

The Role of Reuse in the Successive Implementation of Conversational Agents

Corinna Rutschi

Institute of Information Systems, University of Bern
corinna.rutschi@iwi.unibe.ch

Prof. Dr. Jens Dibbern

Institute of Information Systems, University of Bern
jens.dibbern@iwi.unibe.ch

Abstract

More and more companies rely on the implementation of conversational agents (CAs) to automate certain of their processes. While CAs' initial development often mirrors an innovative process, their successive implementation can be made more efficient by drawing on other CAs' prior developments. We rely on data from a case study where a series of chatbots (which represent one type of CAs) was implemented. Routine theoretical concepts help us better understand how CAs may dynamically evolve and how their implementation can be accelerated. We found that a) the reuse of emerging or intentionally constructed means may allow accelerating the implementation of multiple successive CAs, b) means can be reused to extend or transfer functionality (through mutation or inheritance); and c) changes in the conversational context determine if means can be reused directly or not (through reproduction or recreation).

1. Introduction

Companies are introducing conversational agents (CAs) – one type of which are chatbots – to automate conversational business processes [17, 30, 32]. Human users can interact with CAs in conversations on specific topics. Companies often start implementing CAs in a particular area or department, such as customer support [34]. Based on such initial experiences, companies may identify additional departments or contexts in which they could implement more CAs. However, if they start from the ground up for each new CA to be built, this results in high costs and time expenditure. Accordingly, companies may ask themselves how they could reduce the overall implementation time and costs when successively building and implementing multiple CAs.

Prior research has examined different types of CAs [24, 34] and shed light on the kinds of processes that such CAs can take over [17, 32]. In addition, it was explored how such CAs can be designed [28] or distinguished based on various anthropomorphic

features [12, 28]. This included investigating when it would be useful to design CAs more or less human-like [12, 29]. However, prior research on the implementation of CAs has mainly focused on their one-time implementation. This has led to initial insights into the steps to be taken and the underlying conditions for automating particular (business) processes [9, 25, 26].

Potential processes CAs can perform resemble the notion of organizational routines. Routines can be described as "executable capabilities for repeated performances" that were learned within a particular organizational context [7]. Routine theory illuminates how humans perform routines, which allows us to open the black box of how routines are composed for humans and how they must be recomposed if they are to be adopted by some technological artifact [9, 26]. Thus, routines can be enacted by humans, by technological artifacts (such as, for example, CAs), or by some combination of the two [9, 20]. In the latter case, technological artifacts can support humans to perform a particular routine better. But artifacts (in our case CAs) may also completely replace humans and take over certain routines entirely [9]. In this case, the CA becomes the center of the routine and thus the routine's materiality [9, 22]. Materiality is then related to a CA's material agency, which determines what a routine must look like for the CA to perform it [9, 11]. It has been shown that automating routines through CAs requires an understanding of the structure of both existing routines to be automated and the operating principle of a CA [25, 26]. Since humans and CAs execute routines differently, routines must be translated to some degree for CAs to perform them. Implementing CAs then essentially means translating routines for CAs [26].

Our data suggest that CAs are often implemented in succession, with multiple CAs being implemented successively rather than just one. In case companies implement each CA separately and from the ground up, this describes a highly time-consuming and potentially inefficient approach. To avoid such inefficiencies, a better understanding of how CAs can be built and implemented and how the implementation

of one CA can benefit the implementation of another is needed. To understand the extent to which companies can build and implement multiple CAs successively, we aim to answer the following research question: *How and under which conditions can the implementation of multiple CAs be accelerated based on previously implemented CAs?*

To answer our research question, we rely on data from a case study where a bank implemented a series of chatbots in their customer contact center. We show exemplarily how the bank implemented four successive chatbots (i.e., a German and a French FAQ bot, an e-banking bot, and a voice bot).

We contribute to routine theory and literature on CAs by examining how the implementation of CAs can be accelerated when building on practices of reuse. Specifically, we found that a) when implementing multiple CAs or extending the functions of existing CAs, one can build on what already exists (i.e., means); b) means can be reused for functionality extension or transfer (i.e., mutation vs. inheritance); and c) reuse depends on how the context of a conversation changes (i.e., type of conversational context change) which determines if means can be reused directly (i.e., reproduction) or through extension or adaptation (i.e., recreation).

The paper is organized as follows. First, we introduce literature on CAs. Then, we will introduce the basic concepts of routine theory, especially in relation to routine dynamics. Next, we will outline the methodological approach and the case we studied. The following is an extended analysis of our data. Finally, we will introduce our findings and draw a conclusion.

2. Literature Background

Conversational agents (CAs) – one type of which are chatbots – are designed to perform specific tasks by following specific behavior patterns or rules. Thereby, CAs are essentially computer programs that are capable of conducting conversations with human users in written or spoken form [24]. Companies are increasingly introducing CAs to automate conversational business processes [31] that were previously performed by humans [25, 26]. Suitable processes must be rule-based, uncomplicated, and carried out in high volumes [17, 34]. Often, companies can build CAs by dragging and dropping individual elements within a dedicated software environment. Conversations are then structured in decision trees which refer to the structure of a specific dialog [25, 26]. After releasing a CA, the human user can, for example, interact with the CA via a user interface (UI), such as a pop-up window integrated on a website. Thus, users, who may be employees or customers, can

engage in dialogs with CAs to obtain the information they would otherwise have asked customer service, for example [31]. Thereby, CAs are no longer just "passive tools waiting to be used" [3, 23]. They carry out processes more or less autonomously, exerting some degree of agency. Additionally, CAs seem to adopt some human-like capabilities [28-30], whereas they outperform humans in executing specific tasks and processes [17, 32]. Our paper further builds on previously established grounds on how CAs can be implemented [25, 26] while focusing on how such efforts can be reduced and the overall implementation of multiple CAs accelerated.

3. Routine Theoretical Concepts

If processes are automated by CAs, such processes must first be translated to some degree. Translation is necessary because the operating principles of CAs are different from those of humans. Translating processes then requires some understanding of how humans have performed processes before [9]. Processes that CAs can take over resemble the notion of organizational routines. As mentioned above, routines can be described as "executable capabilities for repeated performances" that were learned within a particular organizational context. A capability can be defined "as the capacity to generate action, to guide or direct an unfolding action sequence, that has been stored in some localized or distributed form". The context describes the setting within which a routine is performed and which counts as a "source of necessary inputs to actions". And the term 'learned' reflects that different enactments of the capability may be possible depending on who (e.g., a human or a CA) is performing a particular routine [7]. Considering a CA, a CA's capability would be to conduct a certain conversation. The context would describe what the conversation is about (e.g., a specific product) and who is involved (e.g., customers). And 'learned' would describe the extent to which a sequence of a conversation is implemented in the CA in the form of a decision tree.

Routine theory helps us to unlock the black box of how humans perform routines [20]. However, besides humans also artifacts can perform routines. Previous literature on routines has argued that technological artifacts that perform routines move from the periphery to the center of the routine. When such artifacts like CAs take over routines, they become the materiality of routines [9, 22]. Thereby, they can influence the routine [6]. If we accept the artifact as part of the routine and agree that it becomes the routine's material, which means that it has affordances and agency, then this allows us to open the black box

of how artifacts, such as CAs, can shape performances of routines [9, 11].

Routines are composed of ostensive and performative aspects. The ostensive aspect refers to formal rules and procedures that can be described as the 'guidelines' of performing the routine. With respect to CAs, such guidelines (i.e., the ostensive aspect) represent the dialog structures and rules responsible for variations in the flow of a dialog. For example, if a CA is built to answer questions about multiple products, it needs specific information from the user to provide appropriate answers. The performative aspect refers to the actual performance of the rules and procedures. Thus, based on if-then rules, and depending on the information provided by the user, the CA scans the underlying decision tree for suitable answers and responds to the user accordingly [26].

Unlike CAs, humans do not always perform routines in the same way [16] because they are creative and adaptive [13, 18]. When humans enact routines, both the performative and the ostensive aspects are different from each other and influence each other. For example, if an employee needs to complete a specific task, he or she can deviate from the prescribed steps (i.e., the ostensive aspect) and perform the task (i.e., the performative aspect) differently. Over time, this can change the routine of which the task is a part [9, 16].

This is different for CAs. CAs initially influence how the ostensive aspect of the routine needs to be formalized (i.e., the structure of a dialog and the rules by which it needs to be conducted). Thus, CAs specify how the ostensive aspect of a routine executed by a human must be translated in order for the CA to perform it. Later on, however, a CA cannot influence the routine itself anymore but executes it by strictly following given rules (i.e., the ostensive aspect). Consequently, no continuous reciprocity between the ostensive and the performative aspects can be observed, as is the case for humans. Rather the ostensive aspect corresponds with the performative aspect as long as the CA does not get further extended or adapted by the developer who acts as a mediator. Thus, once the ostensive aspect is initially defined, CAs are incapable of changing it again on their own (yet); human developers have to do this to change or extend a CA's capabilities [26, 27]. The more often the developer has to adapt or further extend the CA's capabilities, the more resources (i.e., time and money) need to be invested.

Rutsch and Dibbern [25, 26] introduce an iterative framework to explain to what extent an individual CA can be implemented. However, implementing a CA may not stop when it first goes live. Instead, CAs need to be continuously extended

and adapted according to new or additional requirements or changes in their context. For example, a CA could provide information about different products of an online store. If an additional product is offered that is not yet known to the CA, the CA cannot provide information about it. For it to be able to do so, dialog structures must be extended accordingly. Let us assume that a similar CA should be implemented in a different context. Then it is no longer sufficient to extend the dialog structures, but they must also be adapted according to the new context [11]. We could argue that each CA takes over one routine in a particular area, which in turn can represent a composition of different subroutines. In other terms, each CA engages in a broader conversation consisting of multiple speech acts. These speech acts would then describe individual streams of dialog per broader conversation topic [2]. The CA from the example mentioned above would then conduct conversations to provide specific product information. This routine could be divided into two speech acts or subroutines. One subroutine could be about providing information about the features of each product. The other subroutine could be about where to buy the product. Thus, one CA could, over time, take on several subroutines that are part of a broader routine. Through this, the CA can be continuously expanded [15].

Recent advances in routine dynamics shed light on the dynamic evolution of routines through the performance of preceding routines. As mentioned before, humans do not always perform routines in the same way [18, 23]. Ditrach and Seidl [13] argue that the performance of routines can lead to the emergence of means. Such emerging means can be reused to define and achieve current and new ends. Reusing means in any subsequent performance of a routine implies that over time a routine converts into a different form or new routines materialize [16]. For example, an employee might need to create slides. He or she could create both the content and the design of the slides from the ground up. However, the employee could also create a master slide set in which he or she defines the slide design. This master slide set could then be reused for other presentations and thus be considered an example of a mean.

The notion of means may also be helpful to understand how to accelerate the successive implementation of multiple CAs. However, means play a different role when humans perform routines [13] than when CAs do so. CAs cannot autonomously make use of emerging means, but the human developer that builds the CAs can do so [26, 27]. Based on our data, companies often seem to aim to implement several CAs successively. Therefore, they could consider two approaches. Either they could build or

extend each CA from the ground up or build on previous implementations. Such previous implementations may have resulted in certain building blocks or components that can be described as means and whose reuse may accelerate the overall implementation process [13]. Thus, following the second approach, the overall effort of implementing multiple CAs may be reduced. In the following, we elaborate on this using our data.

4. Methodological Approach

To answer our research question, we have chosen a case study research method [14]. Based on theoretical sampling, we identified and studied a series of chatbots (as one type of CAs) to gain an understanding of how multiple CAs evolve and what role contextual conditions play in the reuse of means. For this, we identified a single case in the context of which multiple chatbot instances were implemented. Given that our key objective was to build theory, the research thrust was exploratory [5]. We conducted ten semi-structured interviews between October and November 2017, in a second round in September 2018 and in a third round in March and April 2020. Conducting the interviews helped us to obtain a holistic picture of the case [35]. The interviews were conducted with the initial project manager, the product owner, the scrum master, the external partner, two application managers, and the content manager. Thereby, we interviewed the product owner, the scrum master, and the external partner twice each. The interviews lasted between 17 and 90 minutes, with an average of 51 minutes. The reason for the limited number of interviews is that the project team was relatively small. We analyzed other data, such as software suite manuals and project documentation, to make up for this. After transcribing the interviews, the first author applied Corbin and Strauss' [8] coding approaches to examine the data and identify patterns. Based on this, we derived how means can be reused and distinguished among two types of contextual changes. In the following, the case is presented in detail.

5. Case Narrative

The project we examined was carried out in a European bank that operated regionally. The bank had around 3700 employees and served almost 3 million private and business customers. As part of the bank's digital transformation strategy, chatbots were to be implemented. In line with this, the bank's objective was to transfer processes previously performed by call

agents in the customer contact center to chatbots. This should allow costs to be saved through efficiency gains. The project was initiated in October 2016. The implementation of the chatbots was approached in a somewhat exploratory manner, and a small project team was assembled for this purpose. The team consisted of an initial project manager (a product owner later replaced her), a scrum master, an external partner, an application manager (another application manager later replaced him), and the content team. The software solution the bank chose to build and implement the chatbots was designed so that no programming skills were required, but chatbots could be built on a graphical user interface. This allowed decision trees for dialogs to be modeled graphically and variations and synonyms to be easily added. During the project, four chatbot instances were built and implemented. We describe these four instances as four phases the project went through. The four phases include the building of the German FAQ bot (Phase 1), the French FAQ bot (Phase 2), the e-banking bot (Phase 3), and the voice bot (Phase 4). Below, we will discuss each of the four project phases.

5.1. Phase 1 – German Bot

The project was launched by building a German FAQ bot. This chatbot should be able to answer general frequently asked questions of the bank's customers. The idea was that customers would not always have to call the customer contact center in the future but that the chatbot could answer certain of their questions 24/7. Those questions should be simple and related to general information (e.g., information on the different types of accounts offered by the bank and how to apply for them); occur in high volumes; contain self-service components (e.g., how to change the correspondence address) or aspects that the customers could handle themselves; and refer to non-value-adding processes. Initially, such a chatbot should only be developed for German-speaking customers. Since this was the bank's first chatbot project, the project team had to build the German bot entirely from the ground up. Decision trees were modeled to structure dialogs around main questions, which constituted the root, while possible direct answers and follow-up questions formed the branches. One main question then required about 100 variations so that the chatbot was able to answer accurately. "Still, if there is a 101st question and the syntax is wrong, we are pretty sure the chatbot is going to map the question to the right main question." (External Partner). Decision trees refer to a dialog's structure, while the main questions refer to dialog topics. The conversations the chatbot could conduct did not require any system integration.

Customers could access the chatbot through the bank's website without being logged in.

During this first phase of the project, the German bot was continuously built and tested. Through testing, it was possible to determine whether any dialogs needed to be extended or adapted or whether any topics had not yet been implemented but on which customers could ask the chatbot questions. Testing, then, allowed the project team to ensure throughout the project (including for the later chatbots) that negative trends of fit, in terms of what the chatbots could answer versus what customers would like to ask, could be prevented. The German FAQ bot was officially released in November 2017. However, this did not mean that the development ended, but the German bot was continuously adapted and further developed after it went live. For further developments, it was possible in part to build on existing dialog structures. Additional dialog (sub-)topics could be identified through the interaction of the German bot with customers.

5.2. Phase 2 – French Bot

After the German bot was launched, the bank did not want to deprive its French-speaking customers of such a service. In the second phase of the project, a French bot was built that, like the German bot, would answer customers' frequently asked questions. Having already built a similar bot, the project team assumed they would not have to start from the ground up for the French bot. The German dialogs represented in decision trees already contained a considerable number of questions and corresponding answers. The project team assumed that the dialogs previously set up in German could simply be translated into French. Thus, the assumption was that everything once built could be reused. However, this turned out to be more difficult. Existing German dialog structures did not fit the new French-language context. The project team did not realize upfront that language was linked to cultural aspects, and thus conversations were structured and conceived differently in German than in French. Notably, the French-speaking customers expressed themselves and structured dialogs differently than the German-speaking customers did. To overcome this misfit of the German dialog structures in the French language context, the project team had to understand better how their French-speaking customers structured dialogs. Based on this, they could build dialogs accordingly for the French bot. Therefore, the content team got expanded to include a native French speaker. In the following, the content team had to translate and restructure the German dialogs to fit the new language context.

Thereby, the project team could build on knowledge gained when implementing the German bot. The French chatbot was released in October 2018. Again, however, this did not mean that its development ended there; the French bot was continuously adapted and further developed after going live. Additional topics and corresponding dialog structures that were built for the French bot could be used backward for the further development of the German bot after being translated and restructured accordingly.

5.3. Phase 3 – E-Banking Bot

The development of the e-banking bot was initiated in the summer of 2018. The e-banking bot should allow the bank's customers to ask customer-specific questions while being logged into the bank's e-banking system. Until then, the German and French bots could not provide any customer-specific information. With the integration into the e-banking system, the e-banking bot could retrieve such information from the system. This meant that dialogs needed to be customer-specifically tailored.

Throughout the development of the e-banking bot, the project team was once again able to draw on the knowledge it had already gained during the developments of the German and French bots. Dialog topics that had previously been modeled in German and French could be reused. However, since these had previously been modeled in a general and non-customer-specific way, they had to be extended so that the e-banking bot could conduct customer-specific dialogs. In addition to the already existing dialog topics, new topics had to be covered by new dialogs. The reuse of existing dialog topics and structures enabled the project team to develop the e-banking bot more efficiently. A first version of the e-banking bot was released in early 2019. As with the German and French bots, the e-banking bot was continuously adapted and further developed after going live. Additional topics and corresponding dialog structures built for the e-banking bot could be used backward to further develop the German and the French bots after being restructured accordingly.

5.4. Phase 4 – Voice Bot

The development of the voice bot was initiated shortly after starting the development of the e-banking bot. The voice bot should allow customers to interact not merely by text but also by voice input. Customers could reach the voice bot by phone, just as they had reached contact center employees before.

At first, the project team again assumed that a significant part of the existing dialog topics and

structures of the German, French, and e-banking bots could be reused to develop the voice bot. This was, however, not as easy as one had hoped. The project team realized that not only the German and French-speaking customers expressed themselves differently, but all customers spoke differently from what they wrote. "For example, the syntax is completely different when the customer asks, 'Can I check my account balance, please?'. Then he writes on the text channel: 'Account balance please.' Maybe two words. [...] However, when he enters it in the voice channel, it is more of a dialog, and he says, 'Yes, I think I got my paycheck yesterday, and I need to know what my balance is and check if I can pay my bills.' [...] And you just cannot compare how the customers write and how they talk to the assistant [voice bot]." (Product Owner). Thus, it was challenging to reuse existing dialog structures for the voice bot. Nevertheless, the project team translated some of the text dialogs into voice dialogs. "We will not be able to make 100% of the content we have modeled suitable for voice. That would lead to too much effort at the moment." (Product Owner). Some of the text dialogs hardly seemed suitable for the voice channel. Thus, the project team had to rethink its approach. They did this by adopting a voice-first approach and by training all team members accordingly. Voice-first meant that the project team would generate new dialogs for the voice context first, in a form that could potentially be reused for the text context later on. Reuse was reversed herein that the voice bot's dialog structures were to be reused retroactively for the text-based bots. The development of the voice bot was still underway when the data collection was completed. A first version of the voice bot was released in June 2020.

7. Analysis and Findings

In the following, we will shed light on how and under which conditions the successive implementation of multiple CAs can be accelerated based on previously implemented CAs. In the case presented, four successive CAs were built and implemented. Each of them operated in a different context within which each CA could perform specific conversational processes. We could argue that each of the four CAs was built to take over one broader routine that consisted of multiple subroutines. For the German bot the broader routine would then be conducting general FAQ conversations on multiple topics in German. Associated subroutines then describe the individual conversation streams per topic that the German bot could have a conversation about. For example, one subroutine could be related to the different account types offered by the bank, while another subroutine

could be about how a customer could open an account. The French bot is analogous to the German bot, simply in a different language context (i.e., French). For the e-banking bot, the broader routine is conducting customer-specific conversations in German and French related to e-banking. An example of a subroutine here would be displaying the current account balance to a customer. For the voice bot, the broader routine is to conduct customer-specific voice-based conversations in German.

During the successive implementation of each CA the project team could build on what was built before. Routine theory sheds light on how routines can dynamically evolve and change over time [13]. Thereby, routines consist of an ostensive and a performative aspect [26]. Depending on who is performing the routine (humans or technological artifacts such as CAs), the ostensive and performative aspects may diverge. This is at least the case where humans perform routines [9, 16]. When CAs perform routines, they become the materiality of the routine and determine how the ostensive aspect must be in order for them to perform it. Once the developer has defined the ostensive aspect, the CA strictly performs it. Thus, the ostensive and performative aspects become equivalent. This does not mean that the routine cannot change, but it does mean that the CA cannot change it autonomously (yet). However, the developer who creates the CA can do so. Changing the routine then essentially means that the developer changes specific subroutines or adds additional subroutines so that the CA can handle even more topics that belong in the broader routine context [26, 27].

Such changes can be addressed separately or, in the case that several CAs are implemented successively, the developer may build on the development efforts that have already been made. Dittrich and Seidl [13] introduce the concept of means, which emerge from the performance of routines and which, if built upon, can change routines over time. We observed similar dynamics in the case analyzed. During the implementation of the four successive CAs, components were generated, and knowledge was acquired that facilitated any subsequent implementation efforts. We argue that these generated components and the acquired knowledge can be compared to the idea of means Dittrich and Seidl [27] introduce. Here we describe the generated components as explicit means, while we describe the acquired knowledge as cognitive means.

When analyzing the implementation of the four CAs, we were able to identify several explicit means. The implementation of the German and French bots resulted in dialog structures and topics in German,

respectively French for various general FAQ conversations. The implementation of the e-banking bot resulted in dialog structures and related topics around customer-specific e-banking conversations. The implementation of the voice bot resulted in dialog structures and related topics around customer-specific voice-based banking conversations. During the implementation of the four CAs the project team acquired knowledge that allowed them to become better at knowing how to build and implement CAs. We define this knowledge as the cognitive means.

The overall project pace could be accelerated once the project team could reuse explicit and cognitive means. Here, we distinguish the area of reuse (i.e., in the same or a different context through mutation or inheritance) and the type of reuse (i.e., through reproduction or recreation). In the following, we will first discuss the area within which means could be reused. In the project, means were partially reused in the same context (i.e., for the same CA through mutation) and partially in a different context (i.e., for a different CA through inheritance). Both the cognitive and explicit means that resulted from the implementation efforts of all CAs were reused in times in the same context and in other times in different contexts. Reusing means allowed the project team to extend functionality (i.e., conversational subroutines) of a current CA (see quadrant A of Figure 1) or to transfer functionality to a different CA in a different context (see quadrants B and C of Figure 1).

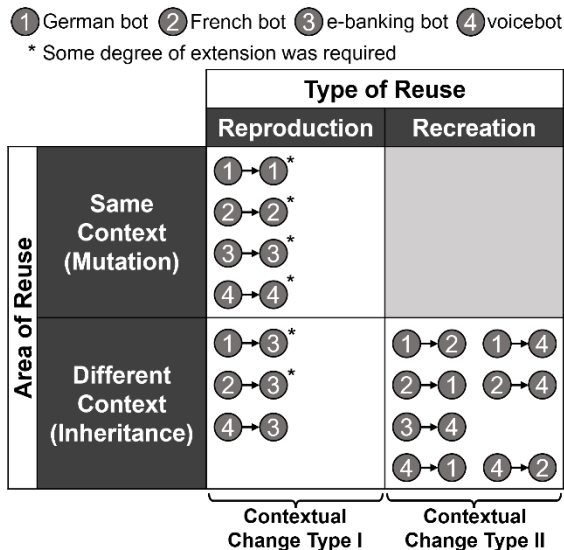


Figure 1. Means reuse in the case

Reusing means to extend the functionality of a current CA can be referred to as mutation [33, 36]. For the German bot, explicit means (i.e., German dialog structures) could be reused to build additional dialog

structures faster for other German topics. The same was true for the French bot and its French dialog structures. For the e-banking bot, customer-specific dialog structures in German and French could be reused to add even more dialogs around more topics quicker. Furthermore, customer-specific voice-based structures were reused to add additional functionality to the voice bot.

In the project, means were reused in the same context to add functionality and in different contexts to transfer functionality. Reusing means to transfer functionality to new contexts can be referred to as inheritance [33, 36]. In the bank's case, we found that specific means could be directly reused in a different context. Other means needed to be extended or adapted so that they could be reused in a different context. Where means could be directly reused in the same and a different context, we can also say that means could be reproduced (see quadrants A and B of Figure 1). Thus, reproduction indicates that means can be directly reused regardless of potential contextual changes [10, 13, 16]. It may also be the case that means can be reused directly but must be extended to some degree. We observed both direct reuse of means and direct reuse where specific extensions were needed. Explicit means from the German and French bots (i.e., general FAQ dialog structures in German and French) could be directly reused for the e-banking bot. However, those dialog structures needed to be extended from non-customer-specific to customer-specific. Explicit means from the voice bot could be directly reused for the e-banking bot with no need for an extension.

Where means could not be directly reused in a different context (see quadrant C of Figure 1), we can say that means needed to be recreated in order to be reused. Recreation indicates that means must be adapted depending on contextual conditions to be reused [10, 13, 16]. In the chatbot case, explicit means from the German bot (i.e., dialog structures and topics in German) could be reused by recreating them for the French bot. This required translating German dialog topics and restructuring German dialog structures. Similarly, explicit means from the French bot could be reused by recreation for the German bot. Explicit means (i.e., general FAQ dialog topics and structures in German and French) from both the German and the French bots could be reused for the voice bot by recreating them from text to voice-based and from non-customer-specific to customer-specific. Explicit means (i.e., customer-specific dialog topics and structures) from the e-banking bot could be reused by recreating them from customer-specific to non-customer-specific for the German and French bots. Additionally, e-banking dialog structures could be

reused for the voice bot by recreating them from text-based to voice-based. Finally, explicit means (i.e., customer-specific voice-based dialog topics and structures) from the voice bot could be reused by recreating them from customer-specific to non-customer-specific for the German and French bots.

It is essential to understand the contextual conditions that prevent direct reuse and how to modify means so that they can be reused (see Adler et. al [1]). We distinguish among two types of contextual changes. Change Type I refers to different languages (German vs. French) or forms (writing vs. speaking) in which a conversation can be conducted. Change Type II refers to different topics (e.g., general FAQ vs. customer-specific e-banking topics) that a conversation can be conducted about. Altogether, in the case studied, we observed two instances of Change Type I: the change from the German to the French language context and the change from text to voice-based context. Thereby, we observed that explicit means could be reused through recreation. In addition, the project team built voice-first dialog structures for the voice bot that could be directly reused for the e-banking bot through reproduction. Thus, we may assume that if means are intentionally constructed, they may be reproduced even in case of a contextual change of type I. Besides the two instances of Change Type I, we observed one instance of Change Type II: the change from non-customer-specific to customer-specific banking dialogs. In the case of Change type II, we observed that explicit means could be reused through reproduction with the need for an extension.

In summary, whenever the way a conversation is conducted in (Change Type I) changes, it seems that explicit means may be reused through recreation, or intentionally constructed means (as was the case in project phase 4) may be reused through reproduction (in some cases). Whenever the broader topic of a conversation changes (Change Type II), explicit means may be reused through reproduction with a potential need for some degree of extension.

7. Discussion

How and under which conditions can the implementation of multiple CAs be accelerated based on previously implemented CAs? Companies implement CAs to automate specific conversational processes that their human employees would otherwise perform. Thereby, they aim to reduce the workload for their human employees and, associated with this, reduce costs [17, 32]. Those goals can only be achieved if the effort required to automate processes through CAs' implementation is inferior to

the effort human employees make when performing the processes in a non-automated way. However, based on our analysis, companies often seem to invest a lot of time and money to build and implement such CAs. Implementing a CA implies continuously extending its capabilities and adjusting the CA to potentially changing environmental conditions [26]. Adapting or further extending CAs can be done more efficiently if the necessary resources (i.e., time and money) are kept as low as possible. Thereby, approaches of means reuse can be considered.

When humans perform routines means may emerge that can subsequently be reused so that humans can perform a different routine or a current routine differently [16]. In the context of CAs, reusing such means can be considered a mechanism that accelerates how one or multiple successive CAs can be built and implemented [4]. Thereby, means may be reused in the same (i.e., mutation) or a new context (i.e., inheritance) [33, 36]. Each such context could describe one main routine that is taken over by one CA, where each CA can execute several subroutines that are part of the main routine and together make up the main routine a CA performs.

Dittrich & Seidl [13] study routine dynamics and the emergence of means resulting from it mainly unidirectionally. Additionally to what Dittrich and Seidl [13] describe, our data indicate that when implementing CAs means can be reused for current or subsequent routines and retroactively for prior routines. Thus, we understand the process of means emergence and reuse of such as not merely unidirectional but multidirectional.

We found that while means can partially emerge, means can also be intentionally constructed (e.g., voice-first dialog structures). However, not everything can be reused directly, but specific means may first have to be modified to be reused. Contextual conditions determine if and how means can be reused. For example, in the case studied, the non-customer-specific dialogs of the German and French bots could not be reused directly for the e-banking bot. They first had to be converted into a customer-specific structure. It is essential to understand the contextual conditions that prevent direct reuse and how means can be adapted to be reused [1]. We have identified two ways in which the context changed in the case studied. Contextual changes occurred in three instances related to different languages (i.e., German vs. French), different input channels (i.e., text vs. voice), and different topics (i.e., general FAQ vs. customer-specific dialogs). Furthermore, we observed that means could be reused for a current, a subsequent, or a previous CA.

When CAs take over routines [25, 26] they become the routine while carrying and propagating it in order to enact it. Whenever routines are enacted, materiality is involved [9, 16, 19]. The materiality of CAs describes the space of possibilities such CAs offer or the potential of how they can take over routines and perform them. Due to its materiality, a CA affords and constrains routine enactment. When implementing a CA over time, its materiality may change and thus offer new potential to change the materiality of again other CAs. Means are then material properties that afford further developments and implementations of CAs [21].

Thereby, means can be reused directly through reproduction with a need for some extension or indirectly through recreation. The extent to which the conversational context changes determines whether and when which type of reuse can take effect. We distinguish among two types of contextual change: change in the way a conversation is conducted (Change Type I) (e.g., language, or text vs. voice input), and change in what a conversation is about (Change Type II) (e.g., context or topic). In alignment with the shift towards digital technology that takes over organizational routines is also the notion of material agency with agency defined as the ability to take on action [23]. Thus, not only humans but also technological artifacts can take on agency [23]. CAs as a type of such artifacts can incorporate material agency [9]. Material agency, however, is different from human agency. We discussed above how humans are capable of changing routines [16]. The kind of CAs we are studying here is not (yet) capable of autonomous learning or adapting to changes in the environment [27]. However, their material agency determines how digital materiality (i.e., means) can be reused for the implementation and further development of current, preliminary, and future CAs.

Overall, we contribute to a better understanding of the relationship between routines and CAs with three essential findings. First, to accelerate the implementation of multiple CAs, developers may build on explicit and cognitive means. Second, means may be reused to extend a current CA's functionalities (i.e., mutation) or to transfer functionality to other CAs (i.e., inheritance) [4, 33, 36]. Third, means may not always be reused directly due to changing contextual conditions (e.g., different language contexts or input channels). Thus, reuse depends on how conversational contexts change, determining if means can be reused directly or through adaptation (i.e., reproduction vs. recreation).

8. Conclusion

The division of labor is shifting in a direction where work is mainly distributed among digital technologies such as CAs [23]. One may build on emerging or intentionally constructed means to accelerate how multiple CAs can be implemented successively. Means may be reused to extend a current CA's functionality or transfer functionality to other CAs (mutation vs. inheritance). Depending on how conversational contexts change, means may be reproduced or recreated in order to be reused. Besides the implications of our research, we also must acknowledge its limitations. Our findings need to be further refined and substantiated with additional data. Further research could focus on different CAs to deepen the understanding of means emergence and reuse and its impact on the implementation of multiple successive CAs.

10. References

- [1] P.S. Adler, B. Goldoftas, and D.I. Levine, "Flexibility Versus Efficiency? A Case Study of Model Changeovers in the Toyota Production System", *Organization Science*, 10(1), 1999, pp. 43-68.
- [2] E. Auramäki, R. Hirschheim, and K. Lyytinen, "Modelling offices through discourse analysis: the SAMPO approach", *The Computer Journal*, 35(4), 1992, pp. 342-352.
- [3] A. Baird and L.M. Maruping, "The Next Generation of Research on IS Use: A Theoretical Framework of Delegation to and from Agentic IS Artifacts", *MIS Quarterly*, 45(1), 2021, pp. 315-341.
- [4] V.R. Basili, L.C. Briand, and W.L. Melo, "How reuse influences productivity in object-oriented systems", *Communications of the ACM*, 39(10), 1996, pp. 104-116.
- [5] I. Benbasat, D.K. Goldstein, and M. Mead, "The Case Research Strategy in Studies of Information Systems", *MIS Quarterly*, 11(3), 1987, pp. 369-386.
- [6] N. Berente, K. Lyytinen, Y. Yoo, and J.L. King, "Routines as shock absorbers during organizational transformation: Integration, control, and NASA's enterprise information system", *Organization Science*, 27(3), 2016, pp. 551-572.
- [7] M.D. Cohen, R. Burkhart, G. Dosi, M. Egidi, L. Marengo, M. Warglien, and S. Winter, "Routines and other recurring action patterns of organizations: contemporary research issues", *Industrial and corporate change*, 5(3), 1996, pp. 653-698.
- [8] J.M. Corbin and A. Strauss, "Grounded Theory Research: Procedures, Canons, and Evaluative Criteria", *Qualitative Sociology*, 13(1), 1990, pp. 3-21.
- [9] L. D'Adderio, "Artifacts at the centre of routines: performing the material turn in routines theory",

- Journal of Institutional Economics, 7(2), 2011, pp. 197-230.
- [10] L. D'Adderio, "The replication dilemma unravelled: How organizations enact multiple goals in routine transfer", *Organization Science*, 25(5), 2014, pp. 1325-1350.
- [11] L. D'Adderio and M. Safavi, "Truces and routine dynamics", in *Cambridge Handbook of Routine Dynamics*, Editor^Editors, Cambridge University Press, 2020.
- [12] S. Diederich, T.-B. Lembcke, A.B. Brendel, and L.M. Kolbe, "Understanding the impact that response failure has on how users perceive anthropomorphic conversational service agents: Insights from an online experiment", *AIS Transactions on Human-Computer Interaction*, 13(1), 2021, pp. 82-103.
- [13] K. Dittrich and D. Seidl, "Emerging intentionality in routine dynamics: A pragmatist view", *Academy of Management Journal*, 61(1), 2018, pp. 111-138.
- [14] K.M. Eisenhardt, "Building Theories from Case Study Research", *Academy of Management Review*, 14(4), 1989, pp. 532-550.
- [15] M.S. Feldman and B.T. Pentland, "Reconceptualizing organizational routines as a source of flexibility and change", *Administrative science quarterly*, 48(1), 2003, pp. 94-118.
- [16] M.S. Feldman, B.T. Pentland, L. D'Adderio, and N. Lazaric, "Beyond routines as things: Introduction to the special issue on routine dynamics", *Organization Science*, 27(3), 2016, pp. 505-513.
- [17] H.P. Fung, "Criteria, use cases and effects of information technology process automation (ITPA)", *Advances in Robotics & Automation*, 3, 2014.
- [18] J.A. Howard-Grenville, "The persistence of flexible organizational routines: The role of agency and organizational context", *Organization science*, 16(6), 2005, pp. 618-636.
- [19] P.M. Leonardi, "Digital materiality? How artifacts without matter, matter", *First monday*, 15(6), 2010.
- [20] P.M. Leonardi, "When flexible routines meet flexible technologies: Affordance, constraint, and the imbrication of human and material agencies", *MIS Quarterly*, 35(1), 2011, pp. 147-167.
- [21] P.M. Leonardi and S.R. Barley, "Materiality and change: Challenges to building better theory about technology and organizing", *Information and organization*, 18(3), 2008, pp. 159-176.
- [22] W.J. Orlikowski and S.V. Scott, "10 sociomateriality: challenging the separation of technology, work and organization", *Academy of Management annals*, 2(1), 2008, pp. 433-474.
- [23] B.T. Pentland, M.S. Feldman, M.C. Becker, and P. Liu, "Dynamics of organizational routines: A generative model", *Journal of Management Studies*, 49(8), 2012, pp. 1484-1508.
- [24] R. Pfeifer, M. Lungarella, and F. Iida, "Self-organization, embodiment, and biologically inspired robotics", *science*, 318(5853), 2007, pp. 1088-1093.
- [25] C. Rutschi and J. Dibbern, "Mastering Software Robot Development Projects: Understanding the Association between System Attributes & Design Practices", in *Proceedings of the 52nd Hawaii International Conference on System Sciences*, Hawaii, USA, 2019, pp. 5971-5980.
- [26] C. Rutschi and J. Dibbern, "Towards a Framework of Implementing Software Robots: Transforming Human-executed Routines into Machines", *ACM SIGMIS Database: the DATABASE for Advances in Information Systems*, 51(1), 2020, pp. 104-128.
- [27] A. Salovaara, K. Lyytinen, and E. Penttinen, "High reliability in digital organizing: Mindlessness, the frame problem, and digital operations", *MIS Quarterly*, 2019, pp. 555-578.
- [28] S. Schuetz and V. Venkatesh, "Research Perspectives: The Rise of Human Machines: How Cognitive Computing Systems Challenge Assumptions of User-System Interaction", *Journal of the Association for Information Systems*, 21(2), 2020, pp. 460-482.
- [29] R.M. Schuetzler, G.M. Grimes, J.S. Giboney, and H.K. Rosser, "Deciding Whether and How to Deploy Chatbots", *MIS Quarterly Executive*, 20(1), 2021, pp. Article 4.
- [30] A.-M. Seeger, J. Pfeiffer, and A. Heinzl, "Texting with human-like conversational agents: Designing for anthropomorphism", *Journal of the Association for Information Systems: JAIS*, 2021, pp. tba.
- [31] B.A. Shawar and E. Atwell, "Different measurements metrics to evaluate a chatbot system", in *Proceedings of the Workshop on Bridging the Gap: Academic and Industrial Research in Dialog Technologies*, Association for Computational Linguistics, Rochester, NY, USA, 2007, pp. 89-96.
- [32] J.R. Slaby, "Robotic Automation Emerges as a Threat to Traditional Low-Cost Outsourcing", *HfS Research Ltd*, 2012.
- [33] F. Svahn, L. Mathiassen, and R. Lindgren, "Embracing Digital Innovation in Incumbent Firms: How Volvo Cars Managed Competing Concerns", *Mis Quarterly*, 41(1), 2017, pp. 239-253.
- [34] L. Willcocks and M.C. Lacity, "Service automation: Robots and the future of work", *Steve Brookes Publishing*. 2016.
- [35] R.K. Yin, "Case study research", in *Applied Social Research Methods Series*, Editor^Editors, Sage Publications, Beverly Hills, CA, USA, 2003.
- [36] Y. Yoo, R.J. Boland Jr, K. Lyytinen, and A. Majchrzak, "Organizing for innovation in the digitized world", *Organization science*, 23(5), 2012, pp. 1398-1408.