

Please quote as: Elshan, E., Ebel, P. A., Söllner, M.; Leimeister, J. M. (2023).
Leveraging Low Code Development of Smart Personal Assistants: An Integrated
Design Approach with the SPADE Method. *Journal of Management Information
Systems (JMIS)*, 40, 96-129.

Leveraging Low Code Development of Smart Personal Assistants: An Integrated Design Approach with the SPADE Method

Edona Elshan, Philipp Ebel, Matthias Söllner & Jan Marco Leimeister

To cite this article: Edona Elshan, Philipp Ebel, Matthias Söllner & Jan Marco Leimeister (2023) Leveraging Low Code Development of Smart Personal Assistants: An Integrated Design Approach with the SPADE Method, Journal of Management Information Systems, 40:1, 96-129, DOI: [10.1080/07421222.2023.2172776](https://doi.org/10.1080/07421222.2023.2172776)

To link to this article: <https://doi.org/10.1080/07421222.2023.2172776>



© 2023 The Author(s). Published with license by Taylor & Francis Group, LLC.



View supplementary material [↗](#)



Published online: 30 Mar 2023.



Submit your article to this journal [↗](#)



Article views: 635



View related articles [↗](#)







View Crossmark data [↗](#)



OPEN ACCESS



Leveraging Low Code Development of Smart Personal Assistants: An Integrated Design Approach with the SPADE Method

Edona Elshan ^a, Philipp Ebel ^a, Matthias Söllner ^b, and Jan Marco Leimeister ^{a,b}

^aInstitute of Information Management, University of St. Gallen, St. Gallen, Switzerland; ^bResearch Center for IS Design, Information Systems and Systems Engineering (ITeG), University of Kassel, Kassel, Germany

ABSTRACT

Smart personal assistants (SPAs), such as Alexa for example, promise individualized user interactions owing to their varying interaction possibilities, knowledgeability, and human-like behaviors. To support the widespread adoption and use of SPAs, organizations such as Google or Amazon provide low code environments that support the development of SPAs (e.g., for Google Home or Amazon's Alexa). These so-called low code platforms enable domain experts (e.g., business users without programming skills or experience) to develop SPAs for their purposes. However, using these platforms alone does not guarantee a useful and good conversation with novel SPAs due to non-intuitive design choices. Following a design science research approach, we propose the Smart Personal Assistant for Domain Experts (SPADE) method to address the missing link. This method supports domain experts in the development and contextualization of sophisticated SPAs for various application scenarios and focuses especially on conversational and anthropomorphic design steps. Our proof of concept and proof of value results show that SPADE is useful for supporting domain experts to create effective SPAs in different domains beyond private set-ups.

KEYWORDS

Anthropomorphic design; conversational agents; conversational design; IS development; low code development platform; smart personal assistant; citizen development; end-user programming

Introduction

New technological developments, such as advances in natural language understanding and processing, have given rise to a relatively new class of systems called smart personal assistants (SPAs). SPAs are software agents designed to aid users in performing various activities by interacting with them via natural language [53, 63]. SPAs' key characteristics include their ability to react to user utterances in real time and adapt their answers accordingly, thereby engaging in dialog comparable to human-human interaction. They can also absorb contextual factors, such as a user's workflows, current knowledge state, and dialog routines, to adapt their responses accordingly. SPAs can increase the quality of various private and work-related task processes and outcomes [108]. In work environments, they are mostly expected to increase workers' productivity by helping them conduct their tasks and work routines more efficiently [52]. In education, research has shown that SPAs

CONTACT Edona Elshan  edona.elshan@unisg.ch  University of St. Gallen, Institute of Information Management, Müller-Friedbergstrasse 6/8, St. Gallen CH-9000, Switzerland

Supplemental data for this article can be accessed at www.tandfonline.com/mmis.

 Supplemental data for this article can be accessed online at <https://doi.org/10.1080/07421222.2023.2172776>.

© 2023 The Author(s). Published with license by Taylor & Francis Group, LLC.

This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

have the potential to increase learners' success in different contexts. In private contexts, they are often designed to enhance users' well-being and comfort [121].

SPAs have become increasingly popular in both the public (e.g., at work) and private domains. Recent studies have shown that 4.2 billion SPAs are frequently used in households worldwide [119]. Consequently, several prominent technology providers (e.g., Amazon, Google, IBM, and Facebook) are offering SPAs and technological platforms in a bid to conquer this rapidly growing marketplace [128].

The development of a SPA is challenging for designers because to the high degree of contextualization (such as context adaptive conversations, e.g., individual coaching) required to deliver flawless user experiences [26, 107]. During the development process, they must account for the user's goals, the user's preferences with respect to the assistant's personality and speech style, and prepare a script that encompasses all possible permutations of user/SPA interactions. Designers must also anticipate the user's actions and design appropriate SPA responses. In sum, these design choices require the developer to perform various activities (i.e., writing a script and implementing it via natural language processing techniques, determining the SPA's personality, and implementing the underlying technological infrastructure) and train the knowledge base for the content of the conversation. As a result of these SPA-specific design challenges (i.e., contextualization), conventional approaches to interaction design often fall short, since design iterations based on prototyping and user testing are difficult to realize for domain experts [69].

Consequently, many SPA technology providers have developed toolkits that support the process of designing and developing SPAs without requiring extensive programming knowledge nor experience [20]. This shall enable domain experts (e.g., business representatives such as product managers) to develop useful SPAs independently, thereby reducing the need for extensive user research and testing. Examples of this can be found at Google, Amazon, and IBM, which offer low code development platforms (LCDPs) that allow users to create their SPAs without much technological know-how and time. Within companies, these LCDPs are part of a larger trend of technology democratization [16], referring to any undertaking that traditionally required coding but can now be accomplished by a business user. Domain experts (irrespective of the domain they work in, be it education, marketing, accounting, design, or operations) thus have greater control over the development tasks and workflows. Consequently, no domain expert ends up in a back-and-forth with developers and faster higher-quality outputs are promised [8].

In reality, however, these LCDPs provide only the necessary technological functionalities while fading out certain socio-technical factors, such as anthropomorphism, necessary to create an effective experience with SPAs. The lack of guidance during the development, becomes even more obvious when having a closer look at the number of abandoned SPAs and the number of SPAs introduced to the market with little to no user acceptance (e.g., growth rate for Amazon Alexa Skills dropped over the past years, suggesting lower developer enthusiasm [59]). At present, aside from SPAs' omnipresence, interactions with them frequently result in questions like "I did not catch that, would you repeat it?" and abrupt ends to conversations. Nonetheless, SPAs can sustain themed discussions nearly 85 percent of the time [95]. Considering that social or commercial encounters rely on user engagement and considerable experience, the remaining 15 percent of interactions with the SPA are crucial: if the SPA is unable

to smoothly handle the users' conversation, breakdowns may lead the user to abandon the service [3]. In addition to general natural language or linguistic errors, interaction and conversation errors as well as motivational failures can result in conversational breakdown and, thus, failed interactions [6, 12]. These potential errors can lead to total rejection of and loss of trust in SPAs and their respective providers [33]. As many service encounters rely on SPAs, these failures have serious consequences. Poor interaction with an SPA may influence a customer's buying decisions [114] and can affect word of mouth promotion and repeat purchase intention [10].

To fill this gap, in this paper, we seek to answer the following research question: *How can we support domain experts' development of useful smart personal assistants within low code development environments?*

To address this question and to unite the scientific research, and expertise from previous SPA development projects, we followed a design science research (DSR) approach to develop and evaluate the comprehensive and generalized Smart Personal Assistant for Domain Experts (SPADE) method [92]. Due to the lack of knowledge of supporting domain experts in designing useful SPAs, we follow a theory-motivated design approach [14, 73]. Thereby, we combined theoretical insights from SPAs, user-centered design, and end user programming research with practical insights from user interviews to identify the requirements that our solution needed to meet. Second, using these requirements, we developed SPADE, a method that guides domain experts through the SPA development process step by step and provides information on the major activities in each step and how to complete them. To demonstrate SPADE's feasibility and value, we followed Nunamaker et al. [89, 90] in providing proof of concept and proof of value evaluation. In our proof of concept, we conducted 39 interviews with domain experts with experience of developing SPAs in a low code environment. Within the proof of value, in the first step, we used two in-depth studies to demonstrate the application of the method for two different kinds of channels (i.e., voice- and text-based assistants). In these in-depth studies, we demonstrated that end-users (i.e., clients or employees) consider the resulting SPAs useful for their intended purpose. Second, we used a study of 38 domain experts in the field of education (i.e., educators without programming knowledge or experience) to demonstrate our method's usefulness in guiding domain experts through the SPA development process. We thus generate design knowledge to literature and practice, that helps domain experts explore and leverage SPAs' potentials in their areas of expertise.

Theoretical Background

Smart Personal Assistants

Smart Personal Assistants are software agents designed to aid users in performing various activities by interacting with them via natural language [40, 70]. SPAs' novelty lies in two aspects: how users interact with them and their human-like behavior. To analyze these new modes of interaction, the dimensions "degree of intelligence of the system" and "degree of interaction implemented in the system" were identified in Knotte et al.'s framework [62, 63]. The dimension "degree of interaction" includes the factors of communication mode, direction of explicit interaction, query input, response output, and action. "Degree of intelligence" includes the factors of assistance domain, accepted commands, adaptivity,

collective intelligence, and embodiment. The main functionality, the SPA's "brain," is typically housed as a cloud service that uses machine learning and natural language processing techniques to handle voice data (converting voice-to-text, performing linguistic context analysis, and providing answers to questions [20]). In sum, SPAs can be considered a special type of software that is known as a dialog-based system. Thus, SPAs are distinguished from other forms of user interfaces by their use of natural language interfaces (e.g., graphical or code-based user interfaces) [2]. Unlike typical user interfaces (point-and-click interfaces), which are mostly concerned with visual design, SPA design also takes into account interactions with synthetic entities through natural language. Thereby, compared to other types of software, SPA development also entails designing conversations rather than merely designing interfaces and trains computer programs instead of simply programming them [80]. Within this article we will primarily focus on the first distinguishment, the shift from designing interfaces to designing conversations.

Despite modern SPAs' capabilities (such as understanding and communicating through natural language), successful user experience design still represents a major challenge, particularly due to high user expectations [76, 112]. SPA interaction experiences may or may not foster user satisfaction [93, 133]. An experience with a SPA is positive if the user has sensations or acquires knowledge through interaction with different elements of a context that has been purposefully designed by a provider [48, 131]. For instance, a therapeutically successful conversation in healthcare is dependent on reflective listening or a reflection of emotions back to the patient to demonstrate understanding, resulting in connection and involvement [67]. In the context of e-commerce, for example, insufficient engagement during transactions leads to a high percentage of order cancelations and frequently results in customer dissatisfaction [61, 115]. These examples highlight the importance of good conversation design. Within this realm, research has lately focused on the recovery of conversational breakdowns [6].

In order to facilitate good conversation design, research has also focused on anthropomorphic SPA design, which promotes a more human-like SPA appearance and behavior (e.g., [2, 19, 27, 35]). Such anthropomorphic design elements are likely to increase user anthropomorphism [113]. In this context, anthropomorphism is the psychological phenomenon of attributing human-like characteristics to nonhuman agents, such as motivations, feelings, and intentions [34]. Two types of anthropomorphism are commonly studied. The unconscious attribution of human features to nonhuman interaction partners is known as mindless anthropomorphism [110], and it is the subject of the psychological theory of anthropomorphism [34]. Mindful anthropomorphism, by contrast, is the intentional belief that a technological interaction partner is human rather than a computer [56]. Correspondingly, the Computers are Social Actors (CASA) paradigm proposes that humans exhibit the same social responses to computer systems as they do to other humans [85, 86], and several empirical studies based on this paradigm have shown that anthropomorphic design influences human perception and behavior. For example, studies have shown that anthropomorphism can increase user trust (e.g., [19]) and stimulate favorable attitudes, such as liking, empathy, and enjoyment (e.g., [9]).

These design considerations may help to alleviate the challenges associated with SPAs. To ensure a sufficient level of anthropomorphism, designers must consider the user's preferences regarding the assistant's personality and speech style and to prepare

a corresponding script for the interaction with the SPA [32]. The user's actions must also be anticipated, and appropriate reactions on the part of the SPA must be developed. This leads to situations in which the designer must perform various activities and in which conventional approaches to interaction design often fall short [69].

Consequently, previous studies have attempted to integrate end-users directly into the development process to better account for their needs and preferences. In the section that follows, we provide an overview of these efforts.

Domain-oriented Approaches for Developing Smart Personal Assistants

User integration (i.e., domain expert integration) into the SPA development process is a relatively new phenomenon. While several studies have investigated SPAs' technological foundations, relatively few have examined SPAs' socio-technical design from the domain expert's perspective [102]. When reviewing the existing literature on user-oriented methods for developing SPAs, we identified three categories of approaches: First, those that focus on the collection and evaluation of interaction requirements from the user's perspective; Second, role-play approaches that focus on evaluating the solution; and thirdly LCDPs that allow the user to directly build a new SPA without possessing programming skills or experience.

Various studies have examined the collection and evaluation of user interactions [72, 84] based on the premise that preliminary explorations of the interaction logic are possible without implementing the actual interface. For instance, Wobbrock et al. [130] developed an approach to deriving gestures from users by first illustrating the effects of a gesture and then asking potential users to perform its cause. When it comes to the evaluation of interaction requirements, Jiang et al. [54] described an approach in which predefined tasks for controlling the SPA are simulated to determine the user's satisfaction with the interaction. In addition, scenario techniques (i.e., typical scenarios for controlling a device) are used to evaluate user satisfaction with these interaction scenarios [60]. However, these approaches provide only limited possibilities for users to become active participants in the development process.

The second category of user-oriented development approaches relates to role-playing approaches, which have been used in a variety of studies [105]. For instance, Lee et al. [71] used a modified "Wizard of Oz" approach [24] when potential users evaluated an SPA in the early design stages. Using this approach, the researchers were able to illustrate how the users would perform input gestures and how they would create feedback for the SPA to further improve the system. However, these role-play approaches do not provide direct design implications because the actual system is not implemented and therefore the users cannot attain a sufficient understanding of the overall system [124].

The third category of approaches relates to the provision of development to foster the development of software by people within a certain application domain. Different technology providers (such as Amazon or Google) offer rich ecosystems with intuitive interfaces that allow many users to create SPAs without having programming knowledge or experience. These SPA ecosystems may also be classified as low code development platforms (LCDPs), as they "*enable a rapid application delivery with a minimum of hand-coding and quick set-up and deployment*" [97, p. 2]. These LCDPs (like Amazon's Alexa Skills Kit) empower users to combine pre-programmed

components on a visual interface and to create functional SPAs [11]. Consequently, SPAs can be built on a much simpler way using LCDPs than using traditional development processes [101]. The accelerated speed of development [30] is, for instance, visible in the over 100,000 chatbots that were created on Facebook Messenger within one year [55]. Amazon's Alexa has also exhibited rapid growth in terms of her skillset: in 2016, Alexa possessed 300 skills; by 2020, she possessed more than 100,000 skills worldwide [120]. However, as Iacucci et al. [51] pointed out, even with the help of these LCDPs, developing proper interactions can be rather complex, involving various preparatory tasks and sufficient knowledge of basic interaction rules. This problem is, for instance, reflected in the actual usage of the skillset provided by Amazon's Alexa: analysis shows that 61% of all skills have never received a user rating [58].

In sum, the outlined approaches provide insights into how the conversation with a SPA and thus the end user's experience might be improved. However, all the outlined approaches have specific disadvantages. The first two categories of approaches still follow the logic of classical systems development, whereby SPAs are built by programmers and together with domain experts according to our definition and therefore provide little guidance for our method. The last category seeks to empower domain experts without programming skills or experience to leverage their expertise in the desired usage context as well as in the context of existing user needs and to autonomously develop SPAs quickly. Here, the challenge lies in the fact that the provision of the capabilities to develop SPAs with programming knowledge or skills alone is often insufficient to ensure the development of SPAs that offer satisfying user experiences.

The last category sublimates in the research stream of end user development (EUD) [4]. EUD refers to the empowerment of people who are not professional software developers to develop and adapt digital artifacts in a way that fits their existing practices, knowledge, or skills [74]. In this regard, it is important to point out that EUD not only refers to the integration of end users (i.e., consumers) but can also be applied to the integration of domain experts into the digital artifact development process [23, 39, 106]. It is also essential to note that EUD not only addresses programming activities but encompasses the entire software development process. Within EUD, domain-specific requirements regarding the digital artifact are considered, as domain experts can contextualize the artifacts not only during the design process but also during the later usage phases [4]. This extension distinguishes EUD from other user-oriented development approaches [116]. Because EUD can contextualize new digital artifacts during both development and use, it has aroused significant interest in research and practice. Consequently, an increasing number of studies have extended EUD to new application domains, such as the Internet of Things [1, 5], embedded technologies [68, 125], and intelligent technologies [25, 78]. Consequently, this last category seems a good starting point for our research.

Research Methodology

Design science research (DSR) is a pragmatic research paradigm that promotes the creation of innovative and valuable artifacts to solve real-world problems [46, 50]. According to Kuechler and Vaishnavi [65], DSR analyzes the use and performance of

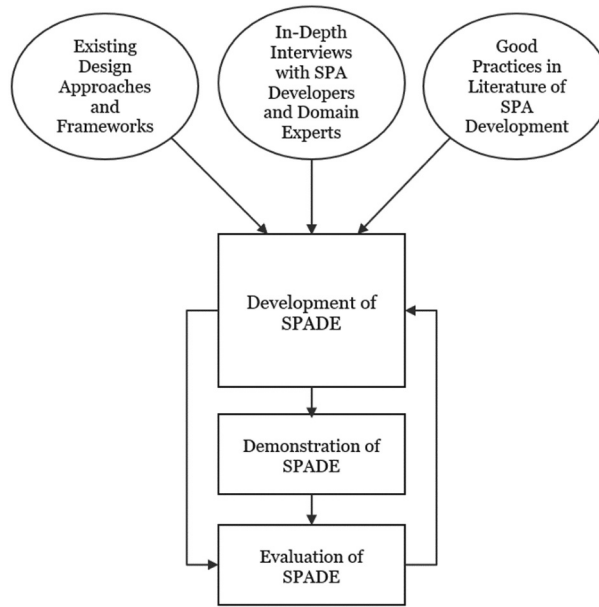


Figure 1. Design science research approach for Smart Personal Assistant for Domain Experts (SPADE). Adapted from Kolfshoten and de Vreede [64].

designed artifacts to understand, explain, and often improve on those artifacts. DSR projects develop these artifacts by identifying requirements from theory and from the application domain, translating them into a corresponding instantiation, and rigorously evaluating this instantiation (i.e., 40, 54, 77). When developing our method SPADE, we applied these guidelines, as DSR constitutes a widely used and accepted approach within the field of information systems. In Figure 1, we summarize our DSR approach, which is adapted from Kolfshoten and de Vreede [64]. We implemented the depicted activities and an iterative process to create SPADE—an SPA design and development method for domain experts.

In the first step, we closely examined existing design approaches and frameworks. We identified the research stream of end user development, which relies on the integration of the end-user into the creation or modification of digital artifacts. The domain-oriented design environments (DODEs) approach particularly informed our method. Furthermore, we identified what current low code platforms lack during our in-depth interviews with SPA developers and domain experts. We further asked the domain experts about the guidance they would need to build SPAs. Finally, we relied on theory applied to the development of SPAs and for this last step, we adopted a theory-motivated design approach [14] to capture the non-intuitive design choices that add value to the development and might lead to outperforming SPAs. In the realm of our design approach, we iteratively demonstrated and evaluated SPADE. First, we conducted a proof-of-concept evaluation by interviewing 39 domain experts with experience of developing SPAs in a low code environment. We then evaluated the proposed method's value by asking domain experts in various domains (e.g., smart home, education, and finance) to apply the method to their SPA development. Based

on the change requests during the evaluation, SPADE was adjusted and refined to support domain experts in designing and developing SPAs.

SPADE: A Method to Develop Smart Personal Assistant for Domain Experts: Iterative Design and Evaluation

The SPA method for domain experts (SPADE) guides domain experts through the process of designing and developing SPAs without the help of a developer and without the need for programming skills or prior experience. SPADE comprises twelve activities, with corresponding key questions and artifacts. By following the method chronologically (from activity one to twelve), domain experts are unlikely to miss any important steps when designing and developing SPAs. After each activity, the domain expert should have an artifact that builds the basis for the next activity. The resulting SPA should be fully functional and contextualized in the corresponding domain.

Design Approaches and Frameworks

As shown in [Figure 2](#), we began by reviewing existing design approaches and frameworks in related domains. To inform the development of SPADE, we drew on the body of knowledge on existing approaches regarding the end user's involvement in the creation or modification of digital artifacts. This body of knowledge can be subsumed under the research stream of EUD [4].

EUD has gained increasing interest in research and practice due to the characteristics of domain experts. Domain experts often lack knowledge and skills regarding programming and implementing digital artifacts. Consequently, it is often insufficient to emulate the principles of traditional, user-centered development approaches. Rather, EUD relies on dedicated development frameworks and tools that can be mastered by domain experts and that can be integrated into their domain practice [36]. In the domain of EUD, these design frameworks are called domain-oriented design environments (DODEs) [39]. DODEs were developed to facilitate the integration of domain experts into the process of developing new digital artifacts and allow them to act as designers throughout the development and use process [37, 38]. These environments thereby support domain experts in achieving two goals: exploring new and different design solutions for a given problem (instead of using existing solutions) and developing a corresponding digital artifact [39]. To reach these goals, DODEs rely on a combination of several components that must be in place to ensure the proper development of a new digital artifact by a domain expert. A specification component allows domain experts to specify their problems. A development toolkit consists of basic solution elements. An evaluation component allows users to assess their design. A simulation component helps to view the effects of certain decisions during the design process. And there is also a catalog of design solutions that domain experts can build upon when developing new digital artifacts. In the section that follows, we use these components of DODEs as a knowledge foundation in determining our method's requirements [Table 1](#).

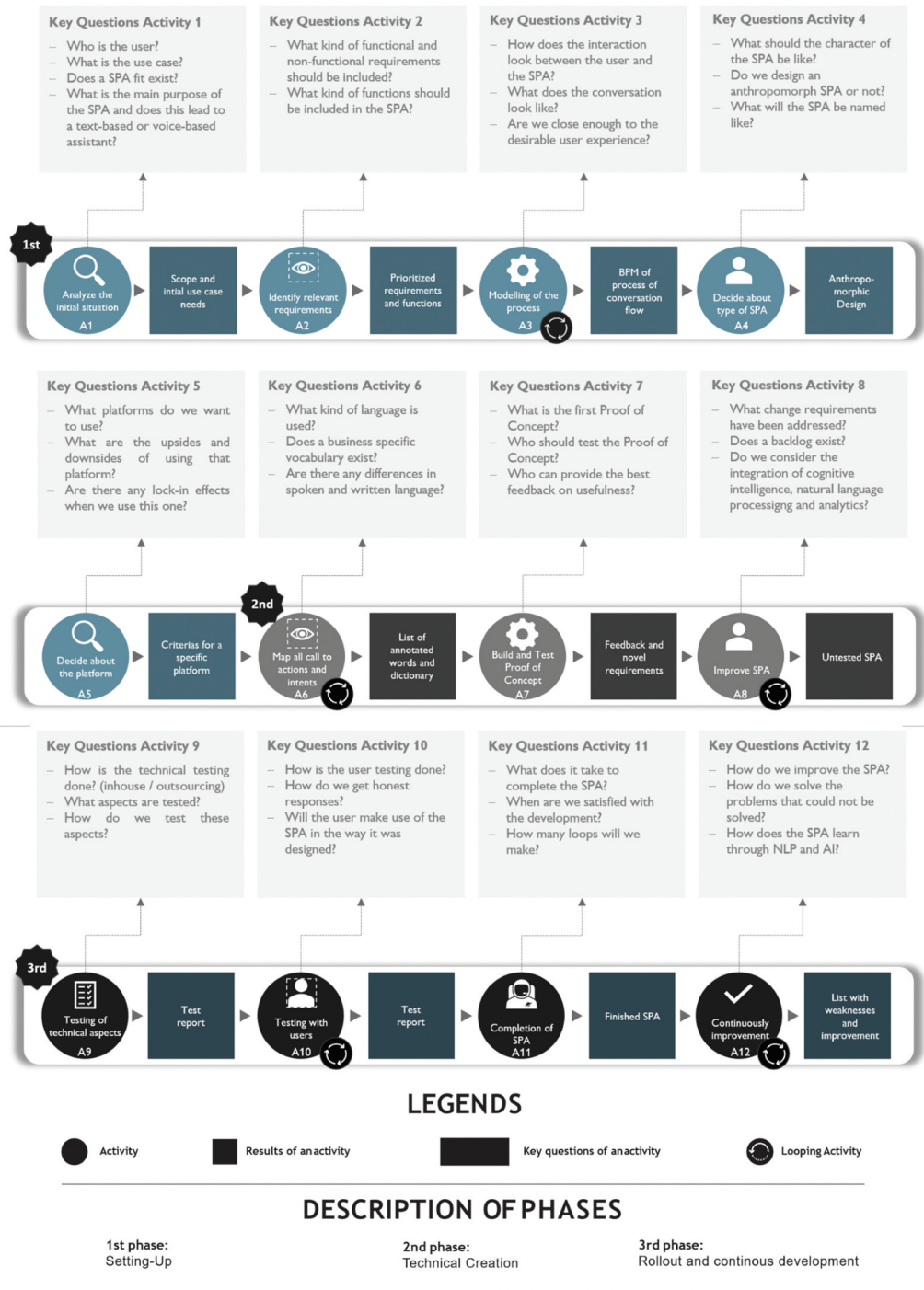


Figure 2. Smart Personal Assistant for Domain Experts (SPADE) including activities, artifacts, and key questions.

Table 1. Requirements derived from literature on end-user development.

Component of domain-oriented design environments	Resulting meta-design requirements
Specification component [36]	The method should allow domain experts to specify the problem that they want to address.
Development toolkit [37, 38]	The method should provide domain expert the possibility to make use of a construction kit that consists of basic solution elements.
Catalog of design solutions [39]	The method should allow domain experts to make use of a catalog of design solutions that domain experts can build upon when developing new solutions for similar problems.
Evaluation component [36]	The method should allow domain experts to evaluate their design decision.
Simulation component [39].	The method should allow domain experts to anticipate the effects of their design decisions during the design process.

We next drew on corresponding SPA-specific literature. For our SPADE, we organized the existing body of knowledge on SPA development and design into six clusters: analyzing user needs, modeling the interaction process, building the interaction model, specifying the communication, considering the adaptivity, and error handling. We then used each cluster to derive a requirement for SPADE. Table 2 lists these components and their associated requirements.

SPA Designers' Design Practices

Although the analyzed literature provided insights into novel aspects of SPA development, it yielded no specific insights into the design choices and challenges that domain experts face. We thus interviewed (1) SPA developers (programming experts) and (2) domain experts (no prior programming knowledge or experience).

First, we interviewed four experienced SPA developers (more than 10 years' experience) from the programming departments of Microsoft, IBM, ABB, and a western European country-based medium-sized company called VoicePoint. We interviewed them to gather insights into how programming experts would proceed when developing SPAs and to gather transferable insights

Table 2. SPA-specific requirements from literature.

Components	Resulting meta-design requirements
Analyzing User Needs [43, 129]	R1: A method for developing SPAs should start with analyzing the user's needs (e.g., conversation preferences, goals, etc.) in detail, in order to understand the kind of assistance that must be provided by the SPA.
Modeling Interaction Process [77, 91]	R2: A method for developing SPAs should guide the developer to model the interaction process that users should run through before technically implementing the SPA.
Building Interaction Model [17, 104, 123]	R3: A method for developing SPAs should include guidance on how to create an interaction model.
Conversation Design [21, 57, 103, 122, 123]	R4: A method for developing SPAs should assist the developer in the decision converting the input as well as the output form of a SPA and provide guidance on the conversation flow.
Considering Adaptivity [42, 99]	R5: A method for developing SPAs should enable the developer to consider various alternative flows within a conversation.
Error Handling [3, 47, 66]	R6: A method for developing SPAs should provide the developer with guidance on how to include error handling mechanisms in the SPA.
Considering Anthropomorphic Design Elements [2, 19, 27, 35]	R7: A method for developing SPAs should include guidance on how to incorporate anthropomorphic design elements in the SPA to stimulate socially desirable behaviors.

Notes: SPA, smart personal assistants.

for SPADE. These interviews lasted about an hour each and were structured as follows. In the first part of the interview, we explained our research project to the SPA developers and then asked them to describe their routine SPA development processes. In the second part, we asked them to specifically list and prioritize the obstacles they faced during design and development.

Second, we interviewed 24 domain experts to gain insights into how we could guide them in autonomously designing and developing SPAs. All 24 experts were educators who worked in different types of schools and various subjects. These domain experts did not have any prior programming knowledge; however, they were familiar with low code SPA development. These interviews lasted around 40 to 60 minutes. We interviewed the domain experts to obtain further details as to how we could support them in SPA design and development. Furthermore, these interviews helped us to better understand why LCDPs still fail. Thereby, in the first part of the interview, we described our research project and then asked them about their experience with the development of SPAs on LCDPs. Following up on their experiences, we asked them to point out challenges and obstacles while designing and developing their SPAs and the expected guidance. These interviews were conducted by two of the researchers and lasted about an hour.

We transcribed the interviews and analyzed them for three things: the reason for specific design choices; the outcomes that should result from a particular design choice; and the expected guidance during SPA development. We also used the method of user stories proposed by Cohn [22]. User stories are part of an agile approach that helps shift the focus from writing about requirements to talking about them. User stories include a written sentence or two about the artifact’s desired functionality [22]. We coded and clustered the user stories with the help of Mayring’s qualitative content analysis [79]. Qualitative content analysis seeks interesting issues based on the theoretical background and research question, which determine the aspects of the textual material taken into account. Finally, we translated the user stories into SPADE components. The user stories are listed in Table 3.

According to the interviewees, the use case and its scope must be clearly defined. If this step is skipped, the project might be doomed to fail from the beginning, as the user’s needs are unknown; the developed SPA may not meet the stakeholder’s expectations and requirements. By specifying the scope, misunderstandings can be resolved, and disappointments prevented.

Table 3. User stories derived from interviews.

User Stories
As a domain expert that would like to design a SPA, I want to ...
... first define and clarify the use case with the stakeholder, so I can set the scope of the project.
... know what kind of needs the users have, so I can address the needs with the SPA functions.
... improve the SPA by analyzing errors and misunderstandings continuously.
... determine if it should be a text-based or voice-based SPA.
... test the interaction with SPAs in different loops with different people to ensure the functionality when the SPA goes live.
... test the technical functions of the SPA, to ensure the functions, run reliably.
... model the interaction process that a user runs through so I can be sure I address the most important aspects of the interaction.
... assess what the user will say in the conversation to capture most of the intents.
... build a proof of concept and test it to know if the SPA runs smoothly.
... determine which learning goals, contents, and methods I want to use the SPA for.
... design SPAs by myself without needing a lot of technological knowledge.
... test the SPA by myself before I use it in my work environment.

Notes: SPA, smart personal assistants.

Furthermore, in line with our experts, after the scope is specified, the domain expert should consider the SPA/user interaction. As evidenced by the literature and the experience of the interviewees, the SPA's complexity of the SPA depends on the communication between the SPA and the human. Therefore, the domain expert should decide whether a simple SPA with a text function is sufficient or if a voice-based SPA is needed [132]. Following the decision regarding the communication, the domain experts should consider the interaction process between the SPA and the human; only by knowing how the user will use the SPA can the domain expert model the process accurately and then build the SPA. Furthermore, the interviewees mentioned that technical testing and user testing are crucial steps that should not be neglected. In addition, the domain expert should not stop after the development of the SPA and should consider continual improvement.

Developing SPADE

When designing SPADE in the first iteration of our design science project, we aimed to integrate the requirements from theory and practice logically, aligning them with classis software development processes. Thus, in a next step, we sequentially ordered the development activities and named the results after each development activity. Moreover, we added key questions to every activity to further guide the domain expert. In the next step, we divided the activities into three phases: (1) setting up, (2) technical creation, and (3) rollout and continuous development. The individual phases and associated activities are described in detail below. Additionally, recommendations for the realization of the activity are given at the appropriate places in the text. [Figure 2](#) illustrates the entire SPADE approach, and the approach is described in detail in the following section.

Phase 1: Setting Up

In activity 1, the domain expert should first analyze the initial situation. To have a clear understanding of what is expected, the domain expert should define the use case precisely according to the end user's requirements [109]. The assistance that an SPA provides can vary from general—such as searching for something on the web or playing music [103]—to specific, such as asking questions for an evaluation of a service [127]. To create an SPA that perfectly matches the users' needs, the domain expert should collect information about the end user. Therefore, it is essential to identify motivated people who are willing to help with the use case and, in later stages, with testing. For this activity, the creation of a canvas about the persona respectively user and their needs may help keep the design and development user-oriented [18].

Moreover, the domain expert should assess whether an SPA is the right fit for the use case in question. [Table 4](#) lists items that check for a use case's SPA fit. If the majority of these items are applicable to the domain experts' use cases, it might be advisable to design and develop the SPA for the given task or topic. Conversely, if this is not the case, it may be more advisable to forego the use of an SPA. In addition to that, SPAs are especially effective in settings with a high number of requests, repeating queries, or users; thereby, the potential usage frequency and scalability must be considered.

Furthermore, within this activity, the domain expert should decide whether the SPA should be built for a short-term or long-term interaction [87]. In addition, the domain experts have to think about whether they wish to create a text-based, voice-based, or hybrid-

Table 4. Smart personal assistant fit assessment for activity 1 (according to Google [45]).

Items to Check for SPA Fit
Users already have human-to-human conversations about this task or topic.
The interaction is brief, with minimal back-and-forth dialog.
Users would have to tap multiple times to complete the task with a screen.
Users might have to navigate multiple apps to complete the task with a screen.
The feature is difficult or cumbersome to find.
Users can do this task while multitasking.
Users can do this task when their hands or eyes are busy.
Users feel comfortable talking or typing about this topic.

based SPA [132]. This decision can only be made if the use case for the SPA is clearly defined. The domain expert must specify both the input provided by the user and the output provided by the SPA. The output provided by the SPA may be textual, voice-based [103], or sensorial [52]. While users interact with text-based SPA primarily on a screen via text or buttons, the interaction with voice-based SPA is in natural language via the user’s voice. Text-based SPAs are frequently used in support services, where they assist users with frequently asked questions and guide them through the support process, while voice-based SPAs have been classified as the next generation of service encounters [62]. Additional advantages of voice-based SPAs arise in complex tasks, such as following navigational paths [13]. As a result of this step, the domain expert can assess the SPA’s fit, determine the project’s scope, and identify the user’s needs.

In activity 2, the domain expert should derive functional and non-functional requirements for the SPA based on activity 1 by applying different technologies collaboratively with users or alone through brainstorming, interviews, or field observation [100]. This step minimizes the risk of delivering an SPA that fails to comply with the stakeholders’ ideas and needs. Additionally, the domain expert should prioritize the most important requirements to focus on the necessary requirements for the SPA in the SPADE activities that will follow. Consequently, the domain expert should compile a list of prioritized requirements.

In activity 3, the domain expert should model the interaction process between the user and the SPA. Although contemporary SPA design platforms usually come with tools to create conversational interfaces, these platforms do not define the interaction [83]. As highlighted by Følstad and Brandtzæg [40], designing SPAs presents a challenge to interaction foundations, since the focus in SPA development has moved from graphical interfaces toward conversational flows. These conversational flows are not fully predictable and depend strongly on the users’ input. Therefore, the domain expert should dedicate time to analyzing the potential interaction process between the user and the SPA. In doing so, the domain expert can anticipate the users’ possible communications and the corresponding SPA answers. There are multiple options to create a first conversation. First, we can make direct use of existing sources if SPA-capable databases or documented user dialogs are available. If none of these prerequisites exist, the relevant data must be developed in-house, for example, by the marketing department or the existing sources must be configured in such a way that they are useable, or knowledge may be obtained externally if accessible [82]. This activity can be performed with the help of conversational storyboards or conversational flows, as illustrated in Figure 3.

To build the conversational flow, we suggest that domain experts start with a task-oriented dialog and specify what the conversation's aim should be. For example, one might want to build a SPA for ordering flowers. Thereby, the aim of the conversation would be "Buy flowers". The domain expert should then consider different alternatives and, therefore, subcategories of the aim (e.g., roses and tulips). Next, the domain expert should include different questions that the SPA might ask the user, such as what color the roses should be or where they should be from. A more granular conversation might take some time to design until the domain expert has addressed all possible and relevant conversation pathways and potential user decisions. Finally, the conversation ending should be designed. Here, the domain expert should think about when and how the conversation should conclude. For example, the conversation in the given example might end after the user has selected the flowers and confirmed the purchase. Table 5 provides an overview of this expository example.

Furthermore, according to the speech act theory proposed by Searle et al. [111], the domain expert should clearly specify whether the SPA is committing to something's being the case (constatives, e.g., answering a question posed by the user), attempting to get the user to do something (directives, e.g., "Choose from the following options"), committing to some future course of action (commissive, e.g., "I will remind you in two weeks") or expressing its attitude toward the user with respect to some social action (acknowledgments, e.g., "I thank you for your input"). The design of this aspect is strongly related to the SPA's actual goal.

Additionally, the domain expert should consider the direction of the interaction, which might be user-to-SPA [17], SPA-to-user [104], or bidirectional [123]. Independently from the

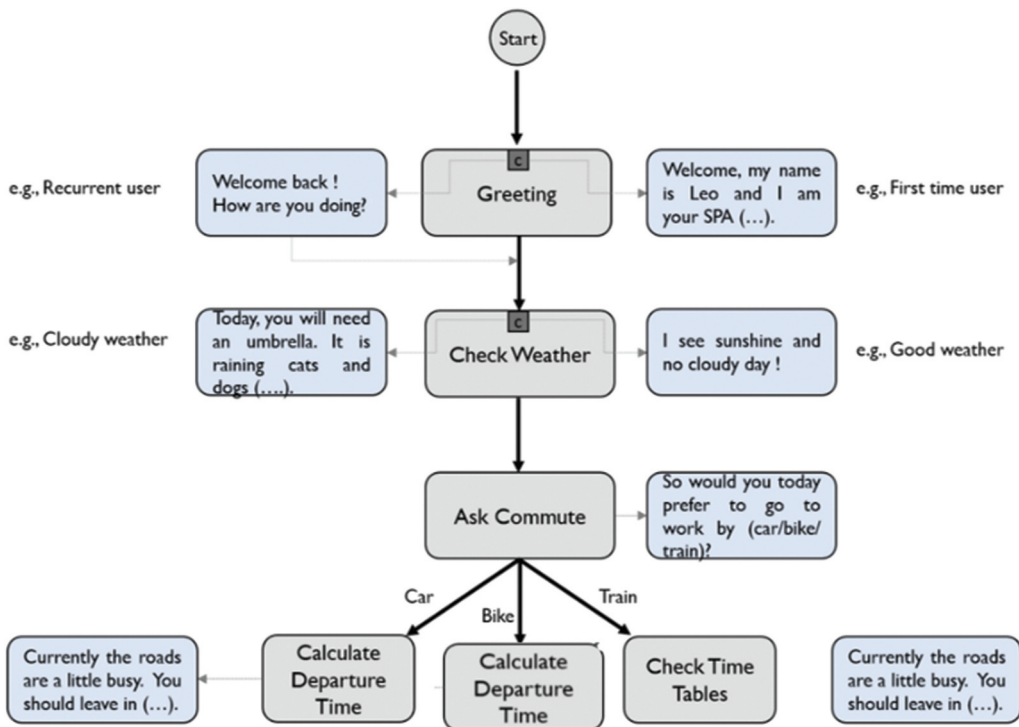
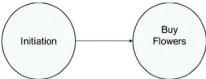
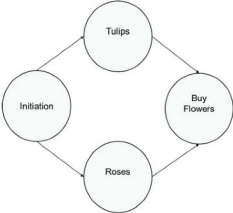
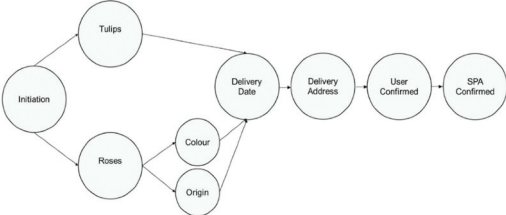


Figure 3. Expository modeling of conversational flows for activity 3 (based on Choi et al. [20]).

Table 5. Building conversations step by step for activity 3.

Conversational flow building	Graphical representation
Buy flowers	
Add choices of flowers	
Ending of the conversation	

use case, the interaction can be modeled with different kinds of modeling languages. The modeling of the interaction process helps detect possible user touchpoints and corresponding SPA user communication. To create meaningful interactions with the SPAs, the domain experts can take advantage of conversational UX patterns. One example design pattern is called “Inquiry (User) Repairs.” This pattern handles requests for information initiated by the user. When the agent understands the user’s inquiry, it produces a response for the user. If the user does not understand the answer, they can ask another question. But since exchanges during dialogue may be missed or misconstrued and communicative interactions are prone to failure, the domain expert should consider implementing recovery strategies within the conversation design [6]. Table 6 presents the recovery strategies and corresponding examples.

In activity 4, depending on the end user’s requirements and the use case, the domain expert should decide on the SPA’s personality and anthropomorphic elements of the SPA. The decisions within this activity should be made with two questions in mind: First, does this design decision enhance the end user’s enjoyment in interacting with this SPA, and second, does the SPA’s personality and design overshadow the task the SPA is expected to perform? The domain expert must specify the degree to which the SPA may be identified as human-like, the verbal aspects, and the nonverbal aspects [113]. Within this activity, the domain experts must decide whether they wish to design a human-like visual representation of the SPA. For instance, the SPA may be represented by an anthropomorphic avatar or a real human face (e.g., [7, 94]). Furthermore, the domain expert should decide on the SPA’s demographic characteristics. Bearing in mind that most of the time the SPA will be customer-facing, the

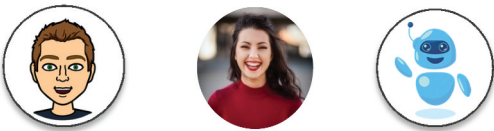
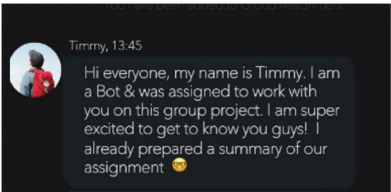
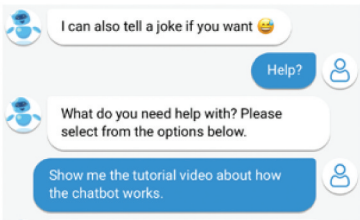
Table 6. Recovery strategies (according to Benner et al. [6]) and expository implementations.

Recovery strategy	Subcategories		
Confirmation	Explicit Rejection	Implicit Rejection	Move-On
Example:	When SPAs are unable to deliver a reasonable response to the user, SPAs must explicitly admit and confirm failure, i.e., by saying, “I do not know”.		
Information	Explanation	Help Messages	Error Feedback
Example	SPA will seek to educate the user on how to solve the breakdown and thereby will describe the situation, send possible useful information, and provide feedback on the occurrence.		
Disclosure	Disclose SPA	Disclose Weakness	Disclose Competences
Example	By revealing flaws and competencies, the user is aware of the SPA’s skills and potential inadequacies. Thus, the domain expert can implement this recovery strategy to further refine the user’s expectations and comprehension of the SPA.		
Social	Apologize	Compensate	Pause & Turn-Taking
Example	When applying this recovery technique, the SPA will attempt to elicit empathy and understanding from users in a manner similar to that seen in human-to-human dialogues.		
Solve	Utterance Template	Options (Multiple)	Alternative (Single)
Example	In this case, the SPA will try to actively solve the conversation failure, for instance, by providing the user with pre-programmed utterances.		
Ask	Repeat Question	Rephrase	Reprompt Input
Example	By implementing this strategy, the burden of recovering the breakdown moves to the user. In this realm, the user has to reframe their question.		

domain expert should consider the messages inherent in the selection of a gender (female, male, or undefined), an age (younger or older than the customer), a name (typical, i.e., relative to typical human names associated with the country in which the user resides, or special, i.e., a name describing the service offered such as Smart Home Assistant), and an ethnicity for their SPA (e.g., [88]). The domain expert can then infuse the conversational style with the SPA’s personality (e.g., being open, modest, friendly). Furthermore, the domain expert shall include self-references such as “me” and “I” for the SPA as well as an active voice (e.g., [19]). Concerning the nonverbal aspects, the domain expert should consider turn-taking gestures, such as pauses when speaking or “is typing” indicators (e.g., [44]). If the SPA is voice-based, the domain expert should additionally consider the pitch and volume of the voice [Table 7](#).

In activity 5, the domain expert should decide on the preferred SPA low code development platform. Currently, the SPA could be built on several major platforms (such as Amazon’s Alexa, Google’s Assistant, IBM’s Watson, and Microsoft’s Cortana) as well as numerous minor platforms (such as VoiceFlow). The most suitable platform depends on the requirements and interaction mode [75, 96]. The choice of platform may also depend on where the SPA is to be implemented. For instance, if it is to be implemented internally in a company, it must be integrated into the intranet works. Additionally, the domain expert should consider what they wish to do with the data generated in the interaction with the SPA. Here, it is important to emphasize that some platforms do not necessarily save the user’s conversations with the SPA. This could be used as a selection criterion, for example. Furthermore, the platform choice should also depend on the degree of freedom when designing the SPA; that is, the developer should check whether the representative can be adjusted. Rather, it should also be checked to what extent the domain expert can influence

Table 7. Expository anthropomorphic design elements (based on [109, 113]).

Anthropomorphic design element	Description	Example
Human-like or Human Representation	Images of human-like avatars, fictional characters, real human faces or humanoid robots	
Demographic Information	Gender, Age, Name, Ethnicity	If the representation is humanoid, a human name would make sense, e.g., Timmy [31]. 
Communication	Self-references ("me", "myself", "I"), active voice	

the design of the conversation and the interface. Some voice operations are not cross-platform, and extra work might be required if features such as flash briefings are to be supported on both Google and Amazon.

Phase 2: Technical Creation

In activity 6, before the domain expert can commence the SPA’s actual development, they must identify the dictionary to be applied for the use case. This dictionary should consist of possible user communications and SPA responses, specifying the words and phrases users can say to activate the SPA’s core functionality (also often characterized as “intent”). Depending on the use case, the dictionary might have different specifications (e.g., specifications for business). Moreover, the domain experts’ application of language might differ from that of SPA users’, particularly in business-specific contexts. To capture all the words that the user might use, the domain experts should work with the user or should at least be user-centered. It is advised that the domain expert use short and simple phrases that have a more comprehensive appeal and are accessible to a range of users. The domain expert should also include fillers, such as “aha,” “okay,” and “um” pauses [62], and an invocation name that allows the user to start a conversation with the SPA. Later, the dictionary should

be integrated into the SPA's interaction model. With the help of machine learning techniques, the SPA can understand users' intentions and choose the right answer. This dictionary can be presented in different languages.

In activity 7, the domain expert should build the SPA. The previous activities and artifacts have laid the basis for the building of the SPA. If a decision or activity has been skipped or omitted, it must be completed at this point. If during the building of the SPA, the domain expert notices that something is inaccurate or has not been predetermined, they must return iteratively (as shown in [Figure 2](#)) to the previous activities until the SPA reaches the draft stage. As soon as a draft version or a minimally viable product is available, the domain expert should test it with the user or any other expert to detect anomalies [98]. Different approaches for testing the SPA with users are available (e.g., think-out-loud or screen recordings of interaction with text-based SPAs). Furthermore, a survey can be conducted to validate the design decisions made. At this stage, it seems reasonable to test the SPA with someone who is not familiar with the SPA or the project that the domain expert has been working on to expose possible usability issues. The user's feedback should be documented to facilitate later adjustments [100]. Although feedback documentation is important, users might be prone to using the SPA differently when recorded or taped. Therefore, we strongly advise taking notes of any issues arising during this stage of testing and asking for feedback concerning the experience with the SPA (e.g., whether it met their expectations) [Table 8](#).

In activity 8, the domain expert should review the feedback, conduct an impact-effort analysis of the suggested changes, and decide which changes should be implemented.

Phase 3: Rollout and Continuous Development

The third and last phase of SPADE starts with the domain expert's testing of the technical aspects in activity 9. Either way, test documents will be used to examine the SPA's functionalities, and subsequently, a test report should be generated. This helps to understand the SPA's flaws and improve it for future users. In activity 10, alongside technical testing, the SPA should be tested with potential users [29] to receive meaningful feedback. After running the tests with the users, a test report will be produced. Using the test reports

Table 8. Items to focus on during user testing for activity 7.

Items to focus on during user testing	Exemplary questions to be answered
Natural Conversation Flow [83]	Does the conversation between the user and the SPA seem natural or forced? Is the user able to speak freely, or does the user feel the urge to speak in keyword-like phrases? Does the turn-taking function as intended? Does the user say anything that you did not expect?
Locus of Control [41]	Does the interaction with the SPA look like intended? Does the user feel alright with the control of the conversation?
User Satisfaction [117]	Does the user look satisfied or frustrated with the interaction and conversation? Are there any situations where the user feels confused or overwhelmed?
Intention to Use [70]	Assuming you had access to the SPA, would you intend to use it? Assuming you had access to the SPA, would you plan to use it?
Ease of Use [126]	Does using this SPA improve your performance in the given context? Does using the SPA improve your effectiveness in the given context? Is the SPA useful for supporting you in your given context?

from the technical testing and the user testing, changes and corrections can be made to the SPA, and the SPA can essentially be completed. This represents the end of activity 11 and mostly of SPADE. However, the final activity—activity 12—is to continuously improve the SPA to maintain its usability and retain user satisfaction. When completed chronologically, these activities should guide domain experts to systematically structure the design of SPAs to improve quality and save time and costs.

Evaluation

With the goal of addressing the real-world problem of domain experts wanting to design SPAs in an efficient and useful manner without the assistance of a programmer, we developed SPADE. In the following, we will elaborate on the first two stages of the “last research mile,” which involves the proof of concept as well as the proof of value [15, 89].

Proof of Concept

In accordance with Nunamaker et al. [89], we created the first initial version of SPADE. This first evaluation concentrates on the artifact’s constituents and on our design decisions concerning SPADE. According to Sonnenberg and vom Brocke [118], it is important to base evaluations of design science projects on two aspects: (1) the artifact’s constituents and design decisions; and (2) the artifact’s usefulness [118]. Therefore, the evaluation’s purpose is to ensure the completeness and correctness of each activity and to determine whether domain experts would use SPADE for the development of their SPAs. The domain experts were primarily from the business development and marketing departments at a variety of companies as well as educators. Each of these domain experts for the development of SPADE was asked to comment on SPADE individually. Thereby, we conducted 39 interviews to evaluate SPADE. The interviews were conducted individually with the participants and lasted around 30 minutes. They were structured as follows. First, we showed them SPADE and explained each activity and the resulting artifacts as well as the corresponding key questions. Second, we asked them to indicate whether they would like to articulate change requests (see Online Supplemental Appendix 1) for further improvement of SPADE. We then refined SPADE as per the obtained feedback.

Proof of Value

With the aim of ensuring the value of SPADE and understanding its feasibility, we applied it in three different domains (smart home, finance, and education). To demonstrate SPADE’s practical applicability and that it is generally true for low code development of SPAs regardless of the choice of channel, it has been applied in two different contexts (smart home and strategic decision-making within companies) and channels (i.e., voice- and text-based assistants). The second evaluation was conducted with 38 domain experts in the field of education. While we illustrate the in-depth applications of SPADE and user satisfaction in the two contexts, we focus on the applicability, usability, and efficacy of SPADE in the replication by domain experts.

In-Depth Application of SPADE in a Smart Home Context¹

The application of SPADE in the smart home context offers a good expository instantiation, as smart home technologies have made significant technological progress in recent years and therefore can be considered intelligent systems. These smart home technologies (SHT) offer

numerous functions, such as controlling lighting, climate, kitchen equipment, entertainment systems, and other appliances. In this case, the focus of the SPA was to provide guided task support (i.e., the SPA could control the majority of these smart home system functions). Before examining the actual development of the SPA by the domain experts, we would like to provide more information on the domain experts. The domain experts were marketing specialists from a large smart home technology provider called TechCorp. For the development of the SPA, the two marketing specialists worked closely together with the product manager of the SHT at TechCorp. The domain experts were between the ages of 25 and 40 years and had no prior programming knowledge. Below, we can observe the development of this SPA through its three phases and the related activities. As we cannot present detailed results of each activity due to space limitations, we will focus on the most interesting results from our perspective. The remaining results are available from the authors upon request.

Phase 1: Setting Up. The first activity in the SPADE was to analyze the situation; therefore, smart homes and their users were examined. SHTs are as pervasive as SPAs. These two technologies' ubiquity is based on the embedding of information and communication technologies (ICTs) within consumer goods, which promises improved functionality, connectivity, and controllability [49]. The emergence of smart homes can ensure that intelligent technologies become part of peoples' everyday lives. Nevertheless, there remain some problems with technology today; users rarely have simple introductions to their smart homes. The communication with the SHT should be natural and simple, without requiring the user to understand complex technical language [81]. To solve this matter, domain experts built an SPA to help the user become acquainted with the essential functions of their SHT. Furthermore, the domain experts had to make the decision regarding the SPA channel: voice-based vs text-based. By analyzing the context of an SHT and the current media discontinuity, the advantages and disadvantages of the two SPA channels in this specific scenario were discovered. The decision regarding which SPA channel to use was then straightforward. Most of the time, users cannot use their hands while also using smart home functions (e.g., while cooking). For example, when they want to program their oven, users prefer to speak to the SPA rather than typing. Therefore, the decision was made to build a voice-based SPA.

Furthermore, in activity 2, it is necessary to identify the SPA's relevant requirements. To gather the requirements and pain points of an SHT owner, the domain experts identified a list of problems in consultation with SHT owners and employees operating in the technical support field of SHT. The interviews asked what was currently going wrong as well as what could support problem-free handling of the SHT.

Next, a current SHT instruction manual was analyzed with the objective of identifying its different functions and flaws. Following the pairing of the flaws in the current solution with the requirements from a users' perspective, the final set of requirements for the development of the SPA was established. In activity 3, the domain experts modeled the process of the SHT's first usage from a user's perspective by asking an SHT owner to describe the process. In this sense, the original interaction process included an instance of media discontinuity, as the SHT user must switch from smart technology to an instruction manual.

Furthermore, no interaction occurs between the user and the SHT in the current process. Therefore, the introduction of the SPA in this scenario had to amend the media discontinuity as well as the interaction between the user and the SHT. Activity 4 included the decision regarding SPA type. The SHT owners could choose whether they wished the

assistant to be female or male. Therefore, the domain experts chose to leave the choice up to the SHT owner. In designing their assistant's personality, they chose to name it Toni as a gender-neutral name. Toni should be friendly and polite but still able to provide entertainment by making small talk and jokes.

Based on the decision regarding the SPA channel, multiple platforms could be chosen. At the time this SPA was built (2019), Google and Amazon were particularly well-represented platforms, given that third parties can program an SPA on their platform with minimal expenditure. For this project, Amazon's Alexa was chosen as the SPA platform. Specifically, the domain experts used Alexa's Skill Development Kit 2.0 with Node.js. This framework seemed to offer one of the most advanced, state-of-the-art capabilities regarding speech recognition and natural language processing. Additionally, Alexa's Skill Development Kit 2.0 provided various blueprints for the smart home context [28].

Phase 2: Technical Creation. To define a dictionary for the SPA, the domain experts created a storyboard to simulate a conversation. This storyboard was created with the help of the instruction manual (by reading it and deriving the instructions the SPA must give to the user) as well as the assistance of an SHT owner. An example storyboard is illustrated in Figure 4, which illustrates the optimum interaction between user and SPA. The domain expert also accounted for the possibility of several interactions; for example, the user may be greeted differently after first-time usage.

After defining the dialog and the potential conversation flow, the designed storyboards were connected to build the SPA. The specified communications in the storyboards map back to the intents within the interaction with the SPA, which represent the actions that the user must follow sequentially. In the example above, the intents that a user might express are the same as the functions the SHT has. Therefore, the response and prompt of the SPA are conceptualized to explain the SHT functions to the user. Consequently, for the various user conversations, the domain expert started from basic communication and created individual story lines within the SPA. For instance, a storyline was created for the function of setting an oven, including the corresponding possible answers.

Subsequently, the SPA was created within the developer console of Amazon Alexa. Following this, the user intents were added, and the back-end rules were set accordingly

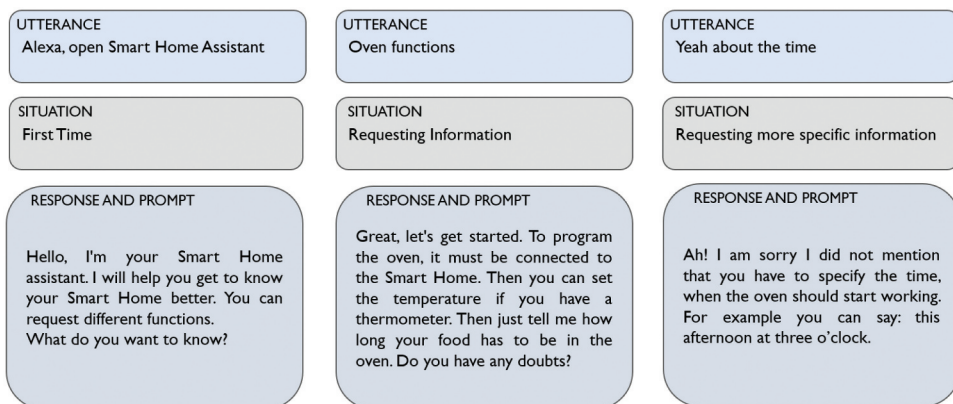


Figure 4. Storyboard of oven functions as a result of activity 6.

to the previously created storyboard. For the purpose of the SPA being perceived as funny, about 20 “Easter egg” intents were created to offer the curious user some entertainment. In activity 8, the SPA was tested internally at the SHT company by four technical support employees. The test users were asked to interact with the SPA and to think out loud about what they were doing and what bothered them. The internal testing identified some possible improvements. For instance, the chosen “wake word” (the word that would activate the SPA) proved unsuitable, as it was too long. The SPA struggled to understand the word as a whole since it consisted of two compound words. Therefore, the wake word had to be altered from “*smart home technology assistant*” to “*smart home assistant*”. Furthermore, some of the SPA’s sentences were too complex and lengthy, and the test user had trouble remembering the SPA’s instructions. For this reason, the sentences were shortened and simplified to support a better understanding. The Easter egg intents were also omitted after the first testing, both due to technical problems (e.g., the Easter eggs had no contexts and therefore confused the SPA) and in response to feedback, which indicated the desire for a more serious conversation design.

Phase 3: Roll Out and Continuous Development. Entering the last phase of the SPADE, activity 9 was completed, with technical adjustments made. In activity 10, the technical aspects were tested with the help of Amazon’s developer testing function. Subsequently, a test report listing the main change requests was created. In activity 11, the SPA was tested with possible users. This final testing of the SPA prior to it entering the market ensured that it performed all functions correctly and smoothly. An example dialog of the designed SPA can be found in Online Supplemental Appendix 2. In this activity’s second iteration, the domain experts tested the voice-based SPA with potential customers in one of their showrooms at TechCorp within a time span of one week. Participants in this user testing were homeowners or individuals who had considered implementing an SHT but did not yet own one. To ensure that only individuals with no prior knowledge of SHT or SPAs participated, the contacted individuals were required to take a pretest. The participants received no monetary reward. In total, 42 individuals were interested in participating in user testing and completed the pretest. Of these 42 potential participants, only 32 could ultimately participate, as some had prior experience with SPAs or SHT. The approved participants were instructed as to how to use their SHT SPA. Subsequently, the participants were allotted 20 minutes to solve a task and, with the help of their SPA, became as well acquainted with the many SHT functions as possible. At the end of the user testing, the participants were required to conduct a post-test activity by completing an online questionnaire with closed-ended questions relating to solving the scenario illustrated below (e.g., how to set the oven timer to 15 minutes). When aiming to solve the task, on average, four out of five questions were answered correctly. Thereby the domain experts assumed that the instructions are given by the SPA and, therefore also its conversation design required no further adjustments. Finally, the participants were asked about their satisfaction with the SPA, which resulted in 4.2 out of 5.

Although this paper includes no discussion of activity 12, continuous improvement of the SPA is essential. The domain experts adapt the SPA with each new functionality that is added to the SHT. In addition, domain experts continue to ensure that the SPA meets users’ needs and learns a broader vocabulary (such as different terms to control shutters).

In-Depth Application of SPADE in a Company-Internal Context

Before examining the domain experts' actual development of the SPA, we wish to introduce this second in-depth study. In the course of this proof of value application, three domain experts built a chat-based assistant. This was a company-internal assistant that was to be used to receive inputs. All domain experts were aged between 25 and 30 years old and thus were familiar with new technologies. They all worked at FinCorp as part of this project. None of the domain experts had programming skills. As we cannot present detailed results of each activity due to space limitations, we will focus on the most salient results from our perspective. The remaining results are available from the authors upon request.

Phase 1: Setting Up. The domain experts first analyzed the initial situation. For example, they determined what the assistant should do and what their goal was. The company faces the challenge that highly complex IT projects often fail due to a lack of stakeholder integration. This causes companies to lose millions in IT projects. This situation must change, as the company has come under severe pressure owing to the digital transformation. For this reason, managers must gather feedback from the IT project's stakeholders as quickly as possible to take action and steer them in the right direction. Thus, the use case for the domain experts was as follows: to support the collection of employee feedback.

The assistant's target group was also defined. These were employees of the product management. Thus, the user group was homogeneous in its behavior. Next, the domain experts took a closer look at the users of their planned assistant and conducted interviews with product management staff to understand the current feedback process and requirements for the text-based SPA. This resulted in a list of functional and non-functional requirements for the chatbot in terms of process, conversation design, and the design of the text-based SPA. For example, a functional requirement was that the ability to reference previous conversations should be available. Although a non-functional requirement, the domain experts stated that the text-based SPA should have a friendly, entertaining, and humorous character. The domain experts also decided that they would build a text-based SPA for this use case.

After the requirements were defined, the domain experts defined the interaction between the assistant and the FinCorp employee. Since the domain experts already had good knowledge of Business Process Modeling Notation, they decided to model their interactions in that modeling language.

In this step, the domain experts also investigated how to model the conversation so that employees would be motivated to leave their feedback and ideas. For example, small talk was added to the conversation (see [Figure 5](#)). After the conversation was modeled, the next step was to examine more closely how the assistant should be designed. In this sense, the domain experts determined what personality the assistant should have: open and personal. In this step, the experts also decided which gender the avatar should have and how the assistant's representative should be designed.

The domain experts decided to build the text-based SPA on Blitzico's LCDP (EntrepriseBot's platform). To determine which platform was suitable, the domain experts looked at their use case and compared it with the possibilities offered by the various LCDP providers. The chosen platform was suitable for the project because fixed dialog flows are defined, and free text entries can easily be linked to follow-up questions. It was also important that the text-based SPA be integrated into the FinCorp intranet.

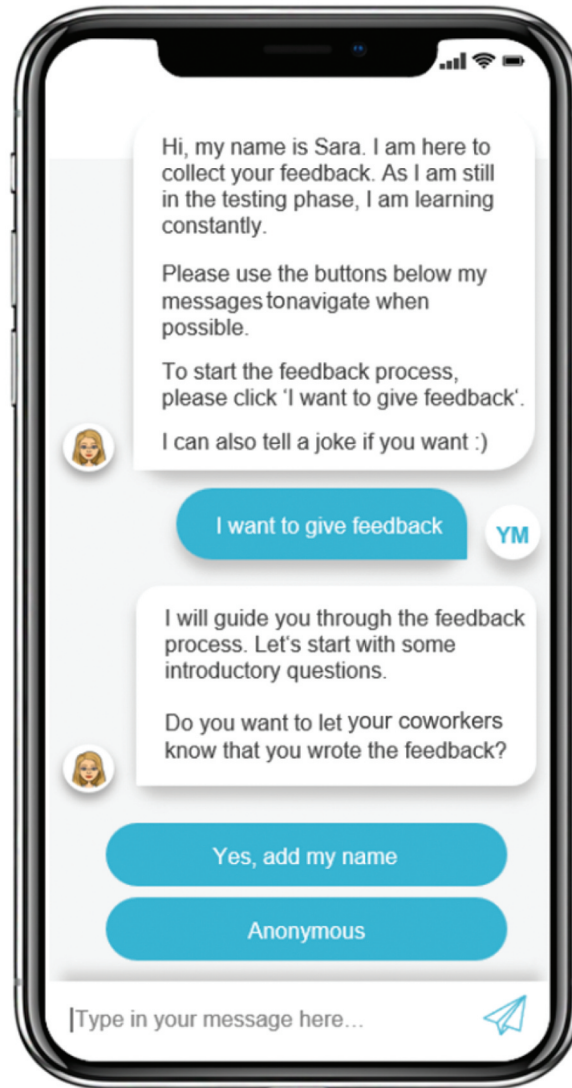


Figure 5. Expository dialog between user and smart personal assistants (SPA) for activity 3.

Phase 2: Technical Creation. To determine which language was used, the use case was examined again. It transpired that little to hardly any specific vocabulary should be included. A short list of business-specific terms was created, and this was incorporated into the first prototype of the text-based SPA. On the platform chosen by the domain experts, a predefined user flow is indirectly possible by capturing different intents, training phrases, and fallbacks. Thus, the domain experts developed and tested an initial prototype of the SPA. Testing of the prototype then took place with 20 FinCorp employees with the intention of testing assumptions regarding the conversation design and the design of the text-based SPA. This step was performed iteratively by the three domain experts. In the three iterations, it became apparent, for example, that the SPA could definitely work with

more emojis. However, the feedback also mentioned the interface, such as the recommendation that the messages should be divided into individual bubbles. The domain experts then implemented these adjustments step by step until the finished text-based SPA was finally available.

Phase 3: Rollout and Continuous Development. Before the rollout of the SPA could take place, a technical test was carried out to ensure that the SPA was functioning properly from a technical perspective—in particular, whether the integration into FinCorp’s intranet worked and whether the employees’ answers were also saved. In addition, with the help of several employees, the domain experts tested whether the SPA also works with several simultaneous inquiries. Thus, a survey was conducted in user testing to validate the selected design elements and the anatomy of the SPA. In order to test their SPA, the domain experts asked for the intention to use, the usefulness as well as for the overall satisfaction with the SPA. For this, they formulated assumptions, for example, *How helpful do you consider the conversational flow concerned with how to give good feedback?*, and tested them with a 5-point Likert scale and benchmarked them also against their current solution. In total, another 20 employees of FinCorp participated in the user testing. Overall, they were very satisfied with the SPA, which resulted in a score of 4.6 out of 5. When tested against their current solution, the user testing revealed that the newly proposed SPA was preferred in almost every case. Thereby, the domain experts incorporated the obtained feedback and planned the go-live of the SPA. The SPA is currently in use at FinCorp and is continuously adapted according to the users’ needs. For example, it was noted that users would like the SPA to be dressed differently depending on the season to make it more “human.” The domain experts have taken this on board and will make these adjustments according to seasonal conditions.

Discussion and Limitations

The goal of our paper was to examine how the development of smart personal assistants by domain experts can be facilitated within low code development environments. We, therefore, presented the Smart Personal Assistants for Domain Experts method. SPADE builds on existing knowledge of SPA development and EUD and was systematically built and tested in the course of our DSR study and apply it to two different contexts. Our proof-of-value evaluations show that SPADE can empower domain experts to develop SPAs in different channels (voice and text) that are perceived as valuable by end users (smart home context and finance context). Furthermore, the evaluation in the education context showed that domain experts consider SPADE to be useful and valuable in providing guidance in the SPA development process. This demonstrated SPADE’s applicability across different domains and across different SPA channels, highlighting its flexibility and providing first insights into the approach’s potential proof of use.

The contributions to existing research on SPAs are twofold. First, we add to the body of knowledge on how to design effective SPAs. People without programming skills or experience particularly struggle to develop effective SPAs even though many technology providers offer LCDPs. To address this issue, we developed a method that guides domain experts through the SPA development process step by step to ensure that they do not overlook non-intuitive design choices and activities in their development journey. Furthermore, we provide expository examples of the central activities of SPADE that distinguish the development of SPAs

from traditional software development. With this, we aim to provide additional support during the application of SPADE. From a theoretical perspective, this contribution resembles a theory of design and action [46]. Second, we contribute to the EUD literature by developing the initial version of a novel DODE for SPA development. Thereby, we demonstrate that the underlying logic of EUD also holds true in the SPA domain.

We further offer important implications for practice. SPAs offer various potentials in different areas of application. By applying our method, practitioners can now more easily and swiftly explore and leverage these potentials in low code environments. SPADE clearly and transparently structures the domain experts' work in 12 subsequent steps and includes non-intuitive design choices. The proposed key questions for each activity aid domain experts in addressing the SPA's specific development activities more effectively, and the template offers further guidance on the most challenging steps. SPADE is designed to work independently of any specific domain and of any programming skills or experience. Therefore, SPADE offers domain experts a practical means of documenting and recording the SPA development process using the artifacts that result after each fulfilled activity.

Conclusion

Many organizations intend to deploy and adopt SPAs within their organizational context and thus capitalize at the enormous potential. To leverage the potential, many organizations turn to LCDPs for the development of the SPA. However, this development approach still inherits some issues. Solely using the platform does not guarantee an useful and good conversation, since these platforms do not provide the required support for domain experts when it comes to non-intuitive design choices. We drew on a design science approach, where we systematically proposed, built and tested Smart Personal Assistant for Domain Experts (SPADE) method to address the missing link.

Note

1. A replication study conducted with domain experts in the field of education can be found in Online Supplemental Appendix 3.

Acknowledgments

We would like to thank the experts involved in the studies and also Rainer Winkler for his ideas in the early phases of this paper. Furthermore, this research builds on a paper that has been presented at the 53rd Hawaii International Conference on System Sciences 2020. We thank the reviewers and attendees for their valuable feedback that helped us to improve our research and to write this paper. Last but not least, we thank the Special Issue Senior Editors for the guidance as well as the two anonymous reviewers for their constructive feedback during the review process.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

We thank the Swiss National Science Foundation for funding parts of this research (100013_192718). P. Ebel acknowledges funding from the Basic Research Fund (GFF) of the Open access funding provided by University of St. Gallen.

Notes on contributors

Edona Elshan (edona.elshan@unisg.ch; corresponding author) is a research associate at the Institute of Information Management at the University of St.Gallen, Switzerland. She holds a bachelor's degree in Business Administration and Information systems from the University of Zurich, Switzerland, and a master's degree in Business Innovation from the University of St. Gallen, Switzerland. Her research focuses on low code development, cloud native platforms, conversational interfaces, and composable applications. Her work has been published in leading information systems and management journals and conference proceedings.

Philipp Ebel (philipp.ebel@unisg.ch) is an assistant professor and head of a research group at the Institute for Information Management at the University of St.Gallen. His research focuses on the integration of internal and external knowledge sources into digital innovation ecosystems, the (agile) development of new products and services, and the systematic design of human-machine-collaborations. Dr. Ebel's research has been published in such journals as *Information Systems Journal*, *Electronic Markets*, *Business & Information Systems Engineering*, *International Journal of Innovation Management*, and *International Journal of Entrepreneurship Venturing*.

Matthias Söllner (soellner@uni-kassel.de) is full professor and chair of Information Systems and Systems Engineering, as well as a director of the inter-disciplinary Research Center for Information Systems Design at the University of Kassel. His research focuses on understanding and designing successful digital innovations in domains such as higher education, vocational training, and hybrid intelligence. His research has been published in such journals as *Journal of Management Information Systems (JMIS)*, *MIS Quarterly (Research Curation)*, *Journal of the Association for Information Systems*, *European Journal of Information Systems*, *Journal of Information Technology*, and others. Dr. Söllner serves on the editorial boards of *European Journal of Information Systems* and *Journal of Information Technology*. He has received several awards and recognitions for his research, teaching, and community service.

Jan Marco Leimeister is Chaired Professor and Managing Director of the Institute of Information Management at the University of St.Gallen, Switzerland. He is furthermore a Director at the Research Center for IS Design at the University of Kassel, Germany. His work covers Digital Business, Digital Transformation, Digital Service Management, Crowdsourcing, Digital Work, Digital Learning Services and Collaboration Engineering. Jan Marco has been internationally recognized for outstanding research, teaching and education. He ranks repeatedly since 2009 among the top 1% of the most productive researchers and professors in business administration in the German-speaking area. He serves on the AIS Leadership Council, is Co-Editor in Chief of the *Journal of Information Technology (JIT)* and serves on the Editorial Board of *Journal of Management Information Systems (JMIS)* and *Information Systems Research (ISR)*. He is also an Entrepreneur and works as Senior Advisor, Board Member and Keynote Speaker for national and international organizations.

ORCID

Edona Elshan  <http://orcid.org/0000-0002-4737-2392>

Philipp Ebel  <http://orcid.org/0000-0001-9505-2484>

Matthias Söllner  <http://orcid.org/0000-0002-1347-8252>

Jan Marco Leimeister  <http://orcid.org/0000-0002-1990-2894>

References

1. Akiki, P.A.; Bandara, A.K.; and Yu, Y. Visual simple transformations: Empowering end-users to wire internet of things objects. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 24, 2 (2017), 1–43.
2. Araujo, T. Living up to the chatbot hype: The influence of anthropomorphic design cues and communicative agency framing on conversational agent and company perceptions. *Computers in Human Behavior*, 85(2018), 183–189.
3. Ashktorab, Z.; Jain, M.; Liao, Q.V.; and Weisz, J.D. Resilient chatbots: Repair strategy preferences for conversational breakdowns. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, Glasgow, UK, 2019, pp. 1–12.
4. Barricelli, B.R.; Cassano, F.; Fogli, D.; and Piccinno, A. end-user development, end-user programming and end-user software engineering: A systematic mapping study. *Journal of Systems and Software*, 149(2019), 101–137.
5. Barricelli, B.R.; and Valtolina, S. Designing for end-user development in the internet of things. In *Proceedings of International Symposium on End User Development*, Cham, Switzerland, 2015, pp. 9–24.
6. Benner, D.; Elshan, E.; Schöbel, S.; and Janson, A. What do you mean? A review on recovery strategies to overcome conversational breakdowns of conversational agents. In *Proceedings of the International Conference on Information Systems*, Austin, USA, 2021, pp. 1–15.
7. Berry, D.C.; Butler, L.T.; and De Rosis, F. Evaluating a realistic agent in an advice-giving task. *International Journal of Human-Computer Studies*, 63, 3 (2005), 304–327.
8. Betton, F. 2022. Why are “no code” and “low code” software development platforms on the rise? *ITProPortal*. <https://www.itproportal.com/features/why-are-no-code-and-low-code-software-development-platforms-on-the-rise/> (last accessed on November 4, 2022).
9. Bickmore, T.W. and Picard, R.W. Establishing and maintaining long-term human-computer relationships. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 12, 2 (2005), 293–327.
10. Bitner, M.J.; Brown, S.W., and Meuter, M.L. Technology infusion in service encounters. *Journal of the Academy of Marketing Science*, 28, 1 (2000), 138–149.
11. Bock, A.C. and Frank, U. Low-code platform. *Business & Information Systems Engineering*, 63, 6 (December 2021), 733–740.
12. Brandtzaeg, P.B. and Følstad, A. Chatbots: Changing user needs and motivations. *Interactions*, 25, 5 (2018), 38–43.
13. Brennan, S.E. Conversation as direct manipulation: An iconoclastic view. In *Proceedings of The Art of Human-Computer Interface Design*, Boston, USA, 1990, pp. 393–404.
14. Briggs, R.O. On theory-driven design and deployment of collaboration systems. *International Journal of Human-Computer Studies*, 64, 7 (2006), 573–582.
15. Briggs, R.O.; Nunamaker, J.F.; and Sprague, R. Special section applied science research in information systems: The last research mile. *Journal of Management Information Systems*, 28, 1 (2011), 13–16.
16. Brinker, S. Democratizing martech: Distributing power from IT to marketing technologists to everyone. 2018. *Chief Marketing Technologist*. <https://chiefmartec.com/2018/05/democratizing-martech-marketing-technologists/> (last accessed November 4, 2022).
17. Campagna, G.; Ramesh, R.; Xu, S.; Fischer, M.; and Lam, M.S. Almond. The architecture of an open, crowdsourced, privacy-preserving, programmable virtual assistant. In *Proceedings of the 26th International Conference on World Wide Web*, Perth, Australia, 2017, pp. 341–350.
18. Chang, Y.; Lim, Y.; and Stolterman, E. Personas: From theory to practices. In *Proceedings of the 5th Nordic Conference on Human-Computer Interaction: Building Bridges*, Lund, Sweden, 2008, 439–442.
19. Chattaraman, V.; Kwon, W.-S.; Gilbert, J.E.; and Ross, K. Should AI-based, conversational digital assistants employ social-or task-oriented interaction style? A task-competency and reciprocity perspective for older adults. *Computers in Human Behavior*, 90, (2019), 315–330.

20. Choi, Y.; Shin, H.; Monserrat, T.-J.K.; Lee, N.; Park, J.; and Kim, J. Supporting an iterative conversation design process. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, New York, USA, 2020, pp. 1–8.
21. Clark, L.; Pantidi, N.; Cooney, O.; Doyle, P.; Garaialde, D.; Edwards, J.; Spillane, B.; Gilmartin, E.; Murad, C.; Munteanu, C.; Wade, V.; and Cowan, B.R. What makes a good conversation? Challenges in designing truly conversational agents. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, Glasgow, UK, 2019, pp. 1–12.
22. Cohn, M. *User Stories Applied: For Agile Software Development*. Boston: Addison-Wesley Professional, 2004.
23. Costabile, M.F.; Fogli, D.; Fresta, G.; Mussio, P.; and Piccinno, A. Building environments for end-user development and tailoring. In *Proceedings of the 2003 IEEE Symposium on Human Centric Computing Languages and Environments*, Piscataway, USA, 2003, pp. 31–38.
24. Dahlbäck, N.; Jönsson, A.; and Ahrenberg, L. Wizard of Oz Studies: Why and How. In *Proceedings of the 1st International Conference on Intelligent User Interfaces*, Orlando, USA, 1993, pp. 258–266.
25. Desolda, G.; Ardito, C.; and Matera, M. Empowering end users to customize their smart environments: Model, composition paradigms, and domain-specific tools. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 24, 2 (2017), 1–52.
26. Diederich, S.; Brendel, A.B.; Morana, S.; and Kolbe, L. On the design of and interaction with conversational agents: An organizing and assessing review of human-computer interaction research. *Journal of the Association for Information Systems*, 23,1 (2022), 96–138.
27. Diederich, S.; Lichtenberg, S.; Brendel, A.B.; and Trang, S. Promoting sustainable mobility beliefs with persuasive and anthropomorphic design: Insights from an experiment with a conversational agent. In *Proceedings of International Conference on Information Systems*, Munich, Germany, 2019, pp. 1–17.
28. Doring, M. 2018. Build skills faster with version 2 of the ASK software development kit for Java. <https://developer.amazon.com/blogs/alexa/post/1a4e8b01-663d-4680-8efd-c28e96e31655/now-available-version-2-of-the-ask-software-development-kit-for-java> (last accessed November 4, 2022).
29. Dumas, J.S.; Dumas, J.S.; and Redish, J. *A Practical Guide to Usability Testing*. Exeter: Intellect Books, 1999.
30. Elshan, E.; Dickhaut, E., and Ebel, P. An investigation of why low code platforms provide answers and new challenges. In *Proceedings of 56th Hawaii International Conference on System Sciences (HICSS)*, Maui, Hawaii, 2023, in press.
31. Elshan, E. and Ebel, P. Let's team up: Designing conversational agents as teammates. In *Proceedings of 41st International Conference on Information Systems*, Online, 2020, pp. 1–9.
32. Elshan, E.; Zierau, N.; Engel, C.; Janson, A., and Leimeister, J.M. Understanding the design elements affecting user acceptance of intelligent agents: Past, present and future. *Information Systems Frontiers*, 24, 3 (2022), 699–730.
33. Engelhardt, S.; Hansson, E.; and Leite, I. Better faulty than sorry: Investigating social recovery strategies to minimize the impact of failure in human-robot interaction. In *Proceedings of WCIHAI@ IVA*, Stockholm, Sweden, 2017, pp. 19–27.
34. Epley, N.; Waytz, A.; and Cacioppo, J.T. On seeing human: A three-factor theory of anthropomorphism. *Psychological Review*, 114, 4 (2007), 864–886.
35. Feine, J.; Gnewuch, U.; Morana, S.; and Maedche, A. A taxonomy of social cues for conversational agents. *International Journal of Human-Computer Studies*, 132, (2019), 138–161.
36. Fischer, G. *End-user Development: From Creating Technologies to Transforming Cultures*. International Symposium on End User Development. Cham: Springer, 2013, 217–222.
37. Fischer, G. and Giaccardi, E. *Meta-design: A Framework for the Future of End-user Development*. End user development. Cham: Springer, 2006, pp. 427–457.
38. Fischer, G.; Giaccardi, E.; Ye, Y.; Sutcliffe, A.G.; and Mehandjiev, N. Meta-design: A manifesto for end-user development. *Communications of the ACM*, 47, 9 (2004), 33–37.
39. Fischer, G.; Nakakoji, K.; and Ye, Y. Metadesign: Guidelines for supporting domain experts in software development. *IEEE Software*, 26, 5 (2009), 37–44.

40. Følstad, A. and Brandtzaeg, P.B. Chatbots and the new world of HCI. *Interactions*, 24, 4 (2017), 38–42.
41. Følstad, A.; Skjuve, M.; and Brandtzaeg, P.B. Different chatbots for different purposes: Towards a typology of chatbots to understand interaction design. In *Proceedings of International Conference on Internet Science*. St. Petersburg, Russia, 2018, pp. 145–156.
42. Garimella, H.A.S. 2017. Robust online i-vectors for unsupervised adaptation of DNN acoustic models: A study in the context of digital voice assistants. <https://www.amazon.science/publications/robust-online-i-vectors-for-unsupervised-adaptation-of-dnn-acoustic-models-a-study-in-the-context-of-digital-voice-assistants> (last accessed November 4, 2022).
43. Ghosh, S. and Pherwani, J. Designing of a natural voice assistants for mobile through user centered design approach. In *Proceedings of 17th International Conference on Human-Computer Interaction*, Los Angeles, USA, 2015, 320–331.
44. Gnewuch, U., Morana, S., Adam, M., and Maedche, A. Faster is not Always better: Understanding the effect of dynamic response delays in human-chatbot interaction. In *Proceedings of European Conference on Information Systems*. Portsmouth, UK, 2018, pp. 1–17.
45. Google. Conversation design. 2020. <https://designguidelines.withgoogle.com/conversation/conversation-design-process/is-conversation-the-right-fit.html#is-conversation-the-right-fit-take-the-quiz> (last accessed November 4, 2022).
46. Gregor, S. The nature of theory in information systems. *MIS Quarterly*, 30, 3 (2006), 611–642.
47. Guo, Y.R. and Goh, D.H.-L. Affect in embodied pedagogical agents: Meta-analytic review. *Journal of Educational Computing Research*, 53, 1 (2015), 124–149.
48. Gupta, S. and Vajic, M. The contextual and dialectical nature of experiences. In Fitzsimmons, J. and Fitzsimmons, M., (eds.), *New Service Development: Creating Memorable Experiences*. Thousand Oaks: Sage, 2000, pp. 33–51.
49. Hargreaves, T. and Wilson, C. *Smart Homes and Their Users*. Cham: Springer, 2017.
50. Hevner, A.R.; March, S.T.; Park, J.; and Ram, S. Design science in information systems research. *MIS Quarterly*, 28, 1 (2004), 75–105.
51. Iacucci, G.; Kuutti, K.; and Ranta, M. On the move with a magic thing: Role playing in concept design of mobile services and devices. In *Proceedings of the 3rd Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, New York, 2000, pp. 193–202.
52. Jalaliniya, S. and Pederson, T. Designing wearable personal assistants for surgeons: An ego-centric approach. *IEEE Pervasive Computing*, 14, 3 (2015), 22–31.
53. Janssen, A.; Passlck, D.; and Rodriguez Cardona, D. Virtual assistance in any context - A taxonomy of design elements for. *Business & Information Systems Engineering*, 62, 3 (2020), 221–225.
54. Jiang, J.; Hassan Awadallah, A.; Jones, R.; Ozertem, U.; Zitouni, I.; Gurunath Kulkarni, R.; and Khan, O. Z. Automatic online evaluation of intelligent assistants. In *Proceedings of the 24th International Conference on World Wide Web*, Florence, Italy, 2015, pp. 506–516.
55. Johnsson, M. The innovation facilitator: Characteristics and importance for innovation teams. *Journal of Innovation Management*, 6, 2 (2018), 12–44.
56. Kim, Y. and Sundar, S.S. Anthropomorphism of computers: Is it mindful or mindless? *Computers in Human Behavior*, 28(2012), 241–250.
57. Kincaid, R. and Pollock, G. Nicky: Toward a virtual assistant for test and measurement instrument recommendations. In *Proceedings of IEEE 11th International Conference on Semantic Computing*. IEEE. San Diego, USA, 2017, pp. 196–203.
58. Kinsella, B. 61% of Alexa Skills Still Have No Ratings and Only 1% Have More Than 100. *Voicebot.ai*, 2018. <https://voicebot.ai/2018/10/05/61-of-alexa-skills-still-have-no-ratings-and-only-1-have-more-than-100/> (accessed on November 4, 2022).
59. Kinsella, B. 2020. Amazon Alexa skill growth has slowed further in 2020. *Voicebot.ai*, <https://voicebot.ai/2020/10/25/amazon-alexa-skill-growth-has-slowed-further-in-2020/> (last accessed November 4, 2022).
60. Kiseleva, J.; Williams, K.; Jiang, J.; Hassan Awadallah, A.; Crook, A. C.; Zitouni, I.; and Anastasakos, T. Understanding user satisfaction with intelligent assistants. In *Proceedings of*

- the 2016 ACM on Conference on Human Information Interaction and Retrieval*, Copenhagen, Denmark, 2016, pp. 121–130.
61. Knijnenburg, B.P. and Willemsen, M.C. Inferring capabilities of intelligent agents from their external traits. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 6, 4 (2016), 1–25.
 62. Knote, R.; Janson, A.; Söllner, M.; and Leimeister, J.M. Classifying smart personal assistants: An empirical cluster analysis. In *Proceedings of the 52nd Hawaii International Conference on System Sciences*, Maui, USA, 2019, pp. 2024–2033.
 63. Knote, R.; Janson, A.; Söllner, M.; and Leimeister, J.M. Value co-creation in smart services: A functional affordances perspective on smart personal assistants. *Journal of the Association for Information Systems*, 22, 2 (2020), 418–458.
 64. Kolfshoten, G.L. and de Vreede, G.-J. A design approach for collaboration processes: A multimethod design science study in collaboration engineering. *Journal of Management Information Systems*, 26, 1 (2009), 225–256.
 65. Kuechler, B. and Vaishnavi, V. On theory development in design science research: Anatomy of a research project. *European Journal of Information Systems*, 17, 5 (2008), 489–504.
 66. Kulkarni, G., Nagaraj, D., and Gupta, S. System and method for smart error handling mechanism for an application. Washington, DC: U.S. Patent and Trademark Office, 2017.
 67. Laranjo, L.; Dunn, A.G.; Tong, H.L.; Kocaballi, A.B.; Chen, J.; Bashir, R.; Surian, D.; Gallego, B.; Magrabi, F.; Lau, A.; and Coeira, E. Conversational agents in healthcare: A systematic review. *Journal of the American Medical Informatics Association*, 25, 9 (2018), 1248–1258.
 68. Lee, J.; Garduño, L.; Walker, E.; and Bursleson, W. A Tangible programming tool for creation of context-aware applications. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Zurich, Switzerland, 2013, pp. 391–400.
 69. Lee, S.; Lee, J.; and Lee, K. Designing intelligent assistant through user participations. In *Proceedings of the 2017 Conference on Designing Interactive Systems*, Edinburgh, UK, 2017, pp. 173–177.
 70. Lee, S.; Lee, N.; and Sah, Y.J. Perceiving a mind in a chatbot: Effect of mind perception and social cues on co-presence, closeness, and intention to use. *International Journal of Human-Computer Interaction*, 36, 10 (2020), 930–940.
 71. Lee, S.-S.; Chae, J.; Kim, H.; Lim, Y.; and Lee, K. Towards more natural digital content manipulation via user freehand gestural interaction in a living room. In *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Zurich, Switzerland, 2013, pp. 617–626.
 72. Lee, S.-S.; Kim, S.; Jin, B.; Choi, E.; Kim, B.; Jia, X.; Kim, D.; and Lee, K. How Users manipulate deformable displays as input devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Atlanta, USA, 2010, pp. 1647–1656.
 73. Leimeister, J.M.; Huber, M.; Bretschneider, U.; and Krcmar, H. Leveraging crowdsourcing: Activation-supporting components for IT-based ideas competition. *Journal of Management Information Systems*, 26, 1 (2009), 197–224.
 74. Lieberman, H.; Paternò, F.; Klann, M.; and Wulf, V. End-user development: An emerging paradigm. In H. Lieberman, F. Paternò, and V. Wulf (eds.), *End User Development*. Dordrecht: Springer, 2006, pp. 1–8.
 75. López, G.; Quesada, L.; and Guerrero, L.A. Alexa vs. Siri vs. Cortana vs. Google Assistant: A comparison of speech-based natural user interfaces. In I. Nunes (ed), *Advances in Human Factors and System Interactions*. Cham: Springer, 2017, pp. 241–250.
 76. Luger, E. and Sellen, A. Like having a really bad PA: The gulf between user expectation and experience of conversational agents. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, San Jose, USA, 2016, pp. 5286–5297.
 77. Luria, M.; Hoffman, G.; Megidish, B.; Zuckerman, O.; and Park, S. Designing Vyo, a robotic smart home assistant: Bridging the gap between device and social agent. In *Proceedings of 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, New York, USA, 2016, pp. 1019–1025.

78. Martín, D.; Alcarria, R.; Sánchez-Picot, Á.; and Robles, T. An ambient intelligence framework for end-user service provisioning in a hospital pharmacy: A case study. *Journal of Medical Systems*, 39, 10 (2015), 116.
79. Mayring, P. Qualitative Inhaltsanalyse. In K. Mruck and G. Mey (eds.), *Handbuch Qualitative Forschung in der Psychologie*. Beltz: Springer, 2010, pp. 601–613.
80. McTear, M.F.; Callejas, Z.; and Griol, D. *The Conversational Interface*. Cham: Springer, 2016.
81. Mennicken, S.; Vermeulen, J.; and Huang, E.M. From today's augmented houses to tomorrow's smart homes: New directions for home automation research. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Seattle, USA, 2014, pp. 105–115.
82. Meyer Von Wolff, R.; Hobert, S.; and Schumann, M. Chatbot introduction and operation in enterprises – A design science research-based structured procedure model for chatbot projects. In *Proceedings of the 54th Hawaii International Conference on System Sciences*. Maui, USA, 2022, pp. 5933–5942.
83. Moore, R.J. and Arar, R. Conversational UX design: An introduction. In R.J. Moore, M. Szymanski, R. Arar, and G.J. Ren (eds.), *Studies in Conversational UX Design*. Cham: Springer, 2018, pp. 1–16.
84. Morris, M.R.; Wobbrock, J.O.; and Wilson, A.D. Understanding users' preferences for surface gestures. In *Proceedings of Graphics Interface 2010*, Ottawa, Canada, 2010, pp. 261–268.
85. Nass, C.; Fogg, B.J.; and Moon, Y. Can computers be teammates? *International Journal of Human-Computer Studies*, 45, 6 (1996), 669–678.
86. Nass, C.; Steuer, J.; and Tauber, E.R. Computers are social actors. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, Boston, USA, 1994, pp. 72–78.
87. Nißen, M.; Selimi, D.; Janssen, A.; Cardona, D. R.; Breitner, M. H.; Kowatsch, T.; and von Wangenheim, F. See you soon again, chatbot? A design taxonomy to characterize user-chatbot relationships with different time horizons. *Computers in Human Behavior*, 127, (2022), 107043.
88. Nunamaker, J.F.; Derrick, D.C.; Elkins, A.C.; Burgoon, J.K.; and Patton, M.W. Embodied conversational agent-based kiosk for automated interviewing. *Journal of Management Information Systems*, 28, 1 (2011), 17–48.
89. Nunamaker Jr, J.F.; Briggs, R.O.; Derrick, D.C.; and Schwabe, G. The last research mile: Achieving both rigor and relevance in information systems research. *Journal of Management Information Systems*, 32, 3 (2015), 10–47.
90. Pais, S.; Casal, J.; Ponciano, R.; and Lourenço, S. Unsupervised assistive and adaptive intelligent agent in smart environment. In *Proceedings of ICIES 2015: International Conference on Intelligent Environments and Systems*, Prague, Czech Republic, 2015, pp. 23–24.
91. Pearl, C. *Designing Voice User Interfaces: Principles of Conversational Experiences*. Sebastopol: O'Reilly Media, 2016.
92. Peffers, K.; Tuunanen, T.; Rothenberger, M.A.; and Chatterjee, S. A design science research methodology for information systems research. *Journal of Management Information Systems*, 24, 3 (2007), 45–77.
93. Qiu, L. and Benbasat, I. Evaluating anthropomorphic product recommendation agents: A social relationship perspective to designing information systems. *Journal of Management Information Systems*, 25, 4 (2009), 145–182.
94. Qiu, L. and Benbasat, I. A study of demographic embodiments of product recommendation agents in electronic commerce. *International Journal of Human-Computer Studies*, 68, 10 (2010), 669–688.
95. Radziwill, N.M. and Benton, M.C. Evaluating quality of chatbots and intelligent conversational agents. *Software Quality Professional*, 19, 3(2017), 25–36.
96. Rammohan, R.; Dhanabalsamy, N.; Dimov, V.; and Eidelman, F.J. Smartphone conversational agents (apple siri, google, windows cortana) and questions about allergy and asthma emergencies. *Journal of Allergy and Clinical Immunology*, 139, 2 (2017), AB250.
97. Richardson, C. and Rymer, J.R. *New Development Platforms Emerge for Customer-Facing Applications*. Cambridge: Forrester, 2014.

98. Ries, E. 2009. Minimum viable product: a guide. http://soloway.pbworks.com/w/file/fetch/85897603/1%2B%20Lessons%20Learned_%20Minimum%20Viable%20Product_%20a%20guide2.pdf (last accessed November 4, 2022).
99. Ruan, S.; Jiang, L.; Xu, J.; Tham, B.; Qiu, Z.; Zhu, Y.; Murnane, E.; Brunskill, E.; and Landay, J.A. QuizBot: A dialogue-based adaptive learning system for factual knowledge. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, Glasgow, Scotland, 2019, pp. 1–13.
100. Rupp, C. *Requirements-Engineering und-Management: Aus der Praxis von Klassisch bis Agil*. München: Carl Hanser Verlag, 2014.
101. Rymer, J.R.; Koplowitz, R.; Leaders, S.A.; Mines, C.; Sjoblom, S.; and Turley, C. *The Forrester WaveTM: Low-Code Development Platforms For AD&D Professionals, Q1 2019*. Cambridge: Forrester Research, 2019.
102. Sano, S.; Kaji, N.; and Sassano, M. Prediction of prospective user engagement with intelligent assistants. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Berlin, Germany, 2016, pp. 1203–1212.
103. Sansonnet, J.P.; Correa, D.W.; Jaques, P.; Braffort, A.; and Verrecchia, C. Developing web fully-integrated conversational assistant agents. In *Proceedings of the 2012 ACM Research in Applied Computation Symposium*, San Antonio, USA, 2012, pp. 14–19.
104. Sato, A.; Watanabe, K.; and Rekimoto, J. MimiCook: A cooking assistant system with situated guidance. In *Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction*. Munich, Germany, 2014, pp. 121–124.
105. Sato, S. and Salvador, T. COLUMNS-Methods and Tools-Playacting and Focus Troupes: Theater techniques for creating quick, intense, immersive, and engaging focus group sessions focus groups that are used to develop or. *Interactions*, 6, 5 (1999), 35–54.
106. Schobel, J.; Pryss, R.; Schickler, M.; Ruf-Leuschner, M.; Elbert, T.; and Reichert, M. End-user programming of mobile services: Empowering domain experts to implement mobile data collection applications. In *2016 IEEE International Conference on Mobile Services (MS)*. San Francisco, USA, 2016, pp. 1–8.
107. Schuetz, S. and Venkatesh, V. The rise of human machines: How cognitive computing systems challenge assumptions of user-system interaction. *Journal of the Association for Information Systems*, 21, 2 (2020), 460–482.
108. Schuetzler, R.M.; Grimes, G.M.; and Giboney, J.S. An investigation of conversational agent relevance, presence, and engagement. In *Proceedings of Americas Conference on Information Systems*, New Orleans, USA, 2018, p. 12.
109. Schuetzler, R.M.; Grimes, G.M.; Giboney, J.S.; and Rosser, H.K. Deciding whether and how to deploy chatbots. *MIS Quarterly Executive*, 20, 1 (2021), 4.
110. Schuetzler, R.M.; Grimes, G.M.; and Scott Giboney, J. The impact of chatbot conversational skill on engagement and perceived humanness. *Journal of Management Information Systems*, 37, 3 (2020), 875–900.
111. Searle, J.R.; Kiefer, F.; and Bierwisch, M. *Speech act theory and pragmatics*. Cham: Springer, 1980.
112. Seeber, I.; Bittner, E.; Briggs, R.O.; de Vreede, T.; de Vreede, G.-J.; Elkins, A.; Maier, R.; Merz, A. B.; Oeste-Reiss, S.; Randrup, N.; Schwabe, G.; and Söllner, M. Machines as teammates: A research agenda on AI in team collaboration. *Information & Management*, 57, 2 (2020), 103174.
113. Seeger, A.-M.; Pfeiffer, J., and Heinzl, A. Texting with human-like conversational agents: Designing for anthropomorphism. *Journal of the Association for Information Systems*, 22, 4 (2021), 8.
114. Shankar, V.; Smith, A.K.; and Rangaswamy, A. Customer satisfaction and loyalty in online and offline environments. *International Journal of Research in Marketing*, 20, 2 (2003), 153–175.
115. Shawar, B.A. and Atwell, E. Chatbots: Are they really useful? *Ldv forum*, 22, 1, (2007), 29–49.
116. Simonsen, J. and Robertson, T. *Routledge International Handbook of Participatory Design*. London: Routledge, 2012.
117. Skjuve, M.; Haugstveit, I.M.; Følstad, A.; and Brandtzaeg, P.B. Help! Is my chatbot falling into the uncanny valley? An empirical study of user experience in human-chatbot interaction. *Human Technology*, 15, 1 (2019), 30.

118. Sonnenberg, C. and vom Brocke, J. Evaluations in the science of the artificial – Reconsidering the build-evaluate pattern in design science research. In K. Peffers, M. Rothenberger, and B. Kuechler (eds.), *Design Science Research in Information Systems. Advances in Theory and Practice*. Berlin Heidelberg Springer, 2012, pp. 381–397.
119. Statista. 2019. Number of voice assistants in use worldwide 2019-2024. *Statista*. <https://www.statista.com/statistics/973815/worldwide-digital-voice-assistant-in-use/> (last accessed on November 4, 2022).
120. Statista. 2020. Amazon Alexa: Skill count in selected countries 2020. *Statista*. <https://www.statista.com/statistics/917900/selected-countries-amazon-alexa-skill-count/> (last accessed on November 4, 2022).
121. Stieger, M.; Nißen, M.; Rügger, D.; Kowatsch, T.; Flückiger, C.; and Allemand, M. PEACH, a smartphone-and conversational agent-based coaching intervention for intentional personality change: Study protocol of a randomized, wait-list controlled trial. *BMC Psychology*, 6, 1 (2018), 1–15.
122. Sugawara, K.; Manabe, Y.; Shiratori, N.; Yaala, S.B.; Moulin, C.; and Barthès, J.-P.A. Conversation-based support for requirement definition by a Personal Design Assistant. In *Proceedings of IEEE 10th International Conference on Cognitive Informatics and Cognitive Computing*, Banff, Canada, 2011, pp. 262–267.
123. Tsujino, K.; Iizuka, S.; Nakashima, Y.; and Isoda, Y. Speech recognition and spoken language understanding for mobile personal assistants: A case study of “Shabette Concier.” In *Proceedings of IEEE 14th International Conference on Mobile Data Management*, Milan, Italy, 2013, pp. 225–228.
124. Tur, G. Extending boosting for large scale spoken language understanding. *Machine Learning*, 69, 1 (2007), 55–74.
125. Turchi, T.; Malizia, A.; and Dix, A. TAPAS: A tangible end-user development tool supporting the repurposing of pervasive displays. *Journal of Visual Languages & Computing*, 39, (2017), 66–77.
126. Van der Heijden, H. User acceptance of hedonic information systems. *MIS Quarterly*, 28, 4 (2004), 695–704.
127. Wambsganss, T.; Winkler, R.; Schmid, P.S.; and Söllner, M. Designing a conversational agent as a formative course evaluation tool. In *Proceedings of 15th International Conference on Wirtschaftsinformatik*, Osnabrueck, Germany, 2020, 1234–1249.
128. Winkler, R.; Söllner, M.; and Leimeister, J.M. Enhancing problem-solving skills with smart personal assistant technology. *Computers & Education*, 165, (2021), 104148.
129. Winner, H.; Hakuli, S.; Lotz, F.; and Singer, C. *Handbook of Driver Assistance Systems: Basic Information, Components and Systems for Active Safety and Comfort*. Cham: Springer, 2016.
130. Wobbrock, J.O.; Morris, M.R.; and Wilson, A.D. User-defined gestures for surface computing. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*, Boston, USA, 2009, pp. 1083–1092.
131. Xin Ding, D.; Hu, P.J.-H.; Verma, R.; and Wardell, D.G. The impact of service system design and flow experience on customer satisfaction in online financial services. *Journal of Service Research*, 13, 1 (2010), 96–110.
132. Zierau, N.; Hildebrand, C.; Bergner, A.; Busquet, F.; Schmitt, A.; and Leimeister, J. Voice bots on the frontline: Voice-based interfaces enhance flow-like consumer experiences & boost service outcomes. *Journal of the Academy of Marketing Science*, (2022), 1–20.:
133. Zomerdijk, L.G. and Voss, C.A. Service design for experience-centric services. *Journal of Service Research*, 13, 1 (2010), 67–82.