Proceedings of the 54th Hawaii International Conference on System Sciences | 2021

Practical Challenges of Virtual Assistants and Voice Interfaces in Industrial Applications

Marco Gärtler
ABB Corporate Research Center Germany
marco.gaertler@de.abb.com

Benedikt Schmidt
ABB Corporate Research Center Germany
benedikt.schmidt@de.abb.com

Abstract

Virtual assistant systems promise ubiquitous and straightforward access to information, applications, and physical appliances. Their foundation on intent-oriented queries and support of natural language makes them an ideal tool for human-centric applications. The general approach to build such systems and the main building blocks are well-understood and offered as off-the-shelf components. While there are prominent examples in the service sector, other sectors such as the manufacturing and process industries have nothing comparable. We investigate the practical challenges of building a virtual assistant using a representative and simplified case from of knowledge retrieval domain. A qualitative study reveals two significant obstacles: Firstly, while systems can be tuned to tackle the most relevant types of question tasks, user acceptance is lower than for comparable GUI-based systems and, secondly, a disproportional amount of effort to get all details and having a robust system. Overall, implementing a virtual assistant for an industrial application is technically feasible yet requires significant effort and understanding of the target audience.

1. Introduction

Virtual assistants (VA) and, more generally, Voice User Interfaces (VUIs) spread over the last decade are ubiquitously present today and continue to increase their presence in our everyday life. In 2019 more than 140 million smart speakers were sold [31] and every major smartphone platform comes with at least one virtual assistant. In smartphones and other mobile devices, services such as Siri (by Apple), Google Assistant (by Google), or Cortana (by Microsoft) are constant companions. There are the standalone counterparts in our homes, such as Alexa or Amazon Echo (by Amazon) or Google Home (by Google). In all those cases, the products aim to provide simple, easy, and intuitive access to basic functionality, like creating alarms or notifications. They offer convenience in interaction with other services such as home automation. More generally, commercial services and open source components to deal with text-to-speech, speech-to-text, and intent recognition are readily available. Examples are IBM Watson[1], Cognitive Services [22] by Azure or Amazon Transcribe[2], and Amazon Polly[3] by AWS.

The topic of VAs and VUIs has been studied fairly intensively. Starting with dialog systems in natural language [32], extending to spoken language and integration of technical services in [5, 26, 17, 15] to more design-driven and holistic views in [16, 14, 20, 8, 18, 3, 19]. There are many tutorials [21, 1, 23, 27, 4] promising the easy creation of such systems. Among them are several companies that use them to advertise their components and services. Looking at the areas of these VAs, there is a clear focus in the private environment and the service sector. The everyday aspects of VUIs and the implication for design in a private environment have been studied by Porcheron *et al.* in [29]. There is a surprising lack of attempts and success stories in other industrial sectors. Schmidt *et al.* in [30] discuss opportunities of VAs focusing on operation support in the process industry and identified related challenges and opportunities. Yet, there are few attempts to build such more industry-oriented systems.

The research question addressed in this paper is – why are VA systems limited to the private environment and customer service? What are the potential reasons for this? We investigate the degree to which state of the art VA systems can support work tasks in the industrial domain. While customization of a ready-made framework would be most suitable, the existing frameworks show vendor-specific limitations in the design of intents and the communication with backend services. Therefore, we decided to develop a VA system using off-the-shelf components with a higher

[1]https://www.ibm.com/watson
[2]https://aws.amazon.com/transcribe/
[3]https://aws.amazon.com/polly/

degree of freedom regarding customizability.

The foundation section provides a specification of VA systems and work tasks. Following studies of information work and cognitive information processing, we derive a four-level hierarchy of task complexity addressed by a VA system for the industrial domain. The VA architecture section 3 specifies the typical components shared by most VA systems. To investigate work task coverage for the industrial domain, we built a VA system, addressing questions on all four complexity levels to be compared to alternative software solutions (i.e., Excel, Business Intelligence Application). We expect this system to be a blueprint for analytics-oriented VA systems. Our proof of concept uses the following building blocks: a component for speech-to-text, a component to recognize intents, a simple analytic service, and a component for text-to-speech. Besides the analytical service, we use only readily available components and services.

The evaluation asked the participants to perform information extraction tasks, using standard tools and the VA. The tasks were aligned with the four complexity levels and included extraction of plain values, aggregations, or comparisons on a large database (Section 4). The study has been designed to reflect activities in the industrial domain, like monitoring process variables in different regions and comparing them to historic values or other process variables. The setup and the first results of a qualitative evaluation are shown in Section 5. The section also contains insights into the acceptance of users and lessons learned during development.

2. Foundations

VAs or VUIs typically are intent-oriented support systems that make use of an infrastructure of digital services. Intent-orientation basically stands for a system that targets the fulfillment of user intents. Intent-orientation is an obvious goal for every type of user interface. Still, for most applications, goal achievement requires to take a substantial amount of decisions by interacting with the UI. Formatting a text in a word processor is an example – despite just saying "make it look like a letter", it is required to execute multiple formatting steps. For intent-oriented interfaces, the intent is expressed in natural language, and the number of interaction steps required for fulfillment is minimized. This minimization is achieved by removing options or by deriving required information from contextual information sources.

Today's VA systems build on three foundations: 1) *Augmentation of the human intellect:* The idea of high-powered electronic aids embedded in everyday life and support in solving problems [10], 2) *Conversational agents, dialog systems:* Systems capable of using natural language as user interface, to extract intents from natural language and establish a context understanding from a dialog, 3) *Service growth:* an increasing number of small and focused services made available to large user groups – as an effect, the number of services used per user is steadily increasing (e.g., app downloads, number of apps available). The combination of these elements is the state of the art of VA systems. A highly personalized multi-modal system with a strong focus on speech that is available on multiple devices and integrates contextualized and customized access to numerous digital services.

A first concrete vision was established in 1987 in the Apple commercial for the "Knowledge Navigator" [4]. Since then, various research prototypes were developed and evolved into commercial products, like the Calo assistant (Cognitive assistant that learns and organizes) [2], which became Apple's Siri product.

While the commercial systems available are comparable concerning their components and the application domain – speech recognition/synthesis, natural language understanding, intent model with service integration for consumer domain – their actual capabilities still differ. For example, an investigation of the question-answer capabilities showed that Google and Alexa perform considerably better when confronted with general knowledge questions [7].

Still, general knowledge is not the actual core of the systems. To investigate the typical applications which are realized with VA systems, the basic skills and skill extensions for Alexa and the Google assistant have been reviewed[5]. As the latter offers an open eco-system that allows third parties to provide their own abilities, these were selected. In general, it shows that one can distinguish between skills that focus on knowledge access and those which focus on triggering functionality. An example of knowledge access is the response to a question based on a knowledge base like Wikipedia. An example for triggering a functionality is the assistant's use as remote control for the heating system in the house.

Typical usage scenarios include the following:

- Knowledge-based systems, to query structured knowledge bases
- Remote controlling devices or applications (e.g., radio)

[4]https://en.wikipedia.org/wiki/Knowledge_Navigator

[5]https://www.amazon.com/alexa-skills/, https://assistant.google.com/explore

- Media access
- Gaming, fun
- Location-based services, orientation
- Self-organization
- Stub to arbitrary services

Beyond the consumer domain, VA systems are rarely used. The exception is hands-free interaction, like integrating Microsoft's Cortana VA into the Hololens Augmented Reality Headset. Information work still relies on graphical user interfaces used with a combination of mouse, keyboard, and touch. We follow the perspective of the technology acceptance model and expect that perceived usefulness and ease-of-use are major influence factors for VA adoption [9]. Following this, VAs success for information work depends on the degree to which people believe that they increase job performance. The lack of adoption for information work can be related to an insufficient capability to optimize the respective work activities.

A review regarding the application of VAs in the industrial domain was conducted by [30]. They highlight the potential of making a heterogeneous landscape of digital services accessible to a workforce. Especially the situation-specific selection of services and the usage orchestration in a proactive manner are mentioned. Another highlighted aspect is the interaction modularity. Voice user interfaces allow easy context switches. Although information comes from different sources, voice interfaces give a continuous interaction frame which does not change. The review does not investigate the type of information required for work execution. Analysis of information work execution [25] and work that involves controlling complex systems – frequently focusing air traffic controllers [6] – show the critical role of providing access to specific information and simplifying its comprehension based on the situation and job requirements. An example is a business intelligence application with charts. The information worker will select a suitable chart for his question and will try to read out the response from the very rich information every chart contains.

An example for the industrial domain. An operator in a process plant monitors the production process of a batch. The VA is connected to an anomaly detection system, sensors on pipes, and the control system. The VA informs the operator that some production parameters are anomalous, based on the information from the anomaly detection system. The operator asks the system to show the anomaly, then to explain the characteristic of the anomaly. The system summarizes the anomaly characteristic. The anomaly is connected to the flow of material. The system recommends the usage of additional flow sensors to spot the exact location of the error. The operator acknowledges. Next, the system shows the sensor output over the production process. The operator spots the area of the error and asks the system whether an alternative route for the material exists. Based on the control system data, the VA recommends an alternative routing of the material as a preliminary fix of the error.

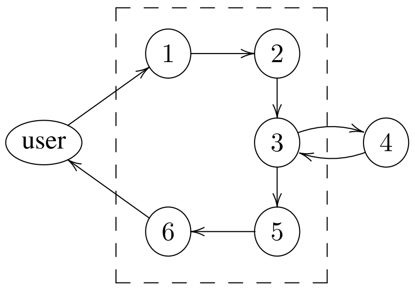
Along this line, we assume that VAs are only successful if they support access to situation-specific information and related cognitive comprehension at least as useful as existing solutions. To investigate this, we will focus on reading out information from a large database with temporally structured information, like sensor values over time. This domain is typically taken over by chart visualizations used in control systems and business intelligence systems.

A systematic evaluation has been made by [28]. He shows basic processes of information extraction, which are applied to graphs. Size, position, grouping, shape, and connectedness combined with scales and text are optimized to read out information. The usefulness of charts lies in their ability to provide information that solves information work tasks. Considering the workplace of a control room operator in a chemical plant, we see a good fit for the mentioned tasks. Therefore domain examples are provided. The tasks are: 1) reading out single values from a coordination system (e.g., the current reading of a sensor), 2) operations on values and trend (e.g., trend of one or multiple signals), 3) value identification criteria (e.g., all points in time, when temperature exceeded the threshold during the last shift), 4) comparing (e.g., how did the sensor trajectory change after modification of the batch recipe).

This paper will investigate the coverage of tasks belonging to the mentioned four levels and the acceptance of such a system. The studied system will be configured to address the mentioned task levels: 1) reading out value, 2) simple operations on values, 3) identification of values by criteria and 4) comparison of values. An evaluation has been conducted to 1) assess whether a state of the art VA covers the mentioned four levels of information access, 2) whether comprehension of the user is good enough to outperform existing solutions with classic chart-based interfaces.

3. Virtual Assistant Architecture

For our proof of concept, we consider VAs that, like Siri (by Apple), Google Assistant (by Google), or Cortana (by Microsoft), are independent services but do not require a dedicated device. More precisely, the user should be able to freely interact with the assistant without interfering with the usual activities. The requests are issued in natural language, either spoken or typed; similarly, the response is always given in natural language, both as text and speech. We use the tutorial from Rasa [27] as inspiration and loosely follow.



Step	Description
1	Pre-Processing, e.g., speech-to-text
2	Intent Recognition
3	Delegation to and execution of specific services for intents
4	Invocation of local services
5	Augmentation of results
6	Post-Processing, e.g., text-to-speech

Figure 1: General Interaction with a Virtual Assistant (dashed boxed).

The general process is shown in Figure 1. The user sends a request to the VA. This can be a text command, a spoken command, or a trigger associated with such a command. The request is pre-processed, which includes the conversion of speech and ends in a normalized text. This text is parsed. Intents and associated parameters are identified. In the configuration, one or more services are linked to each intent. The corresponding services are executed, which might invoke further services of the hosting infrastructure. Once the results are obtained, they are first augmented with local information, second post-processed, which includes the transformation back to natural language and the creating of a suitable audio output. Finally, everything is bundled in the response. While this description is given in a synchronous and blocking way, it can also be implemented asynchronously, i.e., the request to the VA does not block the interaction with other applications or even the assistant itself. This flow can also easily be extended to include services to trigger proactive information delivery for the user.

Table 1: List of components tested in our proof of concept.

Step	Component	Open Source	Cloud Service
1, 6	Azure Cognitive Services (ACS) - Speech		x
1, 6	Web Speech API[a]	embedded in browser	
1	Mozilla DeepSpeech[b]	x	
1	Amazon Transcribe[c]		x
1	Google Cloud Speed-to-Text[d]		x
2	RASA[e]	x	x
2	ACS - Language Understanding		x
2	Amazon Lex[f]		x
2	Google Natural Language[g]		x
6	MARY[h]	x	
6	Amazon Polly[i]		x
6	Google Cloud Text-to-Speed[j]		x

[a]https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API
[b]https://github.com/mozilla/DeepSpeech
[c]https://aws.amazon.com/transcribe/
[d]https://cloud.google.com/speech-to-text
[e]https://rasa.com/
[f]https://docs.aws.amazon.com/lex/index.html
[g]https://cloud.google.com/natural-language/
[h]http://mary.dfki.de/
[i]https://aws.amazon.com/polly/
[j]https://cloud.google.com/text-to-speech

3.1. Selected Components

With our development team, we looked at the individual components, did isolated tests and based on those results judged their usability for building our VA. A list of these components and their associated steps is given in Table 1. We focused on open source tools and Azure components as both were easily available to us. Reading the online documentation and different blog posts, we believe that the other cloud services will act similarly, both with respect to effort for implementation and produced quality. Given the scope of our activity and limited budget, an in-depth comparison could not be done; next we summarize our findings from this phase.

For the speech recognition, the new Web Speech API was a positive surprise. Our phrases with everyday language were often successfully recognized and results matched with the commercial one. Both had problems when using specific terms from the industrial domain. Azure Cognitive Services has options to customize

Table 2: Excerpt from the database.

Index	Time	State	F. State	...
1	01.2019	Germany	BW	...
2	01.2019	Germany	NRW	...
3	01.2019	Switzerland	Aargau	...

Index	...	City	target	forecast	value
1	...	Aach	93	92.95	63.49
2	...	Aachen	82	73.09	45.19
3	...	Aarau	93	93.40	82.73

a language mode by (re-)training, which gives the possibility to improve recognition. For our initial test, we did not consider this option. The results with Mozilla DeepSpeech were mixed. The out-of-the-box model performed badly; after training with some texts from the domain, the recognition rate was acceptable and comparable to the other components. The training option is similar to the customization of the Cognitive Services. Also, on the speech synthesis part, the new Web Speech API performed very well. Here, some testers even ranked the audio better compared to the commercial one. The option to customize Azure Cognitive Services was not evaluated. The results from MARY felt short. However, due to the enormous number of parameters that can be tuned, this comparison is likely unfair. It requires more effort and knowledge of the solution compared to the other two.

For the intent recognition, both RASA and Azure Cognitive Services performed equally well and required the same training amount on our initial samples. In both cases, the training set required examples of requests with variations. We used Chatito[6] to specify the structure of phrases and their components; the tool generates the samples from that. RASA's base configuration only lacked in the recognition of parameters, which could be fixed with a proper configuration.

4. Research Virtual Assistant Demonstrator

For our demonstrator, we wanted to pick an example that feels realistic to an industrial setup, while an evaluation could be done without extensive prior knowledge or on-boarding. We selected a scenario based on control room operators that were inspired by the challenges described in [30]. An operator monitors the plant status and, if necessary, makes adjustments to the automation program in order to optimize production to return to steady states, and to avoid dangerous situations. Thus a VA enables him to extract status

[6]https://rodrigopivi.github.io/Chatito/

Table 3: Sample requests with their levels. Relevant parameters are marked in italic.

Level	Task
L0	What is the *forecast* for *Bad Dürrheim* in *August 2019*?
	What is the *target* for *Nordrhein-Westfalen*?
	What is the *value* for *Austria*?
L1	What is the aggregation of the *value* for *Baden* in *Aargau* in *Q3 2019*?
L2	Who are the *three locations* with the *best forecast* in *Austria* in *July 2019*?
L2	When was the *worst performance* of *Berlin* with respect to *target*?
L3	Compare *Heidelberg* and *Mannheim* with respect to their *value* in *2019*.
	For *Q4 2019* compare the *forecast* of *Wien* to its *value*.

information, aggregate and compare them would be a natural fit. Names and references to equipment would be encoded in technical terms that are not directly part of the natural / everyday language. Also, there is a typical hierarchy in the system that should ideally be reflected in the demonstrator.

4.1. General Setup

We reduced and abstracted this scenario of the control room operator: The technical terms will be represented by the name of cities in Austria, Germany, and Switzerland taken from lists in Wikipedia[7] and the hierarchical information of federal states and states. The status information will be given by three values, a raw value, a forecast prediction, and a target over a time period of one year on a monthly basis. A sample is shown in Table 2. While this may seem like an immense simplification, we will see in 4.2 that there are still many challenges. The associated tasks of information retrieval are grouped in by the four levels of task complexity introduced in the foundation section (L0 = reading out value, L1 = simple operations on values, L2 = identification of values by criteria and L3 = comparison of values and examples are given in Table 3.

On the first level L0, there is the retrieval of values, forecasts, or targets for a specific location and optionally, a given time. If no time is given, the latest entry should be used. If a location is only

[7]https://de.wikipedia.org/wiki/Liste_der_St%C3%A4dte_in_Deutschland, https://de.wikipedia.org/wiki/Liste_der_St%C3%A4dte_in_%C3%96sterreich, https://de.wikipedia.org/wiki/Liste_der_St%C3%A4dte_in_der_Schweiz

specified via federal state or state, then the sum of all its entries is considered. The next level L1 adds basic aggregation in time. For example, “in Q2 of 2019” or “from April to September”. If no specific time range is given, the last quarter is assumed. The level L2 asks for lists of the top k best or worst performers comparing the value to the target or forecast’s value. Like L0, a location or a time can be specified, with the same default behavior when omitted. On the final level L3, a comparison of different locations at a specific time or different time points at a specific location needs to be done. These levels approximately correspond to the number of conceptual operations a user would need to perform if only the raw table is available. L0 is access and optionally summation, L1 requires the filtering and summation of entries, L2 needs calculating intermediate values, sorting, and slicing and the last level L3 combines operations of L1 and L2. While a more sophisticated metric for a cognitive load such as [12] potentially allows for deeper insights, our approximated categorization, with its intuitive notion is sufficient for our evaluation. Overall, the tasks (and their complexity) easily correspond to activities in industrial applications. Especially to KPI operations or work with process variables in different plant segments, relevant for production in chemical processes. The tasks on the first two levels represent generic requests to obtain data and perform simple aggregation. Most industrial applications have such tasks in general; the properties such as names, locations, and time frames will always be domain-specific and thus differ naturally. Our data set has over 2000 unique locations and even duplicated names for cities, comparable to data set of small to mid-sized industrial setups. The more complicated tasks (L2 and L3) do not carry over directly. In our case, we selected these because our testers have backgrounds in data science and analytics, and thus, these tasks are familiar to them. In general, similar tasks that require aggregation, ranking, sorting, slicing, and comparison can be found across industrial applications. Similarly, to the previous deviations in properties, here, the goal and order of such operations are always very domain-specific and thus differ. Still, the dynamic arrangement and the variations in parameters are comparable to our tasks on L2 and L3. Summarizing, we see in our constructed tasks a fair and unbiased representation of activities that arise in the context of industrial applications.

4.2. Building the Demonstrator

Next, we describe our experiences in combining the components of Section 3.1.

We started with RASA to identify intents and extract relevant entities. It requires a set of examples in order to build a model. We did not have a historic set of recorded actions. Thus we needed to generate a training set. As suggested in the documentation of RASA, we used Chatito (`https://rodrigopivi.github.io/Chatito/`) to create a sample file. Chatito uses a DSL to specify the structure of phrases and the list of available entities. A challenge appeared in the identification of locations. For the training, the defined entities for city, federal state, and state in the same way as the database columns. We expected that RASA is able also to identify the type of the location. However, the skew distribution of names, i.e. 3 states, 49 federal states, and over 2400 cities, leads to the bias in the training set that RASA will most likely see cities as standalone locations. Similarly, for phrases like “city in …”, a federal state is much more likely than a state. We decided to keep that limitation and make a proper identification in the service level. Furthermore, we configured RASA to flexibly extract entities and not rely on fixed lists. Thus, typos in names are not corrected.

This trade-off in RASA that types of locations may be inconsistent and that inputs differ from the technical source labels leads to creating an auxiliary module in the service layer. This module normalizes given locations and time frames. In our case, the underlying SQL database did not support full-text search. We implemented a periodical crawl of the entries, built a local data set, and used an established library for fuzzy searches. Our backend to map intents to services is written in NodeJS. Therefore we selected the library Fuse.js[8]. As some of the city names have uncommon forms, e.g. “Singen (Hohentwiel)”, “Falkenstein/Harz” or “Eisleben, Lutherstadt”, we added a heuristic to obtain a normalized form that humans use. Overall, this strategy should cover most issues with misspellings or deviations from technical labels in the database. Unfortunately, this increase in convenience also increases the level of ambiguity. The normalization raises the number of truly duplicated names from the original 13 to 30. The total number rises even to over 200 if we consider close matches, i.e. names with a search score of less than 0.1. Most of these classes, similar to the duplicates, have few members and are expected. Examples are “Feldkirch” and “Feldkirchen in Kärnten”, “Weingarten” and “Leingarten”, and “Bendorf”,“Erbendorf” and “Dübendorf”. To our surprise, there are also classes with more than 10 entries. The maximum class has 93 members. It turns out these cases are names that are sub-strings of other cities, e.g. “Burg”, “Lingen” or “Stein”.

[8]`https://fusejs.io/`

The necessity and complexity of this auxiliary module is clearly due to our specific data set. One can expect that technical names in an industrial setup are more homogeneous. While this is true for the underlying specification, one can also expect that users tend to use related abbreviations or made-up alternatives and overall will make mistakes. A direct comparison is only possible on specific data and thus is skipped. Based on the testers' feedback, it feels realistic with respect to industrial setups in the process industries.

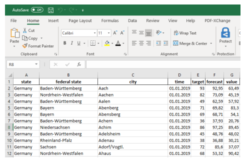
5. Evaluation

We describe our setup and our evaluation, first. Followed by observations and lessons learned from the agile development of the prototype and the evaluation. The reported research roughly follows a design science approach [13]. To explore the adoption topic of VAs in the industry (Motive), we have decided to investigate this topic from the user acceptance perspective based on prototype implementations (Objective). VA prototype design has been studied and resulted in the development of an industry-oriented VA (Design and Development). In the evaluation, we used a laboratory setup to explore user acceptance of the developed VA prototype in a comparative setup with other tools. With a focus on qualitative data collection from participant observation, semi-structured interviews with guiding questions, the approach applies ideas from grounded theory [11].

5.1. Setup

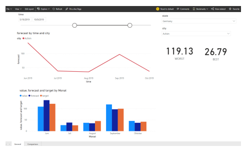
The evaluation setup was as follows. Using snowball sampling we identified 6 professionals who work in industrial automation. All professionals were male and have hands-on experience in process automation, having positions in R&D and product development. The goal was to capture process industry's mindset, even though the demonstrator presents generalized tasks. We scheduled a meeting with every participant and gave a short introduction to three systems, shown in Figure 2. The first application is Excel, with the raw data used for the demonstrator (Figure 2a). The user was allowed to add filters, change the sorting, create pivot tables and other means of auxiliary help. The second one is a business intelligence implementation of the analytics questions using Power BI (Figure 2b). Users were allowed to freely interact with the UI, selecting different inputs or time frames, but did not get administrative permissions to change the general view. The third application is our assistant demonstrator, accessible via a web site either on a desktop PC (Figure 2c) or a mobile phone (Figure 2d).

The participants spent ten minutes with each tool.

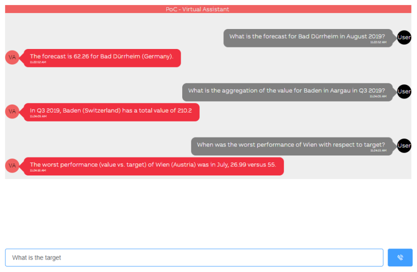


	A	B	C	D	E	F	G
1	state	federal state	city	time	target	forecast	value
2	Germany	Baden-Württemberg	Aach	01.01.2019	93	92,95	63,49
3	Germany	Nordrhein-Westfalen	Aachen	01.01.2019	82	73,09	45,19
4	Germany	Baden-Württemberg	Aalen	01.01.2019	49	62,59	57,92
5	Germany	Bayern	Abenberg	01.01.2019	71	69,82	83,3
6	Germany	Bayern	Abensberg	01.01.2019	69	68,71	54,1
7	Germany	Baden-Württemberg	Achern	01.01.2019	36	37,93	20,76
8	Germany	Niedersachsen	Achim	01.01.2019	86	97,25	89,45
9	Germany	Baden-Württemberg	Adelsheim	01.01.2019	45	48,76	48,02
10	Germany	Rheinland-Pfalz	Adenau	01.01.2019	38	36,88	30,21
11	Germany	Sachsen	Adorf/Vogtl.	01.01.2019	72	85,6	37,07
12	Germany	Nordrhein-Westfalen	Ahaus	01.01.2019	68	53,32	90,42

(a) Excel



(b) Power BI



User: What is the forecast for Bad Dürrheim in August 2019? 11:22:52 AM

VA: The forecast is 62.26 for Bad Dürrheim (Germany). 11:22:52 AM

User: What is the aggregation of the value for Baden in Aargau in Q3 2019? 11:24:05 AM

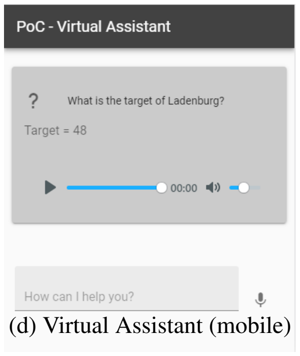
VA: In Q3 2019, Baden (Switzerland) has a total value of 210.2 11:24:05 AM

User: When was the worst performance of Wien with respect to target? 11:24:15 AM

VA: The worst performance (value vs. target) of Wien (Austria) was in July, 26.99 versus 55. 11:24:16 AM

What is the target

(c) Virtual Assistant (desktop)



? What is the target of Ladenburg?

Target = 48

00:00

How can I help you?

(d) Virtual Assistant (mobile)

Figure 2: Screenshots of the different tools used in the evaluation

The order of tool usage was random. The participants had a sheet with tasks to be conducted with each tool. They were asked to work on those tasks, but also to explore the tool in general. During this time, we observed them and took notes. After this, we interviewed the participants and obtained a summary of their experience in short statements. We asked them to consider the difference between the systems in their statements. Based on the field notes, we asked additional questions, if we felt important situations were not mentioned, or if we felt that there was a mismatch between the response and our observation. Like this, we strived for an alignment between field notes and interviews. We iteratively generated clusters from the data. The final clusters were organized regarding technology acceptance and usability to address our research question regarding potential reasons, why VAs are not used in industrial setups. Here, we organized using Nielsen's five qualities of usability [24] and perceived ease-of use and perceived usefulness as two (connected) key factors of the technology acceptance framework [9]. Only memorability was not considered, as the setup did not provide any insights into this.

5.2. Analysis

We report our findings, using usability qualities, perceived ease-of-use and perceived usefulness.

Learnability (Ease-of-use) The decision for speech commands was the biggest challenge for all participants. Even in our small and relatively homogeneous group of testers, we saw a wide range of different formulations. Frequently users mentioned that they were uncertain regarding the commands the assistant was capable of. Some testers used code-like patterns, like "*sum, value, Ladenburg, October, November, December*", while others went for short descriptions using complete sentences with polite forms, like "*Hi there. Please, fetch the values for Ladenburg. You should use the time frame from October to December.*" Once a list of example phrases was provided, the variety decreased.

Interaction with Excel and Power BI did not require further explanation, but was immediately possible. This might be due to a bias in our tester group because Excel and analytic tools, in general, belong to their utility belt.

Errors (Ease-of-use) The overall number of errors with Excel and Power BI was smaller than with the virtual assistant, due to the complexity of learning the voice commands. Only few times the voice recognition was not able to do the text-to-speech correctly. One participant started to argue with the assistant "*I told you to show me the sum. Listen to me.*", which generated new voice recognition errors. This humanization of technology did not happen with Excel and Power BI. An interesting observation was the boundary matching. Two participants started to create new questions. "*Let us see whether you can answer ...*" In a play-like process they created new questions, slowly increasing question complexity, until the system generated errors.

Satisfaction (Ease-of-use) The assistant was the only tool that aroused curiosity. Observations like testing the system boundaries or modifying questions, just to get the response showed joy of use. Also for some basic tasks, the immediate response was commented by support, like "*That was quick.*" or "*Thank you.*" It is unclear, whether curiosity and joy of success stay intact for longer usage periods. The testers considered the VA as modern, but some also stated that the topic is hype. Excel and Power BI were considered a commodity.

Efficiency (Usefulness) The immediate response in natural language from the VA was considered beneficial in general. The feedback was very mixed concerning the length of the spoken answer. Some preferred to have just the answer, while others liked better complete sentences with repetition of the question. The overall tendency was at least neutral and mostly positive. All testers had extensively worked with Excel and analytic tools similar to Power BI. Thus it is no surprise that these were considered as a true baseline. Most participants stated that they would use a VA for low to medium complexity tasks, if available. The ability to work seamlessly in parallel with the VA on the mobile phone and the task sheet was noted positively. The often referenced cognitive overhead [25] of switching between applications or the requirement to translate expressions and intents was noted by one participant: "*This captures data from different sources, right?*". The participants noted the relevance of hands-free interaction and the options to seamlessly embed interaction with other tools. One considered that "In the future, this will be so good...", indicating that such systems are still considered to be in early stage.

While all testers expected just the result, for example, the number, the list of cities, or dates, they also wanted to have a way validate the interpretation of the input and the result easily. Again, complete sentences with the repetition of the entities were perceived as complicated.

Task Success (Usefulness)

Tasks regarding basic complexity questions were solved very quickly with all tools. For medium and high complexity tasks on the assistant, there was confusion regarding the correct questions. Nevertheless, once the question was identified, people were surprised by an "*answer exactly fitting to the question.*" One stated that the assistant is "*simpler, when you know it*". From the direct observations, how our testers interacted with the VA, it became quickly apparent that an assistant can provide immediate access to the required information in general and that our limited prototype already serves this purpose by addressing all complexity levels.

Summarizing, our evaluation indicates that our VA is a capable tool to provide direct access to the required information. The four complexity levels can be addressed, although complex question requires improvement – primarily due to verbose text answers on level 3 and 4. We expect high user acceptance, if a scenario is covered with clear question and answer schemes, requiring good understanding of the usage scenarios, close interaction monitoring and quick improvements based on user feedback.

5.3. Lessons Learned During and After the Implementation

The whole prototype was created by a small team of five developers. They worked on it as a side project over roughly four months. Early on, we adopted an agile manner and used mocks and placeholders during the development. We want to highlight a few items that took us a little bit by surprise as they did not appear during our scouting and research phase. Firstly, proper architecture and integration scenarios. We used a micro-service architecture due to the number of different components and their individual scope. While the general structure was very clear, trying to create a stand-alone and an integrated VA on the same basis did not work for us. Early on, we dropped the way for an integrated version completely. Secondly, internal communication paths, interruptions, and canceling are challenging as different components provide different results that are merged in the UI. Adopting finite state machines (FSM) helped us to keep the overview. Thirdly, the handling of non-success states and errors is similar to the previous item, hard, and the corresponding paths should be ideally never taken. We also used an FSM-inspired approach and tried to cover the different scenarios with tests. Even in our small prototype, we noticed that a significant amount of time is invested here. Finally, building a good language model. Having early and continuous access to a realistic tester group is very valuable and lead to steady improvements. Looking at it from a scalability and extensibility point of view, we expect that even a small number of supported tasks can lead to a large number of phrases, and that is not always possible to match a phrase to a single task uniquely.

6. Conclusion

VA or VUI systems become an increasingly usual application type. Nevertheless, the systems are not yet common in non-consumer facing domains. We investigated this further with respect to the ability of a state of the art system to address reading out information typically required in information work and the acceptance of the realization of these tasks by VAs. For this purpose, we focused on tasks with different complexity and conducted a user study, comparing the acceptance of the VA compared to Excel and Power BI.

The available technologies cover the core necessities for the development of virtual assistant applications: speech recognition and synthesis, natural language understanding, and intent definition with service integration. The convenience for our industrial setup was low, and we faced limitations early on. Customization of speech recognition is necessary and requires substantial effort. Furthermore, the entity extraction from natural language in domains with many variants and synonyms is difficult and will, in many cases, require dedicated post-processing. Another limitation of the considered domain is that most existing components are cloud services. The development of on-premise applications is possible, but the component assortment is smaller and requires more configuration effort.

The user study has shown that it is possible to develop a virtual assistant for a given domain, covering all 4 complexity levels with manageable effort. Nevertheless, providing a first virtual assistant is just a first step in an incremental development process. Further design support for virtual assistants, like it is available for a graphical user interface, would be beneficial. Beyond this, we have identified a number of lessons learned, which are likely to support user experience and general user acceptance of a virtual assistant.

The study showed that design of commands and language seems to be a major obstacle for user acceptance. Users felt most familiar with simple requests, but once they had more complex goals it was challenging to generate the request and have a response of the right length and focus. Future research should investigate language design for VAs further, especially for tasks of higher complexity. Possible directions are languages for formulating complex queries, as we realized that people were quickly adjusting their language to a shortened command language. Another research direction could be the dynamic generation of focused responses to queries. The goal should be immediate and focused answers by a VA. This could lower cognitive burden and position in contrast to todays' trend with BI tools to share complex charts or tables which aggregate large amounts of information. Next to language, we see a challenge in the close investigation of the information workers' workflows to align a solution to daily work. The work presented in this paper applies a system-comparative approach to investigate VA acceptance. Future investigations should also investigate VAs separately, to have an unbiased user feedback regarding acceptance.

"Hello world" assistants are easy to implement. Addressing high complexity tasks and addressing a complex domain without the user losing orientation is a bigger challenge. Despite the mentioned limitations, the study participants enjoyed the simplified access to information the system offered, and were looking forward to future iterations the system. Therefore, we see clear potential in these systems and the need for further research.

References

[1] Amazon: Build Skills with the Alexa Skills Kit (2019)

[2] Ambite, J.L., Chaudhri, V.K., Fikes, R., Jenldns, J., Mishra, S., Muslea, M., Uribe, T., Yang, G.: Design and implementation of the CALO query manager. Proceedings of the National Conference on Artificial Intelligence **2**, 1751–1758 (2006)

[3] Amershi, S., Weld, D., Vorvoreanu, M., Fourney, A., Nushi, B., Collisson, P., Suh, J., Iqbal, S., Bennett, P.N., Inkpen, K., Teevan, J., Kikin-Gil, R., Horvitz, E.: Guidelines for human-AI interaction. In: Conference on Human Factors in Computing Systems (2019)

[4] Baker, J.: Voice User Interfaces (VUI) — The Ultimate Designer's Guide (2018)

[5] Barker, P.: Wired for Speech: How Voice Activates the Human-Computer Relationship. The Electronic Library **24**(2) (mar 2006)

[6] Bentley, R., Hughes, J.A., Randall, D., Rodden, T., Sawyer, P., Shapiro, D., Sommerville, I.: Ethnographically-informed systems design for air traffic control. In: Proceedings of the Conference on Computer-Supported Cooperative Work. pp. 123–129. Publ by ACM, New York, New York, USA (1992)

[7] Berdasco, López, Diaz, Quesada, Guerrero: User Experience Comparison of Intelligent Personal Assistants: Alexa, Google Assistant, Siri and Cortana. Proceedings **31**(1), 51 (2019)

[8] Campagna, G., Ramesh, R., Xu, S., Fischer, M., Lam, M.S.: Almond: The architecture of an open, crowdsourced, privacy-preserving, programmable virtual assistant. 26th International World Wide Web Conference, WWW 2017 pp. 341–350 (2017)

[9] Davis, F.D., Bagozzi, R.P., Warshaw, P.R.: User Acceptance of Computer Technology: A Comparison of Two Theoretical Models. Management Science **35**(8), 982–1003 (aug 1989)

[10] Engelbart, D.: Augmenting human intellect: a conceptual framework, SRI Summary Report AFOSR-3223. Engelbart, D. **1962**, 1–65 (1962)

[11] Glaser, B.G., Strauss, A.L.: The Discovery of Grounded Theory. Routledge (jul 2017)

[12] Ham, D., Park, J., Jung, W.: Model-based identification and use of task complexity factors of human integrated systems. Reliability Engineering and System Safety **100**, 33–47 (2012)

[13] Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Quarterly: Management Information Systems **28**(1), 75–105 (2004)

[14] Johnston, M., Chen, J., Ehlen, P., Jung, H., Lieske, J., Reddy, A., Selfridge, E., Stoyanchev, S., Vasilieff, B., Wilpon, J.: MVA: The multimodal virtual assistant. In: SIGDIAL 2014 - 15th Annual Meeting of the Special Interest Group on Discourse and Dialogue, Proceedings of the Conference. pp. 257–259 (2014)

[15] Jokinen, K., McTear, M.: Spoken Dialogue Systems. Synthesis Lectures on Human Language Technologies **2**(1), 1–151 (jan 2009)

[16] Kopp, S., van Welbergen, H., Yaghoubzadeh, R., Buschmeier, H.: An architecture for fluid real-time conversational agents: integrating incremental output generation and input processing. Journal on Multimodal User Interfaces **8**(1), 97–108 (nov 2013)

[17] Kouroupetroglou, G., Spiliotopoulos, D.: Usability methodologies for real-life voice user interfaces. International Journal of Information Technology and Web Engineering **4**(4), 78–94 (2009)

[18] Liao, Q.V., Hussain, M.M.U., Chandar, P., Davis, M., Khazaen, Y., Crasso, M.P., Wang, D., Muller, M., Shami, N.S., Geyer, W.: All work and no play? Conversations with a question-and-answer chatbot in the wild. In: Conference on Human Factors in Computing Systems - Proceedings. vol. 2018-April (2018)

[19] Lopatovska, I., Rink, K., Knight, I., Raines, K., Cosenza, K., Williams, H., Sorsche, P., Hirsch, D., Li, Q., Martinez, A.: Talk to me: Exploring user interactions with the Amazon Alexa. Journal of Librarianship and Information Science **51**(4), 984–997 (2019)

[20] Luger, E., Sellen, A.: "Like Having a Really Bad PA": The gulf between user expectation and experience of conversational agents. In: Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems. pp. 5286–5297. ACM, New York, NY, USA (may 2016)

[21] Machiraju, S., Modi, R.: Developing Bots with Microsoft Bots Framework, vol. 0. Apress, Berkeley, CA (2018)

[22] Microsoft: Übersicht Cognitive Services (2017)

[23] Mortensen, D.: How to Design Voice User Interfaces (2019)

[24] Nielsen, J.: Usability 101. Nielson Norman Group (2012)

[25] Oliver, N., Smith, G., Thakkar, C., Surendran, A.: SWISH: semantic analysis of window titles and switching history. In: Proceedings of the 11th international conference on Intelligent user interfaces. pp. 201–209. ACM Press (2006)

[26] Owda, M., Bandar, Z., Crockett, K.: Conversation-Based Natural Language Interface to Relational Databases. 2007 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology - Workshops (March 2015), 363–367 (nov 2008)

[27] Petraityte, J.: Build an AI voice assistant with Rasa Open Source and Mozilla tools

[28] Pinker, S.: A Theory of Graph Comprehension. In: Artificial intelligence and the future of testing, pp. 73–126. Lawrence Erlbaum Associates (1990)

[29] Porcheron, M., Fischer, J.E., Reeves, S., Sharples, S.: Voice interfaces in everyday life. In: Conference on Human Factors in Computing Systems - Proceedings. CHI '18, vol. 2018-April. Association for Computing Machinery, New York, NY, USA (2018)

[30] Schmidt, B., Siddharthan, S., Borrison, R., Cohen, A., Dix, M., Gärtler, M., Hollender, M., Klöpper, B., Maczey, S., Siddharthan, S.: Industrial virtual assistants - Challenges and opportunities. In: Adjunct Proceedings of the UbiComp/ISWC 2018. pp. 794–801. ACM, New York (2018)

[31] Watkins, D.: Global Smart Speaker Vendor & OS Shipment and Installed Base Market. Tech. rep., Strategy Analytics (2020)

[32] Weizenbaum, J.: ELIZA—A Computer Program For the Study of Natural Language Communication Between Man And Machine. Communications of the ACM **26**(1), 23–28 (1983)