486
	Generation Y	3.631	1.000			
	Generation X	4.260	0.988			
System response accuracy	Generation Z	3.135	0.882	-1.523 $^{n.s}$	-5.102** †1.221	-4.308** †0.807
	Generation Y	3.349	0.956			
	Generation X	4.281	0.795			
Likability	Generation Z	3.264	0.799	-1.093$^{n.s}$.	-4.668** †1.083	-4.143** †0.769
	Generation Y	3.406	0.871			
	Generation X	4.233	0.745			

** $p < .01$, * $p < .05$, not significant (n.s.) for $p > .05$; † effect size for significant results (Cohen's d)

Table 10 User preferences according to user group (open vs. closed SDS)

User Group	Open SDS	Closed SDS	Total
Generation Z (18–24 years old)	27.32%	7.80%	35.12%
Generation Y (25–44 years old)	38.54%	14.63%	53.17%
Generation X (45–65 years old)	2.93%	8.78%	11.71%
Total	68.78%	31.22%	100.00%

architecture" according to Gregor and Hevner (2013, p. 342). With this contribution, we respond to recent calls for more design knowledge on conversational agents for enhancing user experience in the customer service context in particular (Gnewuch et al. 2017). In the next sections, we discuss the main findings of this study prior to highlighting the major implications for research and practice. In addition, the limitations of this study are outlined with further propositions for future research.

6.1 Implications for Research and Practice

The insights gained in the three iteration rounds of the applied DSR approach contribute to an iterative revision and refinement of our design theory. Based on the key findings of the evaluation rounds, we are able to derive manifold implications for research and practice, which are concerned with the effectiveness and user experience of the proposed design theory. The main findings to the tested hypotheses and the corresponding implications for research and practice are summarized in Table 11.

With regard to the perceived humanness as one of the hypotheses (H1), we find support for the notion that the open SDS is perceived more human-like than the closed system. This result is not only quantitatively validated but is also indicated by the qualitative answers of the survey participants, with positive comments on the humanness of the open SDS, such as "it feels almost like talking to an employee" or "it was like a normal conversation with a real person." In comparison, the strict enumeration of options makes

the closed system appear cold and robot-like. Some users are annoyed by the closed system, as indicated by statements such as "the long announcements are annoying," "mechanical communication," or "too much talk, too many options." Negative attitudes toward SDSs are due to the users' discomfort and distrust when talking to a machine without a personality (Luo et al. 2019). These findings have several implications for research and practice. First, they are consistent with social response theory (Nass and Moon 2000; Moon 2000) and support the human–human trust perspective, according to which anthropomorphic characteristics tend to positively affect user trust (Gnewuch et al. 2017; Seeger and Heinzl 2018). Accordingly, we can confirm the findings of previous studies that human-like characteristics are considered beneficial for the design of conversation-based technologies when the system is intended to substitute a human expert, for example for customer support (Diederich et al. 2020).

We also find that the participants hardly used some functions of the open SDS. For example, the participants are only interested in performing their task and showed no initiative to utilize the small talk function of the open system. Instead, the function is triggered in a few cases and only in an unintentional manner, which in the dialogs caused more misunderstandings than being useful. Thus, small-talk intents should be avoided in a task-oriented SDS because too many different intents increase error probability. This finding is consistent with one of the major assumptions of dialog theory, which posits that task-oriented dialogs are instrumental, with people only engaging in a dialog when they intend to achieve a particular task or goal (Bunt 2000). One major implication that can be derived from this finding is that the design of task-oriented SDSs should be different from the design of social SDSs. Similarly, prior research has concluded that interactions with voice assistants, as is the case with SDSs, should be designed differently than in conventional human–computer interactions (Schmitt et al. 2021). Among others, the human-like design of voice assistants should be context- and task-dependent. Therefore, the investigation of the main similarities and differences between task-oriented and social SDSs in future research would help to enhance the understanding of how to design desirable AI-based digital assistants for different task types.

When investigating the habitability of the open SDS compared to the alternative (H2), we find that the closed SDS is perceived as more habitable than the open SDS. One reason for the higher habitability with the closed SDS is that this form is still predominant in business practice (Dale 2016). Users who are unfamiliar with open dialog strategies feel overstrained when using them. For the design of task-oriented SDSs, this finding has several important implications. On the first call, more assistance should be provided to carefully familiarize users step-by-step with the open system. In particular, the welcome prompt has a significant influence on user expectations; hence, a brief explanation of available self-service options would be useful prior to posing the open question on user intent. By naming the various options, users can easily initiate the desired process and start the conversation without making any mistakes, similar to a closed system. Once users are familiar with the open system (i.e., for subsequent calls or when returning to the main menu), the level of assistance can be reduced. Furthermore, the findings indicate that the clear confirmation of inputs in the closed system is beneficial for enhancing habitability toward the system. Accordingly, the implicit input confirmation should be taken into account more consequently in the design of the open SDS. However, an issue that remains unanswered relates to how an optimal level of assistance can be achieved in

Table 11 Main findings and implications for research and practice

H	Findings	P	Implications and Propositions
H1	F1a: The open SDS is perceived as more anthropomorphic than the closed SDS.	P1a	In customer service, open SDS should be designed with human-like characteristics when the system is intended to substitute a human expert (human–human trust perspective).
	F1b: Users do not utilize small-talk intents when using the open SDS in task-oriented settings.	P1b P1c	The design of task-oriented SDSs should be different from the design of social SDSs such as intelligent personal assistants. Small-talk intents should be avoided in a task-oriented SDSs to reduce the probability of error detection.
H2	F2a: Users perceive the closed SDS as more habitable than the open SDS due to the clear options at any step of the booking process. F2b: The open mode of expression in the open SDS leads to confusion among the users.	P2 P2b	Users should be provided with more assistance during the first call to increase their familiarity with the open system: • Include a brief listing of self-service options in the welcome prompt. • The level of assistance can be reduced when the user is familiar with the system (in subsequent calls or when returning to the main menu).
H3a; H3b	F3a: The open SDS is perceived to be less error-prone than the closed SDS. F3b: According to the recorded data, more errors occur in the open SDS than in the closed SDS. F3c: The number of errors substantially varies from user to user, depending on the target group.	P3a	Help prompts should address the needs of different user groups more individually while keeping the user initiative in an open SDS depending on the context.
		P3b	Help should be provided only in the dialog steps in which the help is relevant rather than at any time.
		P3c	Instead of repetitions, help should already be offered after the first prompt with rewordings requested by the system.
		P3d	To familiarize users with the help functions, the availability of help prompts should already be mentioned in the first prompt of the first call and a short example should be given.
		P3e	Users should be forwarded to an employee after a certain number of errors or after expressed frustration.
	F3d: Unforeseen errors still occur despite several test phases.	P3f P3g	SDS must be continuously supervised by qualified personnel to eliminate sources of error in the long term. Further research is needed to provide deeper insights into the importance of different system characteristics on the user experience (e.g., based on frequency analysis, factor analysis, or other ranking methods).
H4	F4: The open SDS is perceived as more likeable than the closed SDS.	P4	To enable a human-like conversation and support the human–human perspective, the open expression mode should be considered when designing an SDS (cf. P1a).

Table 11 (continued)

H	Findings	P	Implications and Propositions
H5	F5a: The open SDS performs better than the closed SDS in terms of the duration for task completion and the number of dialog steps required to achieve task completion. F5b: The closed SDS performs better than the open SDS in terms of the completion success rate. F5c: Only a few users employ the slot filling function of the open SDS.	P5a	To improve the completion rate in the open SDS, the propositions made in P3a–P3f should be considered when designing the error recovery strategy to avoid future dialog breakdowns.
		P5b	To make users aware of the slot filling function, the welcome prompt of the first call should refer to this function and provide an example.
-	F6a: The majority of the users (68.8%) prefer the open and less than one-third (31.2%) favor the closed SDS. F6b: Age-related differences emerge: younger users prefer the open SDS, whereas older users opt for the closed SDS.	P6a P6b	The design of an SDS should be contextualized and individualized to meet the demand of users of all ages. Future research could explore the impact of hybrid SDSs on the user experience of various user groups (e.g., different age groups, gender, application domains).

H = Hypothesis; P = Proposition

an open SDS while benefiting from open expression for an intuitive human-like conversation, which is frequently perceived as positive by the users.

System response accuracy and the number of errors during the dialogs are additional aspects that substantially affect user experience (H3a/H3b). The findings based on the analyzed quantitative and qualitative data underline the importance of an efficient and error-free dialog. The operation of both systems is generally perceived as easy to learn. However, the users are more satisfied with the control system in the closed SDS due to the higher predictability of communication. On average, both tasks are completed faster in the open system, which is also confirmed by the subjective perception of the users. The majority of users indeed perceive the system response accuracy of the open system as higher than that of the closed system, based on their subjective perception that the open SDS makes fewer mistakes than the closed system. However, this perception is contradictory to the recorded system data, which reveal that users made more mistakes in completing the two tasks in the open system. The lower level of habitability as described above could be one reason why the number of errors is higher in the open system. The contradiction between the perceived system response accuracy and the actual number of errors based on the logging information implies that other system characteristics such as likability or perceived humanness may be more important for users of SDSs than system response accuracy. Thus, user satisfaction with the open SDS in terms of likability or humanness may lead users to underestimate the error rate. However, further research is needed to gain deeper insights into the importance of different system characteristics on the user experience. Future design studies could rank the requirements and DPs by their relative importance (e.g., based on frequency analysis, factor analysis, or other ranking methods).

The number of errors made substantially varies from user to user, with some users sharing their impression with comments such as "The system understood me and was easy to control" or "Good speech recognition." By contrast, other users experience considerable problems with fulfilling their tasks in the open SDS and criticize the number of errors in the open system. Various statements such as "You're somewhere in a menu and can't get any further, even though yelling at the phone" reflected such result. As indicated by DP4, a high priority in designing an SDS should be allocated to the successful handling of errors by including a multi-stage error recovery strategy that provides users with context-sensitive support to successfully communicate their request. Although this DP is considered when designing the open SDS, including two different test phases with different participants in which the error recovery strategy is iteratively improved, unforeseen errors still occur. Frequent failures to recognize user input causes user dissatisfaction and represents a major challenge in the development of SDS (Goetsu and Sakai 2019). Thus, the findings concerning error recovery strategy indicate further issues for improvement.

Given the varying preferences and needs of different user groups, system design should allow for tailored levels of help prompts. Hence, help prompts should be more detailed when the user specifically asks for support. By calling up help prompts, users could control the level of system help themselves. Additionally, more contextualization is required to avoid unnecessary errors and misunderstandings. Instead of allowing users to access the help at any time, this function should only be possible in the dialog steps in which help is relevant. Furthermore, an early provision of help can avoid unnecessary errors. To increase the probability that users request help when necessary, the function can be mentioned in the welcome prompt in the first call. For the developed use case, the system could formulate an example statement with several filled slots. The users then customize the sentence with their desired content and in this way learn how to use the system. Thus, the knowledge gap regarding slot filling can be closed.

Another way of avoiding dialog breakdowns is to implicitly integrate a de-escalation intent. Depending on the length of the dialog, the system should forward the user to a human employee after a certain number of errors to avoid dialog breakdowns. Overall, two steps are necessary: supplementing unexpected user statements in the rules of dialog management and formulating prompts more purposefully when the number of errors is too high. SDSs must therefore be supervised by qualified personnel over the course of their deployment to eliminate sources of error in the long term.

Likability is another major aspect to be considered when designing an open SDS (H4). The survey participants generally prefer the flexibility of the open system, which allows the user to fill several slots at once and helps to determine the course of the system. The open SDS is quantitatively and qualitatively rated as more friendly than the closed SDS, with the participants expressing statements such as "friendly voice" and "I found the robot very nice." In addition, the users describe the navigation in the open system as intuitive and on average have more fun with the system. The open expression mode should therefore be possible throughout the SDS to enable a human-like conversation and to support the human–human perspective according to social response theory (Nass and Moon 2000; Moon 2000).

With regard to functional effectiveness and performance (H5), the hypothesis is only partially supported by the empirical results based on the duration of task completion, the

number of dialog steps required to achieve task completion, and the completion success rate. Based on the duration of task completion and the number of dialog steps required to achieve task completion, the open SDS shows a higher functional effectiveness and performance than the closed SDS. However, when referring to the completion success rate, the closed SDS performs better. The open SDS is described as more professional and useful, but the higher number of unfulfilled tasks in the open system indicates a lower level of effectiveness. To improve the completion rate in the open SDS, the propositions made in P3a–P3f should be considered when designing the error recovery strategy to avoid dialog breakdowns.

However, with 74 positive and only 3 negative comments, the duration of the dialog in the open SDS is clearly considered superior to the duration of the closed dialog, which is criticized in 47 comments and positively mentioned in only 9 statements. Among others, the closed system is criticized for the detailed prompts that cause longer dialogs and the error prompts for partly unnecessary repetitions. This result is reflected in statements such as "the long announcements are annoying," "too long instructions," or "if you know what you want, the selection is annoying." The users' negative perceptions toward the closed SDS are understandable when comparing the average duration of both dialog forms. The average duration of task completion is significantly shorter in the open SDS than in the closed SDS. The shorter prompts and the possibility to capture several variables at once through slot filling contribute to a more efficient dialog in the open SDS. This result is also indicated by the average number of dialog steps required to achieve task completion, which is significantly lower in the open SDS than in the closed SDS. The mean value of the number of user dialog steps in the open SDS is significantly higher than the minimum number. The reason is that only a few survey participants attempt to capture several variables at once to benefit from the slot filling function of the open SDS. The rationale for the non-consideration of the slot filling function can probably be found in the lack of awareness of or the lack of experience with slot filling. To increase the probability that users utilize slot filling, they should be informed in the welcome prompt of the first call about the available function, including an example statement (see also P3d).

Overall, the participants find the open system more pleasant and express a preference for its use in the future. More than two thirds (68.8%) of the 205 participants prefer the open SDS, whereas the remainder of the participants (31.2%) favor the closed SDS. The popularity of the open SDS is also reflected in the qualitative statements of the users based on the frequency analysis of the negative and positive comments (cf. Table 8). The open SDS is predominantly positively emphasized (37 positive versus 12 negative comments), whereas the closed SDS yields a rather mixed ratio with 20 positive and 21 negative comments. However, differences between age groups are observed: older users have difficulties in using the open SDS and thus clearly prefer the closed SDS, whereas younger users generally perceive the open system as more preferable. One reason for this observation has already been described when explaining the users' higher habitability with the closed SDS. Thus, older participants seem to be more familiar with using closed dialog systems and may feel overstrained with the open system. In line with prior findings in the literature on computer self-efficacy (CSE), younger users feel more comfortable using IT compared to older users (Reed et al. 2005; He and Freeman

2010). Thus, the decline of CSE in relation to age may be a further explanation for our observation.

Another explanation for the age-related differences in the preferences can be found in the research on technology acceptance. Consistently, a recent meta-analysis of 144 individual studies on the relationship between age and technology acceptance covering different types of technologies and user groups has revealed that age is indeed an antecedent of technology perceptions such as perceived ease of use, perceived usefulness, and intention to use a technology (Hauk et al. 2018). Additionally, the study has found that the negative relationship between age and technology acceptance is not present for technologies addressing the needs of the older user group. Thus, we can assume that although the acceptance toward the closed SDS is high, the acceptance toward the open SDS may be low. However, prior studies on CSE and technology acceptance are not conducted in a specific context of conversational agents; consequently, these findings may not be fully generalizable in the specific context of this type of technology. Further research is needed to shed light on the moderating effect of age on the preferred design components.

The findings indicate that the design of SDSs is a complex and demanding task; furthermore, the extent to which the design of the open dialog should integrate the elements of the closed SDS depends on the target group. Consequently, the design of an SDS should be contextualized and individualized to meet the demand of the target group of customers. For example, when being designed to serve as a customer service agent for older customers (e.g., to be used in healthcare), an SDS should integrate more structured elements. When being designed to function as a booking assistant for younger customers (e.g., a provider of adventures, as is the case with our "Adventure Guru"), an SDS should be equipped with the DPs of the open SDS. Given these findings, SDSs should have either more features of an open or a closed dialog, depending on the user group. Nevertheless, further research efforts are required to explore the impact of different SDS types on various user groups. For example, deeper insights into the impact of hybrid SDSs on the user experience of different user groups (e.g., age, gender, application domains) could provide useful results for the future design of SDSs.

With the presented design theory, we contribute to research and practice by providing a consistent set of design principles, propositions for further improvement, and future research avenues for addressing an important class of problems in human–computer interaction research. This is of particular importance in the context of customer service, as research on the design of conversational agents that can help to increase user experience is lacking to date (Gnewuch et al. 2017).

Aside from the provided design knowledge, our study shows in a particular context the dialog strategy that is preferred by users to create a user-friendly and efficient human–computer dialog. Thus, our study contributes to the body of knowledge in behavioral research by enhancing the understanding of user preferences toward different dialog strategies.

6.2 Limitations

As with any DSR project, the findings of this study are subject to some limitations that must be considered when interpreting the results. Some methodological limitations exist

with regard to the systematic literature review conducted in this study to gather relevant literature that serves as justificatory knowledge. First, the literature search is conducted in six interdisciplinary databases for a broad and comprehensive search. Despite our efforts to "accumulate a relatively complete census of relevant literature" (Webster and Watson 2002, p. 16), the identified literature is only restricted to the accessed databases and the applied set of search phrases and may not cover all relevant literature in the respective research areas. Second, although two researchers are involved in this study to achieve interrater agreement (Krippendorff 1989), the process of literature screening and assessment and the qualitative analysis of the evaluation results may be affected by selection biases (Templier and Paré 2018).

The third central limitation of our study refers to the evaluation step of our design theory, which is based on expert knowledge (Iterations 1 and 2) and perceived user experience (Iteration 3). Although the experts involved in Iterations 1 and 2 possess valuable knowledge in the application domain, their feedback, which helps refine the requirements and DPs, merely exemplifies the perceptions of these experts and thus may not be representative. Another factor that must be considered is that the user-experience survey is conducted only with German participants, a large proportion of whom are younger adults. Only relatively few participants are aged over 45 years. Hence, the sample of respondents is not demographically representative and only exemplifies a German-based point of view.

Aside from methodological issues, another limitation can be found in the explicit focus on the dialog management of task-oriented SDSs. Thus, the design theory proposed in this paper is only suitable for serving as design knowledge for task-oriented SDSs, and it cannot be generalized in the context of non-task-oriented SDS or text-based dialog systems. Among others, future research could address the extent to which the requirements and DPs for a speech-based dialog system can be adopted for the design of text-based dialog systems. The mode of communication is considered a key design characteristic of conversational agents when using natural language for human–computer communication (Knote et al. 2019; Diederich et al. 2019b). Anecdotal evidence has shown that users perceive voice-based communication with conversational agents as more natural (Novielli et al. 2010; Elshan and Ebel 2020), although the extent of this perception strongly depends on the user group (Novielli et al. 2010). Aside from these few examples, however, studies that exclusively examine the impact of different communication modes on user experience are scarce. A comparison of the requirements and DPs for both speech-based and text-based dialog systems would help to provide more generalizable design knowledge to advance research in the conversational agent research field.

Another limitation relates to the moderate influence of the application domain on the design theory. The design principles are developed based on literature from the research field of conversational agents and dialog systems, including several studies from the customer service domain. In addition, during the formative evaluation, we involve experts who have contributed their experience with dialog systems in customer service to the design of the user experience. However, given the rather moderate focus on customer service in the first design steps, the transferability and generalizability of our research results may be limited. Further studies that exclusively address domain-specific design requirements and design principles (e.g., based on user stories and user focus groups)

should complement our findings. Nevertheless, the results from our summative evaluation (cf. Section 5.2), which we conduct in the customer service domain, clearly demonstrate that the design theory is suitable for satisfying the needs of users from the customer service domain.

Another limitation is concerned with the underlying dialog theory (Bunt 2000) that serves as kernel theory for the derivation of the requirements and DPs. Although dialog theory is central to the design of SDS, it cannot cover all relevant SDS design aspects. The selection of another kernel theory may result in a modified set of requirements and DPs for the design theory. As stated in Sect. 4, several other theories may also serve as kernel theory for guiding the socio-technical design of speech dialog systems, for example, task–technology fit theory, social response theory (Nass and Moon 2000; Moon 2000), and embodied social presence theory (Mennecke et al. 2011). We rely on dialog theory as kernel theory because our focus is on the design of dialog systems that assist users with simple tasks and short dialogs, while taking into account the communicative behavior of the agents (Bunt 2000). When selecting another theory as kernel theory, the focus may be shift to other requirements and DPs. For example, according to embodied social presence theory, technologies such as SDSs are considered as social actors that should be designed as human-like as possible. Consequently, the human-like design of the system may be more important than as is the case with dialog theory.

In this context, an interesting yet still unanswered question in the SDS research field relates to the question of the specific kernel theory that is best suited to guide the design of an SDS. To date, there is a lack of research that provides an interdisciplinary overview of available and appropriate kernel theories, regardless of the respective research disciplines. Such an overview would help to guide future DSR projects for a more rigorous design process.

A further limitation of this study is that it primarily focuses on optimizing efficiency and user experience when developing the design theory, neglecting socio-economic issues. However, aspects such as data privacy, user data protection, or economic factors may have an equally significant impact on the technical design of such a class of artifacts. When using SDS, many users are concerned with the protection of their data (Luo et al. 2019). Particularly in the financial and healthcare sectors, dialog systems are met with skepticism and resistance by the end users, as the mere disclosure of confidential information poses a risk to the user (Carter and Knol 2019). In the course of the dialog, multiple user data are collected, including personal information such as name or address, customer number, credit card data or bank accounts, and these data must be adequately stored and properly handled. Given the sensitivity of such information, many users have privacy concerns (Lopatovska et al. 2020). Therefore, a concept is required to ensure data protection and secure the trustworthy handling of user data.

Furthermore, the implementation of an SDS can cause high costs. Although the costs are expected to be lower than the savings potential, they should not be underestimated (Ivanov and Webster 2017). For example, customizing the system and using conversation-based AI technologies involve development efforts and thus high personnel costs for qualified staff (Kirkpatrick 2017). In addition, due to the lack of technical knowledge, many users consider the high-value, automated services to be inferior and express an unwillingness to pay the same price despite receiving the same service (Ivanov and Webster 2017). The provision of automated services can also convey the impression that

the company is uninterested in personal customer interrelations (Knilans 2014). Aside from the design aspects, a variety of socio-economic aspects are to be considered in future studies referring to designing SDSs for customer support. As stated by other IS scholars, the design of AI-based digital assistants will be associated with both positive and negative consequences for humans, which must be further examined in research (Maedche et al. 2019).

7 Conclusion

Given the major role of dialog systems in today's customer service for answering customer requests, the design of dialog strategies constitutes an important but challenging task for designers of dialog systems. By adopting a design theory-oriented approach according to Walls et al. (1992) and Gregor and Jones (2007), we develop and evaluate a design theory for an SDS dialog strategy, including 14 requirements and five DPs. Based on the quantitative and qualitative results of a user-experience survey with 205 participants, we show that the users' experience with the proposed artifact differs depending on their age. Younger user groups tend to prefer the features of the open SDS, whereas the older user groups clearly opt for the closed SDS. Although there is still room for improvement with respect to error recovery and completion success rate, users appreciate the elements of the open variant, such as open expression, friendliness, and humanness. However, the findings show that the design of SDSs is a complex and demanding task. Nevertheless, we believe that this study contributes to research and practice by proposing a design theory that helps to improve the development dialog strategies of SDSs for enhancing the user experience in the customer service context.

8 Appendices

A.1 Literature Search Process

The literature search is conducted in the interdisciplinary databases EBSCOhost, and Google Scholar in order to gather the broadest possible literature base, ScienceDirect S, Emerald ACM. The search strings applied for the database search include a variety of synonymously used terms for SDSs ("conversational interface" OR "conversational agent" OR "voice interface" OR "voice user interface" OR "voice agent" OR "dialog system" OR "dialog interface" OR "dialog agent" OR "chatbot"). Where possible, the literature search is restricted to the title, abstractthe keywords to enable a more focused search. The search results are further limited to the years between 19902020 to capture only recent developments in the research field around the designevaluation of dialog systems. Besides, we limit the search to GermanEnglish language publications. Finally, we also conduct a forwardbackward search to identify more relevant literature on the research topic. Our initial literature search result in a total of 1,347 articles. After reading their titlesabstracts in the subsequent screeningeligibility step, we exclude 174 duplicates1,009 irrelevant publications. We then screen the full texts of the remaining 164 publications for relevance. To be selected for the literature sample, the publications must address socio-technical issues concerning the design, adoptionuse of SDSs. Moreover,

the contributions should include qualitative studies that provide literature overviews, conceptual descriptions or case studies as well as quantitative studies with empirical findings on SDSs. In order to benefit from both academicpractitioner knowledge, we include academic literature such as journal articles, conference papersworking papers as well as more practice-oriented publications such as technical reportstechnical magazines. In this way, we yield a final sample of 74 publications that are considered relevant for elaborating on the problem identificationmotivation as well as for serving as justificatory knowledge.

A.2 Search Strings Applied for the Literature Search

Database	Search in	Search strings
ACM Library	Title, Abstract, Keyword	(“conversational interface” OR “conversational agent” OR “voice interface” OR “voice user interface” OR “voice agent” OR “dialog system” OR “dialog interface” OR “dialog agent” OR “chatbot”)
EBSCOhost	Title, Abstract, Keyword	(“conversational interface” OR “conversational agent” OR “voice interface” OR “voice user interface” OR “voice agent” OR “dialog system” OR “dialog interface” OR “dialog agent” OR “chatbot”)
Emerald	Title, Abstract	(“conversational interface” OR “conversational agent” OR “voice interface” OR “voice user interface” OR “voice agent” OR “dialog system” OR “dialog interface” OR “dialog agent” OR “chatbot”)
Science direct	Title, Abstract, Keyword	(“conversational interface” OR “conversational agent” OR “voice interface” OR “voice user interface” OR “voice agent” OR “dialog system” OR “dialog interface” OR “dialog agent” OR “chatbot”)
Scopus	Title, Abstract, Keyword	(“conversational interface” OR “conversational agent” OR “voice interface” OR “voice user interface” OR “voice agent” OR “dialog system” OR “dialog interface” OR “dialog agent” OR “chatbot”)
Google Scholar	Full text	(“conversational interface” OR “conversational agent” OR “voice interface” OR “voice user interface” OR “voice agent” OR “dialog system” OR “dialog interface” OR “dialog agent” OR “chatbot”) AND “design” (“conversational interface” OR “conversational agent” OR “voice interface” OR “voice user interface” OR “voice agent” OR “dialog system” OR “dialog interface” OR “dialog agent” OR “chatbot”) AND “evaluation” (“conversational interface” OR “conversational agent” OR “voice interface” OR “voice user interface” OR “voice agent” OR “dialog system” OR “dialog interface” OR “dialog agent” OR “chatbot”) AND “frame*”

A.3 Complete Overview of the Literature Sample (n=74)

		Literature Category			DPs					
#	**Reference**	**SDS**	**CA**	**Theory**	**DP1**	**DP2**	**DP3**	**DP4**	**DP5**	**DP6**
1	(Abushawar and Atwell 2016)	●		●						
2	(Amiri et al. 2019)	●		●						
*3	(Araujo 2018)		●			●				
*4	(Begany et al. 2016)		●					●		
*5	(Bigot et al. 2013)	●								●
*6	(Boyce 2008)	●					●			
*7	(Branham and Mukkath Roy 2019)		●			●				

A.3 Complete Overview of the Literature Sample (n=74)

		Literature Category			DPs					
*8	(Bunt 2000)	●		●	●	●	●	●	●	●
9	(Burgoon et al. 2017)	●		●						
*10	(Cambre and Kulkarni 2019)	●				●				
*11	(Chen et al. 2018)		●			●				
12	(Chordas 2018)		●	●						
*13	(Chu et al. 2005)	●			●					●
*14	(Commarford et al. 2008)	●								●
*15	(Cowan et al. 2017)		●			●				
*16	(Danielescu and Christian 2018)		●			●				
*17	(Delogu et al. 1998)	●			●					
18	(Diederich et al. 2019b)		●	●						
19	(Diederich et al. 2019a)		●	●						
*20	(Diederich et al. 2020)		●			●				
21	(Doherty and Curran 2019)		●	●						
*22	(Eyssel et al. 2012)	●				●				
*23	(Galitsky 2019)		●		●					●
*24	(Gardner-Bonneau and Blanchard 2007)	●							●	●
*25	(Gnewuch et al. 2017)		●			●	●			
*26	(Go and Sundar 2019)		●			●				
*27	(Goetsu and Sakai 2019)	●						●		
*28	(Griol et al. 2017)	●						●		
*29	(Handoyo et al. 2018)		●						●	
30	(Harms et al. 2019)	●		●						
*31	(Hill et al. 2015)		●			●				
*32	(Hossain et al. 2019)		●						●	
33	(Hussain et al. 2019)		●	●						
*34	(Iio et al. 2020)	●						●		
*35	(Jha 2019)		●		●					
*36	(Jain et al. 2018)		●		●	●			●	
37	(Jurafsky 2000)	●		●						
38	(Jusoh 2018)		●	●						
39	(Kaczorowska-Spychalska 2019)		●	●						
*40	(Kiseleva et al. 2016)		●						●	
41	(Klüwer 2011)	●		●						
42	(Knote et al. 2019)		●	●						
43	(Kocaballi et al. 2019)		●	●						
44	(Koetter et al. 2019)		●	●						
45	(Lalwani et al. 2018)		●	●						
46	(Laumer et al. 2019)		●	●						
*47	(Lee et al. 2010)	●					●			
*48	(Lee and Choi 2017)		●			●				
*49	(Lewis 2016)	●			●					
*50	(Linnemann and Jucks 2018)	●			●					
*51	(Lopatovska et al. 2020)		●						●	
*52	(Luo et al. 2019)		●			●				
*53	(Maas et al. 2019a)		●						●	

A.3 Complete Overview of the Literature Sample (n=74)

		Literature Category			DPs					
*54	(Maas et al. 2019b)		●						●	
55	(Mairittha et al. 2019)	●		●						
*56	(Mané and Levin 2008)	●					●			
*57	(McTear et al. 2016)		●		●		●	●	●	
58	(Meng et al. 2003)	●		●						
59	(Merdivan et al. 2019)	●		●						
*60	(Michiels 2017)		●						●	
*61	(Opfermann and Pitsch 2017)	●						●		
*62	(Pearl 2016)		●		●				●	
*63	(Portela and Granell-Canut 2017)		●			●				
*64	(Przegalinska et al. 2019)		●		●	●				
*65	(Robertson et al. 2016)	●			●					●
66	(Savcheva and Foster 2018)	●		●						
*67	(Seeger and Heinzl 2018)		●			●				
*68	(Singh and Arora 2020)	●					●			
*69	(Skantze 2005)	●						●		
70	(Torres et al. 2019)	●		●						
71	(Traum and Larsson 2003)	●		●						
*72	(Uchida et al. 2019)	●						●		
*73	(Vaira et al. 2018)		●						●	
*74	(Verhagen et al. 2014)	●			●	●				

*= Studies guiding the development of requirements and design principles (DPs)

Others = Studies that form the theoretical background

A.4 Construct Operationalization

Construct	Items	Scale	Reference
Likability	The *[SDS variant] speech dialog system is useful. The *[SDS variant] speech dialog system is friendly. I enjoyed using the *[SDS variant] speech dialog system. It is easy to learn to use the *[SDS variant] speech dialog system. I would use this *[SDS variant] speech dialog system.	5-point-Likert	(Hone and Graham 2000)
System response accuracy	The interaction with the *[SDS variant] speech dialog system is unpredictable. The *[SDS variant] speech dialog system didn't always do what I wanted. The *[SDS variant] speech dialog system makes few errors. The interaction with the *[SDS variant] speech dialog system is efficient.	5-point-Likert	(Hone and Graham 2000)
Perceived humanness	The *[SDS variant] speech dialog system appears: Extremely inhuman-like – extremely human-like Extremely unskilled – extremely skilled Extremely unengaging – extremely engaging	5-point-Likert	(Gnewuch et al. 2017)

A.4 Construct Operationalization

Construct	Items	Scale	Reference
Habitability	I sometimes wondered if I was using the right word.	5-point-Likert	(Hone and Graham 2000)
	I always knew what to say to the *[SDS variant] speech dialog system.		
	I was not always sure what the *[SDS variant] speech dialog system was doing.		
	It is easy to lose track of where you are in the *[SDS variant] speech dialog system.		

*open vs. closed

Funding Open Access funding enabled and organized by Projekt DEAL.

References

Abbasi A, Chen H (2008) CyberGate: A Design Framework and System for Text Analysis of Computer-Mediated Communication. MIS Q 32:811–837. https://doi.org/10.2307/25148873

Abushawar B, Atwell E (2016) Usefulness, localizability, humanness, and language-benefit: additional evaluation criteria for natural language dialogue systems. Int J Speech Technol 19:373–383. https://doi.org/10.1007/s10772-015-9330-4

Amiri S, Bajracharya S, Goktolga C et al (2019) Augmenting Knowledge through Statistical, Goal-oriented Human-Robot Dialog. ArXiv190703390 Cs

Araujo T (2018) Living up to the chatbot hype: The influence of anthropomorphic design cues and communicative agency framing on conversational agent and company perceptions. Comput Hum Behav 85:183–189. https://doi.org/10.1016/j.chb.2018.03.051

Begany GM, Sa N, Yuan X (2016) Factors Affecting User Perception of a Spoken Language vs. Textual Search Interface: A Content Analysis. Interact Comput 28:170–180. https://doi.org/10.1093/iwc/iwv029

Benner D, Elshan E, Schöbel S, Janson A (2021) What do you mean? A Review on Recovery Strategies to Overcome Conversational Breakdowns of Conversational Agents. In: International Conference on Information Systems (ICIS)

Bigot LL, Caroux L, Ros C et al (2013) Investigating memory constraints on recall of options in interactive voice response system messages. Behav Inf Technol 32:106–116. https://doi.org/10.1080/0144929X.2011.563800

Boyce SJ (2008) User Interface Design for Natural Language Systems: From Research to Reality. In: Gardner-Bonneau D, Blanchard HE (eds) Human Factors and Voice Interactive Systems. Springer US, Boston, MA, pp 43–80

Branham SM, Mukkath Roy AR (2019) Reading Between the Guidelines: How Commercial Voice Assistant Guidelines Hinder Accessibility for Blind Users. In: The 21st International ACM SIGACCESS Conference on Computers and Accessibility. Association for Computing Machinery, New York, NY, USA, pp 446–458

Bunt HC (2000) Dynamic Interpretation and Dialogue Theory. Struct Multimodal Dialogue 2:139–188

Burgoon JK, Magnenat-Thalmann N, Pantic M, Vinciarelli A (2017) Social Signal Processing. Cambridge University Press
Cambre J, Kulkarni C (2019) One Voice Fits All? Social Implications and Research Challenges of Designing Voices for Smart Devices. Proc ACM Hum-Comput Interact 3:223:1–223. https://doi.org/10.1145/3359325
Carter E, Knol C (2019) Chatbots — an organisation's friend or foe? Res Hosp Manag 9:113–115
Carter RA, Anton AI, Dagnino A, Williams L (2001) Evolving beyond requirements creep: a risk-based evolutionary prototyping model. In: Proceedings Fifth IEEE International Symposium on Requirements Engineering. pp 94–101
Chen C-Y, Yu D, Wen W et al (2018) Gunrock: Building a human-like social bot by leveraging large scale real user data. Alexa Prize Proc
Cho E, Molina MD, Wang J (2019) The Effects of Modality, Device, and Task Differences on Perceived Human Likeness of Voice-Activated Virtual Assistants. Cyberpsychology Behav Soc Netw 22:515–520. https://doi.org/10.1089/cyber.2018.0571
Chordas L (2018) Chatting It Up: Chatbots are making their way into insurance, but they won't replace the need for humans. Best's Rev 119:88
Chu S-W, O'Neill I, Hanna P, McTear MF (2005) An approach to Multi-Strategy Dialogue Management. Unknown Host Publication. International Society for Computers and Their Applications, pp 865–868
Commarford PM, Lewis JR, Smither JA-A, Gentzler MD (2008) A Comparison of Broad Versus Deep Auditory Menu Structures. Hum Factors 50:77–89. https://doi.org/10.1518/001872008X250665
Cowan BR, Pantidi N, Coyle D et al (2017) "What can i help you with?": infrequent users' experiences of intelligent personal assistants. In: Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services. Association for Computing Machinery, New York, NY, USA, pp 1–12
Dale R (2016) The return of the chatbots. Nat Lang Eng 22:811–817. https://doi.org/10.1017/S1351324916000243
Danielescu A, Christian G (2018) A Bot is Not a Polyglot: Designing Personalities for Multi-Lingual Conversational Agents. In: Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems. Association for Computing Machinery, New York, NY, USA, pp 1–9
Davis AM (1992) Operational prototyping: a new development approach. IEEE Softw 9:70–78. https://doi.org/10.1109/52.156899
Delogu C, Carlo AD, Rotundi P, Sartori D (1998) A comparison between DTMF and ASR IVR services through objective and subjective evaluation. In: Proceedings 1998 IEEE 4th Workshop Interactive Voice Technology for Telecommunications Applications. IVTTA '98 (Cat. No.98TH8376). pp 145–150
Deloitte (2019) Conversational AI
Diederich S, Brendel A, Kolbe L (2019a) On Conversational Agents in Information Systems Research: Analyzing the Past to Guide Future Work. Wirtsch 2019 Proc
Diederich S, Brendel A, Kolbe L (2019b) Towards a Taxonomy of Platforms for Conversational Agent Design. Wirtsch 2019 Proc
Diederich S, Brendel A, Morana S, Kolbe L (2022) On the Design of and Interaction with Conversational Agents: An Organizing and Assessing Review of Human-Computer Interaction Research. J Assoc Inf Syst 23:96–138. https://doi.org/10.17705/1jais.00724
Diederich S, Brendel AB, Kolbe LM (2020) Designing Anthropomorphic Enterprise Conversational Agents. Bus Inf Syst Eng Int J Wirtsch 62:193–209
Doherty D, Curran K (2019) Chatbots for online banking services. Web Intell 17:327–342. https://doi.org/10.3233/WEB-190422
Elshan E, Ebel P (2020) Let's Team Up: Designing Conversational Agents as Teammates. In: International Conference on Information Systems (ICIS)
Eyssel F, de Ruiter L, Kuchenbrandt D et al (2012) 'If you sound like me, you must be more human': On the interplay of robot and user features on human-robot acceptance and anthropomorphism. In: 2012 7th ACM/IEEE International Conference on Human-Robot Interaction (HRI). pp 125–126
Field A (2009) Discovering Statistics Using SPSS, 3. Edition. Sage Publications, Los Angeles
Firdaus M, Golchha H, Ekbal A, Bhattacharyya P (2021) A Deep Multi-task Model for Dialogue Act Classification, Intent Detection and Slot Filling. Cogn Comput 13:626–645. https://doi.org/10.1007/s12559-020-09718-4
Forrester (2017) humans vs. machines: how to stop your virtual agent from lagging behind (Technical Report)
Galitsky B (2019) Chatbot Components and Architectures. In: Galitsky B (ed) Developing Enterprise Chatbots: Learning Linguistic Structures. Springer International Publishing, Cham, pp 13–51
Gardner-Bonneau D, Blanchard HE (2007) Human Factors and Voice Interactive Systems. Springer Science & Business Media

Gnewuch U, Morana S, Mädche A (2017) Towards Designing Cooperative and Social Conversational Agents for Customer Service. ICIS 2017 Proc

Go E, Sundar SS (2019) Humanizing chatbots: The effects of visual, identity and conversational cues on humanness perceptions. Comput Hum Behav 97:304–316. https://doi.org/10.1016/j.chb.2019.01.020

Goetsu S, Sakai T (2019) Voice Input Interface Failures and Frustration: Developer and User Perspectives. In: The Adjunct Publication of the 32nd Annual ACM Symposium on User Interface Software and Technology. Association for Computing Machinery, New York, NY, USA, pp 24–26

Goodhue DL (1995) Understanding User Evaluations of Information Systems. Manag Sci 41:1827–1844. https://doi.org/10.1287/mnsc.41.12.1827

Gregor S, Hevner AR (2013) Positioning and Presenting Design Science Research for Maximum Impact. MIS Q 37:337–A6

Gregor S, Jones D (2007) The Anatomy of a Design Theory. J Assoc Inf Syst 8:313–335

Gregor S, Kruse LC, Seidel S (2020) Research Perspectives: The Anatomy of a Design Principle. J Assoc Inf Syst 21. https://doi.org/10.17705/1jais.00649

Griol D, de Miguel AS, Molina JM (2017) FRB-Dialog: A Toolkit for Automatic Learning of Fuzzy-Rule Based (FRB) Dialog Managers. In: de Martínez FJ, Urraca R, Quintián H, Corchado E (eds) Hybrid Artificial Intelligent Systems. Springer International Publishing, Cham, pp 306–317

Gupta B (2021) What are Voice Bots? Difference between Chatbots and Voice Bots. In: BotPenguin. https://botpenguin.com/what-are-voice-bots-difference-between-chatbots-and-voice-bots/. Accessed 19 Oct 2021

Handoyo E, Arfan M, Soetrisno YAA et al (2018) Ticketing Chatbot Service using Serverless NLP Technology. In: 2018 5th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE). pp 325–330

Harms J, Kucherbaev P, Bozzon A, Houben G (2019) Approaches for Dialog Management in Conversational Agents. IEEE Internet Comput 23:13–22. https://doi.org/10.1109/MIC.2018.2881519

Hauk N, Hüffmeier J, Krumm S (2018) Ready to be a Silver Surfer? A Meta-analysis on the Relationship Between Chronological Age and Technology Acceptance. Comput Hum Behav 84:304–319. https://doi.org/10.1016/j.chb.2018.01.020

He J, Freeman L(2010) Understanding the Formation of General Computer Self-Efficacy.Commun Assoc Inf Syst26

Hevner AR, March ST, Park J, Ram S (2004) Design Science in Information Systems Research. MIS Q 28:75–105

Hill J, Randolph Ford W, Farreras IG (2015) Real conversations with artificial intelligence: A comparison between human–human online conversations and human–chatbot conversations. Comput Hum Behav 49:245–250. https://doi.org/10.1016/j.chb.2015.02.026

Hone KS, Graham R (2000) Towards a tool for the subjective assessment of speech system interfaces. SASSI)

Hossain MS, Zhou X, Rahman MF (2019) Customer satisfaction under heterogeneous services of different self-service technologies. Manag Mark Chall Knowl Soc 14:90–107. https://doi.org/10.2478/mmcks-2019-0007

Hudson S, González-Gómez HV, Rychalski A (2017) Call centers: is there an upside to the dissatisfied customer experience? J Bus Strategy 38:39–46. https://doi.org/10.1108/JBS-01-2016-0008

Hussain S, Ameri Sianaki O, Ababneh N (2019) A Survey on Conversational Agents/Chatbots Classification and Design Techniques. In: Barolli L, Takizawa M, Xhafa F, Enokido T (eds) Web, Artificial Intelligence and Network Applications. Springer International Publishing, Cham, pp 946–956

Iio T, Yoshikawa Y, Chiba M et al (2020) Twin-Robot Dialogue System with Robustness against Speech Recognition Failure in Human-Robot Dialogue with Elderly People. Appl Sci 10:1522. https://doi.org/10.3390/app10041522

Ivanov SH, Webster C (2017) Adoption of Robots, Artificial Intelligence and Service Automation by Travel, Tourism and Hospitality Companies – A Cost-Benefit Analysis. Social Science Research Network, Rochester, NY

Jain M, Kumar P, Kota R, Patel SN (2018) Evaluating and Informing the Design of Chatbots. In: Proceedings of the 2018 Designing Interactive Systems Conference. Association for Computing Machinery, New York, NY, USA, pp 895–906

Jha AK(2019) Journey to the Realm of Chatbots.Int J Res Eng Sci Manag2

Jurafsky D (2000) Speech & language processing. Pearson Education India

Jusoh S (2018) Intelligent Conversational Agent for Online Sales. In: 2018 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI). pp 1–4

Kaczorowska-Spychalska D (2019) How chatbots influence marketing. Management 23:251–270. https://doi.org/10.2478/manment-2019-0015

Kirkpatrick K (2017) AI in contact centers. Commun ACM 60:18–19. https://doi.org/10.1145/3105442
Kiseleva J, Williams K, Jiang J et al (2016) Understanding User Satisfaction with Intelligent Assistants. In: Proceedings of the 2016 ACM on Conference on Human Information Interaction and Retrieval. Association for Computing Machinery, New York, NY, USA, pp 121–130
Klüwer T (2011) From chatbots to dialog systems. In: Conversational agents and natural language interaction: Techniques and Effective Practices. IGI Global, pp 1–22
Knilans G (2014) First impressions make lasting impressions. Employ Relat Today 41:39–45
Knote R, Janson A, Söllner M, Leimeister JM (2019) Classifying smart personal assistants: an empirical cluster analysis. In: Proceedings of the 52nd Hawaii international conference on system sciences
Kocaballi AB, Laranjo L, Coiera E (2019) Understanding and Measuring User Experience in Conversational Interfaces. Interact Comput 31:192–207. https://doi.org/10.1093/iwc/iwz015
Koetter F, Blohm M, Drawehn J et al (2019) Conversational Agents for Insurance Companies: From Theory to Practice. In: van den Herik J, Rocha AP, Steels L (eds) Agents and Artificial Intelligence. Springer International Publishing, Cham, pp 338–362
Krippendorff K (1989) In: Barnouw E, Gerbner G, Schramm W, Worth TL, Gross L (eds) Content Analysis. OXford University Press, New York, NY
Kruskal WH, Wallis WA (1952) Use of Ranks in One-Criterion Variance Analysis. J Am Stat Assoc 47:583–621. https://doi.org/10.1080/01621459.1952.10483441
Kvale K, Freddi E, Hodnebrog S et al (2021) Understanding the User Experience of Customer Service Chatbots: What Can We Learn from Customer Satisfaction Surveys? In: Følstad A, Araujo T, Papadopoulos S et al (eds) Chatbot Research and Design. Springer International Publishing, Cham, pp 205–218
Lalwani T, Bhalotia S, Pal A et al(2018) Implementation of a Chat Bot System using AI and NLP.Int J Innov Res Comput Sci Technol-IJIRCST6
Laumer S, Gubler F, Racheva A, Maier C (2019) Use Cases for Conversational Agents: An Interview-based Study. In: AMCIS 2019 Proceedings
Lee C, Jung S, Kim K et al (2010) Recent approaches to dialog management for spoken dialog systems. J Comput Sci Eng 4:1–22
Lee M, Schlogl S, Montenegro S et al (2017) First time encounters with Roberta: a humanoid assistant for conversational autobiography creation. In: Proceedings of the 12th Summer Workshop on Multimodal Interfaces (eNTERFACE'16). pp 30–38
Lee S, Choi J (2017) Enhancing user experience with conversational agent for movie recommendation: Effects of self-disclosure and reciprocity. Int J Hum-Comput Stud 103:95–105. https://doi.org/10.1016/j.ijhcs.2017.02.005
Lewis JR (2016) Practical Speech User Interface Design. CRC Press
Linnemann GA, Jucks R (2018) 'Can I Trust the Spoken Dialogue System Because It Uses the Same Words as I Do?'—Influence of Lexically Aligned Spoken Dialogue Systems on Trustworthiness and User Satisfaction. Interact Comput 30:173–186. https://doi.org/10.1093/iwc/iwy005
Lopatovska I, Griffin AL, Gallagher K et al (2020) User recommendations for intelligent personal assistants. J Librariansh Inf Sci 52:577–591. https://doi.org/10.1177/0961000619841107
Luger E, Sellen A (2016) "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 Systems. Association for Computing Machinery, New York, NY, USA, pp 5286–5297
Luo X, Tong S, Fang Z, Qu Z (2019) Frontiers: Machines vs. Humans: The Impact of Artificial Intelligence Chatbot Disclosure on Customer Purchases. Mark Sci 38:937–947. https://doi.org/10.1287/mksc.2019.1192
Maas P, Meichtry TM, Steiner PH(2019a) Erfolgspotenziale von Conversational Agents - am Beispiel der Assekuranz.Mark Rev St Gallen22–29
Maas P, Meichtry TM, Steiner PH(2019b) Conversational Agents aus Kundenperspektive.Mark Rev St Gallen Mark Für Theor Prax86–94
Maedche A, Legner C, Benlian A et al (2019) AI-Based Digital Assistants. Bus Inf Syst Eng 61:535–544. https://doi.org/10.1007/s12599-019-00600-8
Mairittha T, Mairittha N, Inoue S (2019) Evaluating a Spoken Dialogue System for Recording Systems of Nursing Care. Sensors 19:3736. https://doi.org/10.3390/s19173736
Mané AM, Levin E (2008) Designing the Voice User Interface for Automated Directory Assistance. In: Gardner-Bonneau D, Blanchard HE (eds) Human Factors and Voice Interactive Systems. Springer US, Boston, MA, pp 117–134
Markus ML, Majchrzak A, Gasser L (2002) A Design Theory for Systems That Support Emergent Knowledge Processes. MIS Q 26:179–212

Mayring P(2001) Combination and Integration of Qualitative and Quantitative Analysis.Forum Qual Sozialforschung Forum Qual Soc Res2
McTear MF (2017) The Rise of the Conversational Interface: A New Kid on the Block? In: Quesada JF, Martín Mateos F-J, López Soto T (eds) Future and Emerging Trends in Language Technology. Machine Learning and Big Data. Springer International Publishing, Cham, pp 38–49
McTear MF (2002) Spoken dialogue technology: enabling the conversational user interface. ACM Comput Surv 34:90–169. https://doi.org/10.1145/505282.505285
McTear MF, Callejas Z, Griol D (2016) The conversational interface. Springer
Meng HM, Wai C, Pieraccini R (2003) The use of belief networks for mixed-initiative dialog modeling. IEEE Trans Speech Audio Process 11:757–773. https://doi.org/10.1109/TSA.2003.814380
Mennecke BE, Triplett JL, Hassall LM et al (2011) An Examination of a Theory of Embodied Social Presence in Virtual Worlds*. Decis Sci 42:413–450. https://doi.org/10.1111/j.1540-5915.2011.00317.x
Merdivan E, Singh D, Hanke S, Holzinger A (2019) Dialogue Systems for Intelligent Human Computer Interactions. Electron Notes Theor Comput Sci 343:57–71. https://doi.org/10.1016/j.entcs.2019.04.010
Meth H, Mueller B, Maedche A (2015) Designing a Requirement Mining System. J Assoc Inf Syst 16. https://doi.org/10.17705/1jais.00408
Michiels E (2017) Modelling Chatbots with a Cognitive System Allows for a Differentiating User Experience. In: PoEM Doctoral Consortium. pp 70–78
Miller GA (1956) The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychol Rev 63:81
Moon Y (2000) Intimate Exchanges: Using Computers to Elicit Self-Disclosure from Consumers. J Consum Res 26:323–339. https://doi.org/10.1086/209566
Murdock BBJr (1962) The serial position effect of free recall. J Exp Psychol 64:482–488. https://doi.org/10.1037/h0045106
Nass C, Moon Y (2000) Machines and Mindlessness: Social Responses to Computers. J Soc Issues 56:81–103. https://doi.org/10.1111/0022-4537.00153
Novielli N, de Rosis F, Mazzotta I (2010) User attitude towards an embodied conversational agent: Effects of the interaction mode. J Pragmat 42:2385–2397
Nunnally JC, Bernstein IH (1994) Psychometric theory. McGraw-Hill, New York
Opfermann C, Pitsch K (2017) Reprompts as error handling strategy in human-agent-dialog? User responses to a system's display of non-understanding. In: 2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN). pp 310–316
Pearl C (2016) Designing voice user interfaces: principles of conversational experiences. O'Reilly Media, Inc
Peffers K, Tuunanen T, Rothenberger MA, Chatterjee S (2007) A Design Science Research Methodology for Information Systems Research. J Manag Inf Syst 24:45–77. https://doi.org/10.2753/MIS0742-1222240302
Pfeuffer N, Benlian A, Gimpel H, Hinz O (2019) Anthropomorphic Information Systems. Bus Inf Syst Eng 61:523–533. https://doi.org/10.1007/s12599-019-00599-y
Portela M, Granell-Canut C (2017) A new friend in our smartphone? observing interactions with chatbots in the search of emotional engagement. In: Proceedings of the XVIII International Conference on Human Computer Interaction. Association for Computing Machinery, New York, NY, USA, pp 1–7
Pries-Heje J, Baskerville R, Venable JR (2008) Strategies for design science research evaluation
Przegalinska A, Ciechanowski L, Stroz A et al (2019) In bot we trust: A new methodology of chatbot performance measures. Bus Horiz 62:785–797. https://doi.org/10.1016/j.bushor.2019.08.005
Reed K, Doty DH, May DR (2005) The Impact of Aging on Self-efficacy and Computer Skill Acquisition. J Manag Issues 17:212–228
Rieman J, Franzke M, Redmiles D (1995) Usability evaluation with the cognitive walkthrough. In: Conference companion on Human factors in computing systems. pp 387–388
Robertson N, McDonald H, Leckie C, McQuilken L (2016) Examining customer evaluations across different self-service technologies. J Serv Mark 30:88–102. https://doi.org/10.1108/JSM-07-2014-0263
Ruan S, Wobbrock JO, Liou K et al (2018) Comparing Speech and Keyboard Text Entry for Short Messages in Two Languages on Touchscreen Phones. Proc ACM Interact Mob Wearable Ubiquitous Technol 1(159):1–159. https://doi.org/10.1145/3161187
Rzepka C, Berger B, Hess T (2020) Is it a Match? Examining the Fit Between Conversational Interaction Modalities and Task Characteristics. In: ICIS
Savcheva D, Foster ME (2018) Comparing User Responses to Limited and Flexible Interaction in a Conversational Interface. In: Proceedings of the 6th International Conference on Human-Agent Interaction. Association for Computing Machinery, New York, NY, USA, pp 368–370

Schmitt A, Zierau N, Janson A, Leimeister JM (2021) Voice as a Contemporary Frontier of Interaction Design. ECIS 2021 Res Pap

Schoormann T, Stadtländer M, Knackstedt R (2021) Designing business model development tools for sustainability—a design science study. Electron Mark. https://doi.org/10.1007/s12525-021-00466-3

Seeger A-M, Heinzl A (2018) Human Versus Machine: Contingency Factors of Anthropomorphism as a Trust-Inducing Design Strategy for Conversational Agents. In: Davis FD, Riedl R, vom Brocke J et al (eds) Information Systems and Neuroscience. Springer International Publishing, Cham, pp 129–139

Singh S, Arora S (2020) Dialogue System in context with Natural Language Processing. Stud Indian Place Names 40:1376–1381

Skantze G (2005) Exploring human error recovery strategies: Implications for spoken dialogue systems. Speech Commun 45:325–341

Straub D, Boudreau M-C, Gefen D (2004) Validation Guidelines for IS Positivist Research. Commun Assoc Inf Syst 13. https://doi.org/10.17705/1CAIS.01324

Templier M, Paré G (2018) Transparency in literature reviews: an assessment of reporting practices across review types and genres in top IS journals. Eur J Inf Syst 27:503–550. https://doi.org/10.1080/0960085X.2017.1398880

Torres MI, Olaso JM, Glackin N et al (2019) A Spoken Dialogue System for the EMPATHIC Virtual Coach. In: D'Haro LF, Banchs RE, Li H (eds) 9th International Workshop on Spoken Dialogue System Technology. Springer, Singapore, pp 259–265

Traum DR, Larsson S (2003) The Information State Approach to Dialogue Management. In: van Kuppevelt J, Smith RW (eds) Current and New Directions in Discourse and Dialogue. Springer Netherlands, Dordrecht, pp 325–353

Uchida T, Minato T, Koyama T, Ishiguro H (2019) Who Is Responsible for a Dialogue Breakdown? An Error Recovery Strategy That Promotes Cooperative Intentions From Humans by Mutual Attribution of Responsibility in Human-Robot Dialogues. Front Robot AI 6. https://doi.org/10.3389/frobt.2019.00029

Vaira L, Bochicchio MA, Conte M et al (2018) MamaBot: a System based on ML and NLP for supporting Women and Families during Pregnancy. In: Proceedings of the 22nd International Database Engineering & Applications Symposium. Association for Computing Machinery, New York, NY, USA, pp 273–277

Venable J (2006) A framework for Design Science research activities. In: Emerging Trends and Challenges in Information Technology Management: Proceedings of the 2006 Information Resource Management Association Conference. Idea Group Publishing, pp 184–187

Venable J, Pries-Heje J, Baskerville R (2016) FEDS: a Framework for Evaluation in Design Science Research. Eur J Inf Syst 25:77–89. https://doi.org/10.1057/ejis.2014.36

Venable J, Pries-Heje J, Baskerville R (2012) A Comprehensive Framework for Evaluation in Design Science Research. In: Peffers K, Rothenberger M, Kuechler B (eds) Design Science Research in Information Systems. Advances in Theory and Practice. Springer, Berlin, Heidelberg, pp 423–438

Verhagen T, van Nes J, Feldberg F, van Dolen W (2014) Virtual Customer Service Agents: Using Social Presence and Personalization to Shape Online Service Encounters. J Comput-Mediat Commun 19:529–545. https://doi.org/10.1111/jcc4.12066

vom Brocke J, Simons A, Niehaves B et al (2009) Reconstructing the Giant: On the Importance of Rigour in Documenting the Literature Search Process. ECIS 2009 Proc

Walls JG, Widmeyer GR, El Sawy OA (1992) Building an Information System Design Theory for Vigilant EIS. Inf Syst Res 3:36–59. https://doi.org/10.1287/isre.3.1.36

Walsh J, Andersen BL, Katz JE, Groshek J (2018) Personal Power and Agency When Dealing with Interactive Voice Response Systems and Alternative Modalities. Media Commun 6:60–68. https://doi.org/10.17645/mac.v6i3.1205

Webster J, Watson RT (2002) Analyzing the Past to Prepare for the Future: Writing a Literature Review. MIS Q 26:xiii–xxiii

Wilcoxon F (1992) Individual Comparisons by Ranking Methods. In: Kotz S, Johnson NL (eds) Breakthroughs in Statistics: Methodology and Distribution. Springer, New York, NY, pp 196–202

zendesk (2019) The Zendesk Customer Experience Trends Report 2019

Zhao YJ, Li YL, Lin M (2019) A Review of the Research on Dialogue Management of Task-Oriented Systems. J Phys Conf Ser 1267:012025. https://doi.org/10.1088/1742-6596/1267/1/012025