

SPAM – A Process Model for Developing Smart Personal Assistants

Rainer Winkler
University of St. Gallen
Müller-Friedberg-Str. 8,
9000 St. Gallen
rainer.winkler@unisg.ch

Edona Elshan
University of St. Gallen
Müller-Friedberg-Str. 8,
9000 St. Gallen
edona.elshan@student.unisg.ch

Matthias Söllner
University of Kassel
Henschelstr. 4,
34127 Kassel
soellner@uni-kassel.de

Jan Marco Leimeister
University of St. Gallen
Müller-Friedberg-Str. 8,
9000 St. Gallen
janmarco.leimeister@unisg.ch

Abstract

Information technology capabilities are growing at an impressive pace and increasingly overstrain the cognitive abilities of users. User assistance systems such as online manuals try to help the user in handling these systems. However, there is strong evidence that traditional user assistance systems are not as effective as intended. With the rise of smart personal assistants, such as Amazon's Alexa, user assistance systems are becoming more sophisticated by offering a higher degree of interaction and intelligence. This study proposes a process model to develop Smart Personal Assistants. Using a design science research approach, we first gather requirements from Smart Personal Assistant designers and theory, and later evaluate the process model with developing an Amazon Alexa Skill for a Smart Home system. This paper contributes to the existing user assistance literature by offering a new process model on how to design Smart Personal Assistants for intelligent systems.

1. Introduction

Intelligent systems are becoming more and more sophisticated, which creates a gap between their functional scope and users' cognitive capabilities [1]. For example, when we have a look at the newest smart home technologies, they offer many functions, such as controlling lighting, climate, entertainment systems, and other appliances. For the user, it is sometimes overwhelming to learn all the functions of the system. Traditional user assistance systems such as user manuals or online helps have often tried to help users in getting to know the system. However, there is strong

empirical evidence that such support structures are not as effective as intended [2][3].

New emerging Smart Personal Assistants, such as Amazon's Alexa, Google's Assistant, or Apple's Siri, promise to offer a new way of user assistance. Smart Personal Assistants (SPAs) are computer agents that can assist users by engaging with them via natural dialogue [4]. Compared to traditional user assistance systems, SPAs are characterized by a high degree of interaction and intelligence [1]. They are able to react to user utterances, adapt their answers accordingly, and can build up a dialogue with them similar to human-human communication (interaction). Moreover, they are able to include contextual factors such as users' current knowledge state to adapt their answers to the user (intelligence). This new way of support enables the user to get to know a new system in a more efficient way. For example, Honold et al. [5] developed a home theater companion system that considers the entire situation of users and their environment in current and past states (e.g., the system recognizes which step the user is currently trying to conduct) in order to help users to understand and set up the home theater system in a more efficient way.

Past research in the field of advanced user assistance systems (AUAS) mainly focused on individual instantiations of SPAs. For example, Abdulrahmani [6] used Siri to create an AUAS for blind people. Current research lacks of transferable insights into how to design AUAS for intelligent systems. However, general knowledge on how to design AUAS is necessary to close the gap between increasing capabilities of intelligent systems and human capabilities [1]. Addressing this point, we propose the following research question:

How to design a process model for developing Smart Personal Assistants?

To answer our research question, we follow a design science research (DSR) approach by Hevner [7]. We start the rigor cycle by deriving requirements for a **Smart Personal Assistant process model** (SPAM) on the basis of the theory. Then, we derive requirements from interviews with SPA designers. Based on that, we develop our alpha version of SPAM and evaluate it with the SPA designers. After refining SPAM, we conduct a proof-of-usefulness evaluation by exemplarily developing an SPA for a smart home system.

The process model for developing SPAs represents a body of theoretical knowledge of the type *design and action* [8]. More precisely, it represents a nascent design theory of the type *improvement* [9]. In terms of practical contribution, we provide a process model for SPAs that guides developers in designing SPAs in a more efficient way.

The remainder of this paper is structured as follows. In section 2, we provide an overview of user assistance systems and classify SPAs. In section 3, we describe our design science research approach. Section 4 consists of the SPAM development, and in section 5, we discuss our results and end with a conclusion and a brief outlook.

2. Theoretical Background

2.1 User Assistance Systems

User assistance systems aim to help users to conduct their tasks better [1]. User assistance systems can be differentiated along two dimensions: the degree of intelligence and the degree of interaction. Basic user assistance systems are characterized by a low degree of interaction and intelligence, whereas advanced user assistance systems (AUAS) have a high degree of intelligence and/or a high degree of interaction. An example for a basic user assistance system is the F1 button that provides explanatory information when pressing it. A high degree of interaction means that AUAS are able to adapt its answers to the users' utterances, similar to human-human communication. A high degree of intelligence, on the other hand, means that the system is able to adapt its utterances to some context factors such as the users' current knowledge state.

SPAs can be characterized by a high degree of interaction as they can build up a simple dialogue with the user. Depending on the utterances of the user, SPAs are able to create a different answer. For example, SPAs are able to detect whether the answer to a question is right or wrong [10]. Moreover, by extending the primary systems with features that make them adaptive to their users and given context situations, SPAs can also be considered as intelligent.

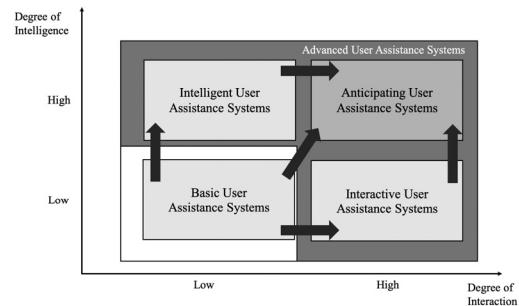


Figure 1. Classification of advanced user assistance system [1].

For example, SPAs are able to detect a user's position in a room [11]. For novice users to learn how to use a new system, it is important for them to first discover capabilities and limitations. Discoverability is a means to achieve learnability [12]. SPAs should help users to discover the capabilities of a new system. Additionally, SPAs are able to help people with some kind of disabilities (e.g., physical impairment) to use smart systems via voice [13].

2.2 Smart Personal Assistants

Smart Personal Assistants (SPAs) are computer agents that are able to assist users by engaging with them via natural dialogue [4]. SPAs have an agent program running on SPA-enabled devices (endpoints) such as Apple's iPhone, iPad, and Mac, Amazon's Echo or Google's Home. The main functionality, the "brain" of an SPA, 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, [14]). SPA providers offer rich ecosystems with intuitive interfaces that allow a large number of users to create their skills without having in-depth programming knowledge, thereby increasing SPA providers' own business value. SPAs can be divided into two types: (1) built-in SPAs that use multi-purpose devices and (2) stand-alone SPAs that use dedicated devices. Examples of built-in SPAs include Siri (for Apple products) and Cortana (for Windows-based PCs). Examples of stand-alone SPAs include Alexa (that uses Echo, Echo Dot, and Tab dedicated devices) and Google Assistant (that uses Google Home dedicated devices, [14]). In our paper, we concentrate on both types of SPAs.

2.3 Software Development

Software development is the process of conceiving, specifying, designing, programming, documenting, testing, and bug fixing involved in creating and maintaining applications, frameworks, or other software components [15]. There are many approaches

on how to create software known as software development life cycle models. Basically, we can differentiate between waterfall models and more recently famous agile software development approaches. Both approaches generally consist of some combination out of the following states: Analyzing the problem, market research, gathering requirements for the proposed business solution, devising a plan or design for the software-based solution, implementation (coding) of the software, testing the software, deployment; maintenance and bug fixing. SPAs can be considered as one special type of software that is known as a dialogue-based system. Compared to other types of software, the interaction process and the corresponding interaction logic are the main points to consider [16].

3. Research Methodology

We follow a Design Science Research (DSR) approach to develop SPAM. The design-science paradigm has its roots in engineering and the sciences of the artificial [17]. Design science can be considered as both a process (set of activities) and a product (an artifact,[18]). We rely on Hevner's [7] three cycle view to structure our design science research process (see Figure 2) and consider SPAM as our artifact.

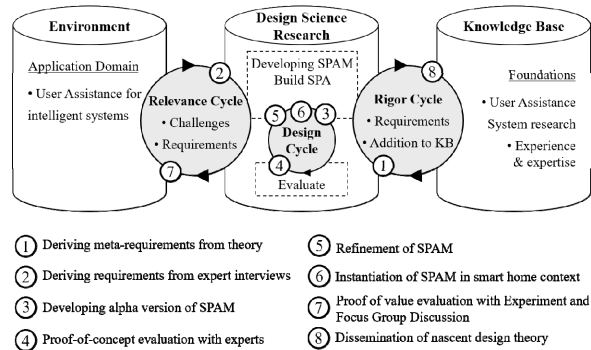


Figure 2. Research approach.

The *Relevance Cycle* connects the application domain of the research project with our design science activities. The *Rigor Cycle* makes sure that the design science activities consider the existing knowledge base of scientific foundations, experience, and expertise. The central *Design Cycle* iterates between the core activities of building and evaluating our design artifact [7]. The following step numbers refer to the number shown in Figure 2.

In step 1, we start the *Rigor Cycle* by deriving meta-requirements for SPAM from theory. In step 2, we start the *Relevance Cycle* by deriving requirements from expert interviews. We conducted semi-structured

interviews with SPA Designers from large-sized (Microsoft and IBM) and medium-sized tech companies (ABB, VoicePoint). In step 3, we started the *Design Cycle* by developing the alpha version of SPAM. In step 4, we conduct a proof-of-concept evaluation with the SPA designers from step 2. After refining SPAM in step 5, we instantiate the process model by developing an SPA for a smart home system (step 6) and evaluate it with the help of an experiment and a concluding focus group discussion (step 7). Last but not least, we discuss our results and disseminate a nascent design theory in step 8.

4. Developing a Smart Personal Assistant Process Model (SPAM)

In the following chapter 4, we will describe the design process of SPAM. Every subsection refers to the corresponding step visualized in Figure 2.

4.1 Step 1: Deriving Meta-Requirements from Theory

In this section, we gather meta-requirements from theory. As mentioned in the theoretical background section, SPAs can be considered as advanced user assistance systems. Thus, we search for requirements of developing SPAM in user assistance system research. We conducted a systematic literature review by vom Brocke [19] and Webster and Watson [20] in order to identify relevant papers to derive meta-requirements. We selected the following databases: "AIS Electronic Library", "ACM Digital Library", "IEEE Xplore Digital Library", "Science Direct", and "EBSCOhost Business Source Complete". Moreover, we covered the following keywords: "smart assistant", "smart user assistant", "conversational agent", "virtual assistant", "personal assistant", "assistance system" and "process model". The database search was constrained to title, abstract, keywords, and a publication period from 2012 to now. We excluded papers that are older than 2012, because our focus is to only include new emerging AUAS. Our literature search resulted in 776 hits. Titles, abstracts and keywords were screened to fit the purpose of the study. Based on that, we selected 36 papers. We included papers that describe the design of a specific SPA and excluded papers that provide a conceptual view on SPAs in general. We clustered similar requirements resulting in four requirement clusters. Table 1 depicts these clusters and the meta-requirements.

Table 1. Meta-requirements from Theory

| Source | Meta-requirements from Theory |
|--------|-------------------------------|
|--------|-------------------------------|

| | |
|---|---|
| Analyzing user needs [21] [22] | MRT1: SPAM should first start with analyzing the user in detail |
| Error handling [23] [24] | MRT2: SPAM should provide the developer guidance on how to include error handling mechanisms into the SPA so that the user will not get frustrated and can continue the interaction. |
| Modeling interaction process [25] [26] | MRT3: SPAM should guide the developer to model the interaction process that users should run through before technically implementing the SPA. |
| Build interaction model [27] [28] [29] | MRT4: SPAM should include guidance on how to create an interaction model |

Before a developer start to create the SPA technically, it is essential to analyze the user and its needs in detail which is also called a user-centered approach (MRT1). Moreover, SPAM should provide the developer guidance on how to include error handling mechanisms so that the user will not get irritated when the SPA does not understand them. For example, when the user asks the SPA a question regarding a specific function, the user should be guided in reframing the question if the SPA does not understand the intent (MRT2). Before it comes to the technical development of the actual SPA, it is further important that SPAM includes a step where the developer has to model the interaction process between the user and the SPA. This step helps the developer to think of possible users' utterances and corresponding SPA' answers. For example, the developer can design a decision tree to capture all possible dialogue variants (MRT3). Last but not least, SPAM should include some guidance on how to build an interaction model with the help of an SPA ecosystem. The developer needs to know how to integrate possible users' utterances as well as how to set the rules for SPA's answers (MRT4).

4.2 Step 2: Deriving Requirements from Expert Interviews

After defining meta-requirements from theory, we derive requirements from experts in the field of SPA development with the help of semi-structured interviews. The interview partners are experts in the field of smart personal assistants development with many years of experience. We chose to interview four experts from the programming department of

Microsoft, IBM, ABB and a Switzerland-based, medium-sized company called VoicePoint. We selected these interview partners to gather insights into how tech companies proceed when developing SPAs and gather transferable insights for SPAM. The interviews lasted about an hour each and were structured as follows: In the first part, we explained the experts our research project and asked them to describe their routine process when developing SPAs. In the second part, we asked them to specifically list and prioritize obstacles that have to be faced during the development. We transcribed the interviews and analyzed them using the method of user stories proposed by Cohn [30]. 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 desired functionality [30]. We coded and clustered the user stories with the help of a qualitative content analysis by Mayring [31]. The main idea of the qualitative content analysis is to look for interesting issues, derived from theoretical background and research question, which determines the aspects of the textual material taken into account. Finally, we translated the user stories into requirements for SPAM. The user-stories and the corresponding requirements are depicted in Table 2.

Table 2. User stories and requirements from experts

| User Stories (Experts) As an SPA developer, I | Requirements from Practice (RP) |
|--|---|
| US1: ... want to first define and clarify the use case with the stakeholder, so I can set the scope of the project. | RP1: The first step of SPAM should be: clarify the use case with stakeholder |
| US2: ... want to know what kind of needs the users have, so I can address the needs with the SPA functions. | RP2: There should be a step in SPAM where the needs of the user are analyzed. |
| US3: ... want to improve the SPA by analyzing errors and misunderstandings continuously. | RP3: At the end of SPAM, a step should be included that covers the continuous improvement of the SPA. |
| US4: ... want to determine if it should be a text-based or voice-based SPA, because the following design decisions are different. | RP4: SPAM has to take in consideration that at one point the user has to choose between voice or text. |
| US5: ... want to test the interaction with smart personal assistant in different loops with | RP5: SPAM should include a user testing phase. |

| | |
|---|--|
| different people, so I can be sure that it works when it goes life. | |
| US6: ... want to test the technical functions of the smart personal assistant, so I can be sure that the functions are running reliably. | RP6: SPAM should include a technical testing phase. |
| US7: ... want to model the interaction process that a user runs through, so I can be sure that I address the most important user needs. | RP7: SPAM should include a process modeling of the interaction process. |
| US8: ... want to know what the user will say; so I can capture most of the different kinds of intents. | RP8: SPAM should include a step to collect different kinds of intents. |
| US9: ... want to build a proof of concept and test it, so I can make a decision if the use case really works. | RP9: SPAM will contain a phase where the proof of concept is built and a decision is taken. |

US10:... want to improve and finetune the proof of concept and expand it to a sellable product.

RP10: SPAM will consist of a step where the SPA will be finetuned.

4.3 Step 3: Developing Alpha Version of SPAM

Based on the requirements from theory and practice, we create our first version of SPAM (see Figure 3).

In order to develop SPAM, we followed the definition of reference (process) models proposed by Roseman and Aalst [32]:

“Reference models are generic conceptual models that formalize recommended practices for a certain domain. They are often labelled with the term «bestpractice» reference models and claim to capture reusable state-of-the-art practices.”

Thus, our aim is to build a reference process model that can be reused in different but similar application

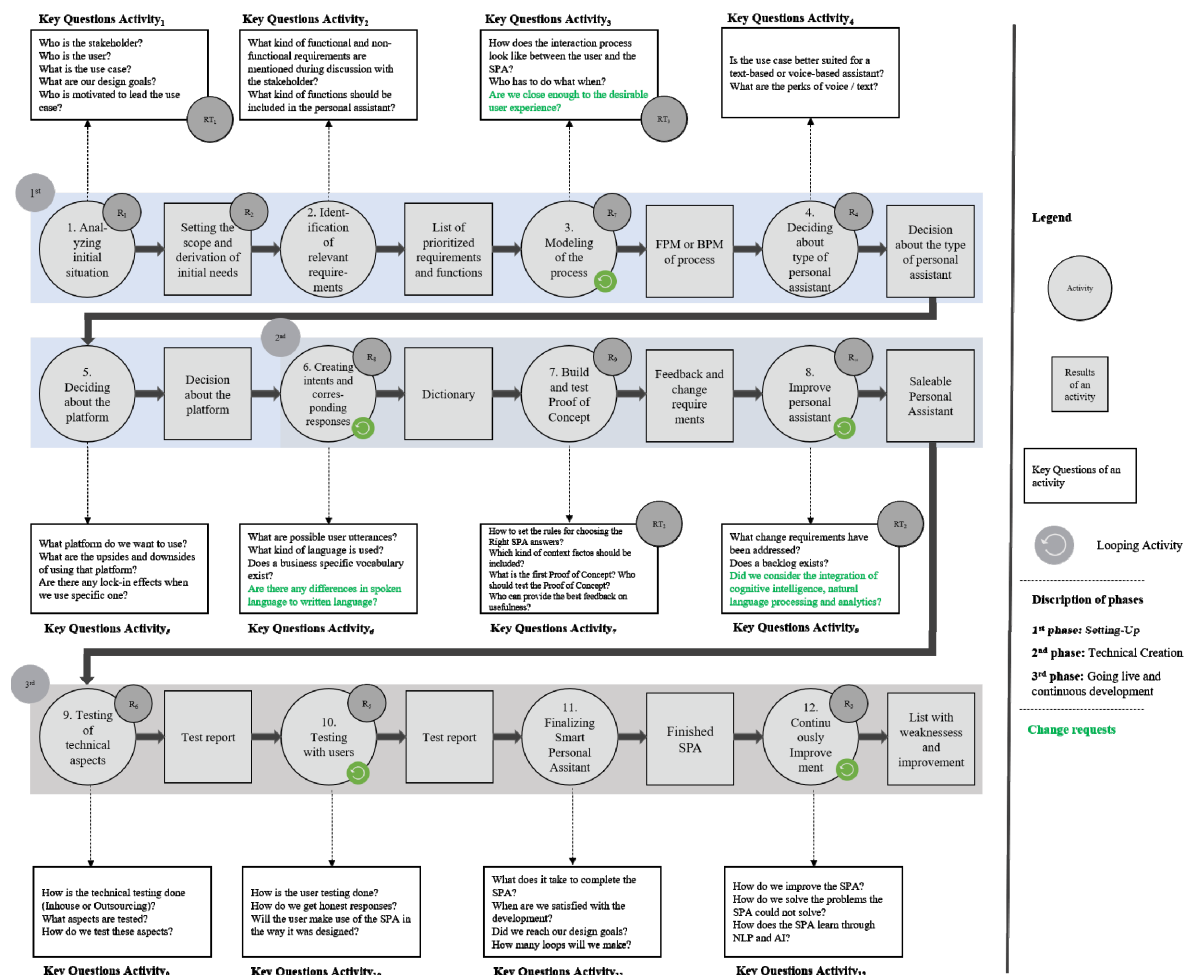


Figure 3. SPAM Method including Key Questions and Change Requests.

scenarios to improve quality of SPAs and to save costs and/or time for SPA developers.

When designing SPAM, we tried to bring the requirements from theory and practice into a sense-making order. We sequentially ordered the development activities and have named the results after each development activity. The development activities are represented in grey circles and the development products in grey rectangles. Moreover, we added key questions to an activity to further guide the developer. The key questions are depicted in white rectangles. In the remainder of this chapter, we will explain every phase in detail.

1. Phase: Setting-up

In step 1, the developer first analyzes the initial situation. In order to have a clear understanding of what is expected, the use case should be defined precisely in corporation with the stakeholder (e.g., client) or the end-user. If the end-user does not participate in the clarification of the use case, it is important to keep in mind what he wants and what his needs are. Moreover, it is essential to identify motivated people that are willing to help with the use case and in later stages with the testing. As a result, the developer can set the scope of the project and derive the needs of the user from this step.

In step 2, the developer should derive functional and non-functional requirements based on step 1. This derivation can happen collaboratively with the users or alone. Moreover, the developer should prioritize the most important requirements in order to be able to focus on the most important future steps. Consequently, the developer should derive a list of prioritised requirements. In step 3, the interaction process between the user and the SPA should be modeled. Independently from the use case, the process can be modeled with different kinds of modeling languages such as BPMN 2.0 or FPM 2.0. The modeling of the interaction process helps to detect possible user utterances and corresponding SPA answers. In step 4, depending on the requirements, the developer should decide whether he wants to create a text-based or a voice-based SPA. Moreover, he can decide whether to create a built-in or a standalone SPA as mentioned in the theoretical background section. In step 5, the decision about the preferred SPA development platform has to be made. Depending on the requirements, some platforms are better suited than others.

2. Phase: Technical Creation

In step 6, before the developer can start with the actual coding of the SPA, he needs to identify the dictionary that is used for the use case. The dictionary consists of possible user utterances and SPA responses. Dependent on the use case, the dictionary might have

different specifications, e.g., business specific. The dictionary can later be integrated into the interaction model of the SPA. With the help of machine learning techniques, the SPA is able to understand users' intentions and can choose the right answer.

Moreover, it is important to note that the users' use of language might differ from the developers one, especially in a business-specific context. In step 7, after creating the dictionary, the SPA can be built. The software developer should also consider which context factors to include in the code itself. For example, he should remember which steps have already been explained to the user, so that he does not always start from the beginning. As soon as it exists on a draft, it should be tested, preferably with an external user. If the user is not available, internal testing can be done. With the given feedback and change requirements, the developer can iteratively improve the SPA in step 8. Finally, a ready-to-market SPA should result in the end.

3. Phase: Going live and continuous development

The third and last phase of SPAM starts with the testing of the technical aspects in step 9. Depending on the expertise of the SPA developer, this step might even be automated. At any case, test documents might be used to examine the functionalities of the SPA. After doing so, a test report should be generated. This helps to understand the flaws and to improve the SPA in the future. In step 10, alongside the technical testing, the SPA should be tested with potential users. Again it is advisable to test the SPA with the end-user to receive meaningful feedback. After running the tests with users, a test report will be produced. With the test report of the technical testing and user testing, the SPA can be completed in step 11. The final step 12 is to continuously improve the SPA. Summing up, these steps should help and guide SPA developer to systematically structure the design of SPAs in order to improve quality and save time and costs.

4.4 Step 4 and 5: Proof-of-concept evaluation with Experts and Refinement of SPAM

According to Sonnenberg and vom Brocke [33], it is important to direct the foci of evaluations on two aspects: (1) the constituents of the artifact and the design decisions taken as well as on (2) the evaluation of the usefulness of the artifact [33]. The first evaluation of SPAM concentrates on the constituents of the artifact and our design decisions. The purpose of the evaluation is to ensure the completeness and the correctness of the different steps as well as to clarify if the expert would use SPAM for the development of their SPAs. We asked the same experts as in step 2 individually. First, we showed them SPAM and explained every step. Second, we asked them to note

their change requests for a further improvement of SPAM. The following Table 3 shows the change requests and how we addressed them.

Table 3. Change Requests and how we addressed them

| Change request | How we addressed them |
|---|----------------------------|
| CR1: Focus more on the building of the interaction model | Included as a key question |
| CR2: Try to display the iterations between building and testing | Included as a“loop” symbol |
| CR3: Migration into other systems | Included as a key question |

The change requests are already included and highlighted in green in Figure 3.

4.5 Step 6: Instantiation of SPAM in Smart Home Context

We instantiated our artifact with the development of an SPA for a smart home system. We chose a smart home context, because these systems have made a great technological progress in recent years and, thus, can be considered as intelligent system. They offer many functions such as controlling lighting, climate, kitchen equipment, entertainment systems, and other appliances. The scenario tries to cover most of the common functions of smart home systems. We created the SPA based on the following scenario:

Table 4. Scenario for Instantiation of SPAM

| Smart Home Task |
|---|
| <p><i>Imagine the following scenario:</i></p> <p>You are expecting guests and want them to feel comfortable in your home. During preparing the meal, you notice that you forgot to buy wine and would have to leave to buy some in the wine shop. A run to the shop and back would take you 20 minutes. Unfortunately, your guests will arrive in about an hour. A glance at the weather app reveals that it might rain in the meantime, so you should also pull down the shutters. Further, you are in the middle of preparing the dinner; the meatloaf would take at least 15 minutes. You decide to go buy the wine and manage the rest of the tasks with your new Smart Home system. Try to find out how you can use your Smart Home System to conduct these tasks!</p> |

1. Phase: Setting-up

In step 1 (analyzing initial situation), the stakeholder has to be identified. New smart home system users are our target audience. They will use the SPA for getting to know the smart home system. Moreover, we chose to build a use case around the introduction of smart home functions to a new smart home owner. The goal of the SPA is to help owners to get to know the most important functions of their smart home system.

In step 2 (identification of relevant requirements), we identified the relevant requirements of the SPA. In order to gather the requirements and pain points of a smart home system owner, we identified a list of problems in consultation with smart home users. The requirements were consolidated into different groups. For example, one of the pain points is that the used vocabulary in manuals is often not easy to understand and not comprehensible. The next step was to look at the current solution of smart home systems (a manual) and identify the different functions as well as the current flaws in the manual. Pairing up the flaws of the current solution and the requirements from a user's perspective, we defined our final set of requirements.

In step 3 (modeling the process), we modeled the interaction process from a user perspective. Due to the context of smart home and the requirements we had, the choice of the personal assistant type was rather straight forward. Most of the time, users cannot use their hands while using smart home functions (e.g. while cooking). For example, when they want to program their oven, they would prefer to speak to the SPA instead of typing. We chose to build a voice-based SPA in step 4 (deciding about type of personal assistant). Based on this decision, we choose Amazon's Alexa as SPA platform in step 5 (deciding about the platform). Specifically, we used Alexa's Skill Development Kit 2.0 with nodeJS. This framework seems to offer one of the most advanced state-of-the-art capabilities regarding speech recognition and natural language processing. Additionally, the Alexa Skill Kit 2.0 provides a variety of blueprints for the smart home context [34].

2. Phase: Technical Creation

With the help of the manual and the discussions with the smart home owners, we created a dictionary including intents and SPA responses in step 6 (creating intents and corresponding responses). In step 7 (build and test Proof of Concept), we built the SPA with the help of an Alexa blueprint that allows to add some basic coding patterns to create an interactive experience [34]. In this step, we first added the user intents (one intent consists of several user utterances) and set the rules for the back-end (What should the SPA answer?) The proof of concept was tested internally in step 8 (improve personal assistant). Afterwards, we made some technical adjustments in step 9 (testing of technical aspects) before the SPA development can entered the next phase.

3. Phase: Going life and continuous development

In step 10 (testing with users), we tested the technical aspects with the help of Amazon's developer testing function. We created a test report listing the main change requests. In step 11 (finalizing smart personal assistant), we tested the SPA with two

possible users. In order to display the dialogue of the designed SPA, an exemplary dialogue is represented in the following Figure 4.

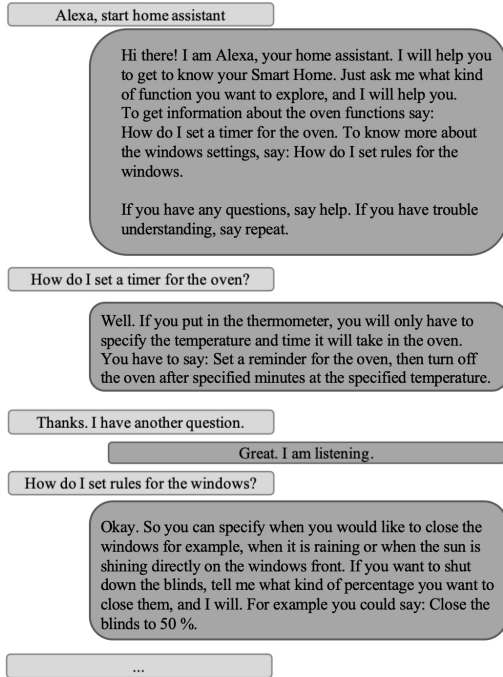


Figure 4. An example of a dialog between the user and the SPA

4.6 Step 7: Proof-of-Value Evaluation with Experiment and Focus Group Discussion

The second evaluation focused on the usefulness of the SPA with the help of an experiment and a concluding focus group discussion to receive more insights into the usefulness of the SPA.

We set up an experiment with 32 participants where half of the participants used an SPA device (Amazon's Alexa Echo Dot Device) for solving the scenario depicted in Table 4 and half of the participants used the already existing manual of the smart home system. The manual had four pages and contained the same information as the one that was installed on Alexa. The participants are homeowners or loaners that think about implementing a smart home systems but do not own a smart home system yet. At the beginning of the experiment, each of the participants had to fill out a pre-test. In this pre-test, the age, the gender as well as the personal experience with similar smart home systems were requested. It was essential to the experiment that the participants did not have prior experience with similar smart home systems. The average age of the experiment group was 37.56 years, with 10 males and 6 females. The average age of the control group was 34.81 years, with 8 males and 8 females. An ANOVA test was conducted to make sure

that the two groups were equal ($p < 0.05$). The test revealed that there was no significant difference in the age of the participants, gender or personal experience with smart home systems.

Both groups had 20 minutes to learn the required functions of the smart home systems and how they would use them to solve the task. The experiment group interacted with Alexa to get information about the functions whilst the control group had time to read and memorize the manual. Alexa started with an onboarding message and then tried to answer participants' questions. After step 2, the participants had to conduct a post-test to check their knowledge about the smart home system. The post-test questions were the same for both groups. The post-test consists of questions to solve the depicted scenario (e.g. how to set the timer for the oven to 15 minutes?). Furthermore, they were asked about the satisfaction with the SPA respectively the manual.

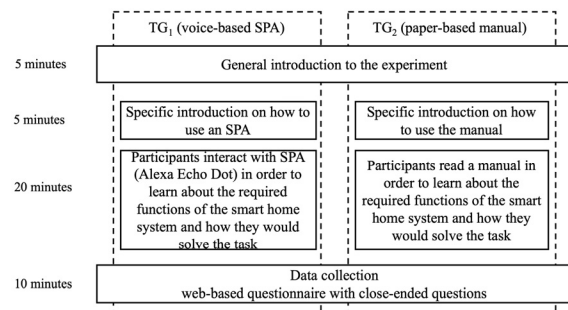


Figure 5. Experimental Procedure

Five persons from the experiment group participated in the focus groups discussion afterwards. The facilitator was one of the researchers. The focus group discussion was structured as follows. In the first part, one of the researchers introduced the goal of the group discussion. Afterwards, we asked the participants to reflect about the positive and negative experiences with Amazon's Alexa instead of a traditional manual.

In the following, the results of the field experiment, as well as the focus groups discussion, are presented.

Table 5. Quantitative Results of the Experiment

| | Group: Alexa | Group: Manual |
|------------------------------|--|--|
| Mean of correct answers | On average, 4.00 out of 5 questions were answered correctly. | On average, 3.31 out of 5 questions were answered correctly. |
| Satisfaction with the manual | 4.2/5 | 3.7/5 |

As shown in Table 5, people who could solve the task with Alexa achieved better results in the post-test than those who had to solve the task with the manual. Moreover, the *Alexa* group was more satisfied in solving the task than the *Manual* group.

After using the SPA as a smart home user assistance, we discussed its use with the participants in the form of a focus group discussion. Most of the participants preferred the interaction with Alexa compared to traditional user manuals. One reason for that is the convenient use via voice especially when they cannot use their hands. Moreover, they positively emphasized the possibility that they were able to ask specific questions without having to search for every function in the manual. Furthermore, they liked to ask again for a better explanation of the functions. Some participants mentioned that an SPA can be an advantage if you directly want to try out the intelligent system (e.g. pull down shutters). Even though most participants stated that they would like to use the SPA for similar tasks in the future, some participants in the focus group discussion also mentioned some negative aspects. Some participants said that the interaction with the SPA felt unnatural and weird to them. Despite various advantages, some participants express concerns about privacy. For example, they mentioned that there are videos on platforms such as YouTube showing SPA devices constantly tracking and recording conversations.

5. Discussion & Conclusions

The goal of this paper was to a) examine what requirements and processes should be considered when designing Smart Personal Assistants as advanced user assistance systems and b) to what extent our proposed method and the resulting Smart Personal Assistant increases a users capability of understanding a smart home system. To design our artifact, we used a design science research approach. Our artifact (SPAM) was derived from the requirements of scientific literature and from expert interviews. The results of our experiment have shown that with the help of SPAM, we were able to create an SPA that helps users to better get to know a smart home system. In the remainder of this section, we describe our contributions to research and practice, as well as our study's limitations and directions for future research.

5.1 Contributions to research

Our work makes two main contributions to research. Whilst many researchers in the field of user assistance research have tried to implement specific SPAs for intelligent systems [28, 29], to the best of our knowledge, transferable insights on how to design SPAs are missing. We contribute to research by proposing a method that helps to systematically design SPAs in a more effective and efficient way. This helps future researchers to create SPAs and further investigate the usefulness of these systems in different

use scenarios. Second, SPAM provides a stronger basis for researchers to report on already existing, alternative reference process models of similar areas and compare and contrast them with SPAM.

5.2 Implications for Practice

Our work has also some implications for practice. With the help of SPAM, SPA designers can use this reference process model as a general guidance to create SPAs for intelligent systems. Moreover, our proposed key questions help them to better address SPA-specific development activities. This reference process model should help SPA designers to improve the development process thereby saving costs and time.

5.3 Limitations & Future Research

A number of limitations have to be considered with respect to our study. First, the list of requirements are derived from a specific field of research and from a certain selection of experts. It is likely that the requirements would be different if we have used different theoretical perspectives and interview partners. However, we tried to select the most relevant research field and a representative sample of SPA designers as interview partners. Second, the evaluation of SPAM was made in a quite narrow context (e.g. smart home system). In order to validate the usefulness and accuracy of the proposed method, further evaluations for other types of smart systems are needed. For example, SPA developers can be instructed to use SPAM to create an SPA for different smart system. This way, we could make sure that SPAM is valuable for practitioners. Future research should focus on this kind of evaluations in order to further develop SPAM and challenge it with already existing process models in similar fields. Third, SPAM does not explicitly take into account the needs of all individuals. For example, SPA development approaches for people with disabilities might look different. Moreover, participants also mentioned some negative aspects about using SPAs, such as privacy concerns. Future research should try to come up with solutions and recommendations to deal with that. Last but not least, future research should focus on different personal assistant traits and which of these traits affect the trust and comfort of different kinds of users.

References

- [1] A. Maedche, S. Morana, S. Schacht, D. Werth, and J. Krumeich, "Advanced User Assistance Systems," *Bus. Inf. Syst. Eng.*, vol. 58, no. 5, pp. 367–370, Oct. 2016.
- [2] T. Ann Sykes, "Support structures and their impacts on employee outcomes: A longitudinal field study of an enterprise system implementation," *MIS Q.*, vol. 39, no. 2, 2015.
- [3] L. F. Barrett, M. M. Tugade, and R. W. Engle,

- "Individual differences in working memory capacity and dual-process theories of the mind.," *Psychol. Bull.*, vol. 130, no. 4, p. 553, 2004.
- [4] S. Pais, J. Casal, R. Ponciano, and S. Lourenço, "Unsupervised assistive and adaptive intelligent agent in smart environment," presented at the Conference: ICIES, 2015, pp. 23–24.
- [5] F. Honold *et al.*, "Companion-technology: towards user- and situation-adaptive functionality of technical systems," presented at the 2014 International Conference on Intelligent Environments, 2014, pp. 378–381.
- [6] A. Abdolrahmani, R. Kuber, and S. M. Branham, "Siri Talks at You: An Empirical Investigation of Voice-Activated Personal Assistant (VAPA) Usage by Individuals Who Are Blind," presented at the Proceedings of the 20th International ACM SIGACCESS Conference on Computers and Accessibility, 2018, pp. 249–258.
- [7] A. R. Hevner, "A three cycle view of design science research," *Scand. J. Inf. Syst.*, vol. 19, no. 2, p. 4, 2007.
- [8] S. Gregor, "The nature of theory in information systems," *MIS Q.*, pp. 611–642, 2006.
- [9] S. Gregor and A. R. Hevner, "Positioning and presenting design science research for maximum impact," *MIS Q.*, pp. 337–355, 2013.
- [10] S. Lovato and A. M. Piper, "Siri, is this you?: Understanding young children's interactions with voice input systems," presented at the Proceedings of the 14th International Conference on Interaction Design and Children, 2015, pp. 335–338.
- [11] Y. Sumi, T. Etani, S. Fels, N. Simonet, K. Kobayashi, and K. Mase, "C-map: Building a context-aware mobile assistant for exhibition tours," in *Community computing and support systems*, Springer, 1998, pp. 137–154.
- [12] F. Chen, *Designing human interface in speech technology*. Springer Science & Business Media, 2006.
- [13] T. Gentry, "Smart homes for people with neurological disability: State of the art," *NeuroRehabilitation*, vol. 25, no. 3, pp. 209–217, 2009.
- [14] H. Chung, M. Iorga, J. Voas, and S. Lee, "Alexa, can I trust you?," *Computer*, vol. 50, no. 9, pp. 100–104, 2017.
- [15] K. Schwaber and M. Beedle, *Agile software development with Scrum*, vol. 1. Prentice Hall Upper Saddle River, 2002.
- [16] M. F. McTear, Z. Callejas, and D. Griol, *The conversational interface*, vol. 6. Springer, 2016.
- [17] H. A. Simon, *The sciences of the artificial*. MIT press, 2019.
- [18] R. H. Von Alan, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Q.*, vol. 28, no. 1, pp. 75–105, 2004.
- [19] J. Vom Brocke, A. Simons, K. Riemer, B. Niehaves, R. Plattfaut, and A. Cleven, "Standing on the Shoulders of Giants: Challenges and Recommendations of Literature Search in Information Systems Research.," *CAIS*, vol. 37, p. 9, 2015.
- [20] J. Webster and R. T. Watson, "Analyzing the past to prepare for the future: Writing a literature review," *MIS Q.*, pp. xiii–xxiii, 2002.
- [21] S. Ghosh and J. Pherwani, "Designing of a natural voice assistants for mobile through user centered design approach," presented at the International Conference on Human-Computer Interaction, 2015, pp. 320–331.
- [22] H. Winner, S. Hakuli, F. Lotz, and C. Singer, *Handbook of driver assistance systems: Basic information, components and systems for active safety and comfort*. Springer, 2016.
- [23] Y. R. Guo and D. H.-L. Goh, "Affect in embodied pedagogical agents: Meta-analytic review," *J. Educ. Comput. Res.*, vol. 53, no. 1, pp. 124–149, 2015.
- [24] G. Kulkarni, D. Nagaraj, and S. Gupta, "System and method for smart error handling mechanism for an application," Oct. 2017.
- [25] C. Pearl, *Designing Voice User Interfaces: Principles of Conversational Experiences*. O'Reilly Media, Inc., 2016.
- [26] M. Luria, G. Hoffman, B. Megidish, O. Zuckerman, and S. Park, "Designing Vyo, a robotic Smart Home assistant: Bridging the gap between device and social agent," presented at the 2016 25th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), 2016, pp. 1019–1025.
- [27] G. Campagna, R. Ramesh, S. Xu, M. Fischer, and M. S. Lam, "Almond: The architecture of an open, crowdsourced, privacy-preserving, programmable virtual assistant," presented at the Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 341–350.
- [28] A. Sato, K. Watanabe, and J. Rekimoto, "MimiCook: a cooking assistant system with situated guidance," presented at the Proceedings of the 8th international conference on tangible, embedded and embodied interaction, 2014, pp. 121–124.
- [29] K. Tsujino, S. Iizuka, Y. Nakashima, and Y. Isoda, "Speech Recognition and Spoken Language Understanding for Mobile Personal Assistants: A Case Study of" Shabette Concier", presented at the 2013 IEEE 14th international conference on mobile data management, 2013, vol. 2, pp. 225–228.
- [30] M. Cohn, *User stories applied: For agile software development*. Addison-Wesley Professional, 2004.
- [31] P. Mayring, "Qualitative inhaltsanalyse," in *Handbuch qualitative Forschung in der Psychologie*, Springer, 2010, pp. 601–613.
- [32] M. Rosemann and W. M. Van der Aalst, "A configurable reference modelling language," *Inf. Syst.*, vol. 32, no. 1, pp. 1–23, 2007.
- [33] C. Sonnenberg and J. vom Brocke, "Evaluations in the Science of the Artificial – Reconsidering the Build-Evaluate Pattern in Design Science Research," in *Design Science Research in Information Systems. Advances in Theory and Practice*, 2012, pp. 381–397.
- [34] M. Doring, "Build Skills Faster with Version 2 of the ASK Software Development Kit for Java," 29-Mar-2018. .
- [35] D. Griol, J. Carbo, and J. M. Molina, "A statistical simulation technique to develop and evaluate conversational agents," *AI Commun.*, vol. 26, no. 4, pp. 355–371, 2013.
- [36] D. C. Derrick, J. L. Jenkins, and J. F. Nunamaker Jr, "Design principles for special purpose, embodied, conversational intelligence with environmental sensors (SPECIES) agents," *AIS Trans. Hum.-Comput. Interact.*, vol. 3, no. 2, pp. 62–81, 2011.